USE OF DOMINANCE-BASED TOURNAMENT SELECTION TO HANDLE CONSTRAINTS IN GENETIC ALGORITHMS

CARLOS A. COELLO COELLO
CINVESTAV-IPN
Depto. de Ingeniería Eléctrica
Sección de Computación
Av. IPN No. 2508
Col. San Pedro Zacatenco
México, D. F. 07300, MÉXICO
ccoello@cs.cinvestav.mx

EFRÉN MEZURA MONTES
Universidad Veracruzana
Maestría en Inteligencia Artificial
Sebastián Camacho No. 5
Col. Centro
Xalapa, Veracruz 91090, MÉXICO
emezura@lania.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.

INTRODUCTION

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

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

subject to:

$$g_i(\bar{x}) \leq 0, \qquad i = 1, \cdots, n \quad (2)$$

$$h_j(\bar{x}) = 0, \qquad j = 1, \cdots, p \quad (3)$$

where $\bar{x}$ is the solution vector $\bar{x} = [x_1, x_2, \cdots, x_n]^T$, 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 (Michalewicz and Schoenauer, 1996; Coello 2001)), 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 (Coello, 1999), it is possible to derive new constraint-handling techniques that are not only easy to implement, but also computationally efficient.

OUR APPROACH

The idea of using evolutionary multiobjective optimization techniques (Coello, 1999) to handle constraints is not entirely new. Several researchers have reported approaches that rely on the use of multiobjective optimization new (see (Coello, 2000) for a review of this type of techniques). 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 (Fonseca and Fleming, 1995) to the new vector $v = (f, f_1, \cdots, f_m)$, where $f_1, \cdots, f_m$ are the original constraints of the problem. An ideal solution $\bar{x}$ would thus have $f_i(\bar{x}) = 0$ for $1 \leq i \leq m$ and $f(\bar{x}) \leq f(\bar{y})$ for all feasible $\bar{y}$ (assuming minimization). However, the existing techniques have several disadvantages, mainly related to efficiency issues and diversity loss. Those disadvantages were the main motivation for the development of the technique proposed in this paper.

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 (Coello, 1999). 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 (Parmme and Purchase, 1994), 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) (Horn et al., 1994) 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, ($1 - S_r$) 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.

COMPARISON OF RESULTS

While the method has been tested on several examples taken fom the optimization literature, only two will be used here to show the way in which the proposed approach works due to space limitations. Detailed descriptions of these problems may be found in the corresponding references.

Design Variables	Best solution found			
	This paper	**Deb(1991)**	**Siddall(1972)**	**Ragsdell(1976)**
$x_1(h)$	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 beam)

EXAMPLE 1

This problem was solved before by Deb (1991) using a simple genetic algorithm with binary representation, and a traditional penalty function as suggested by Goldberg (1989). It has also been solved by Ragsdell and Phillips (1976) using geometric programming. Ragsdell and Phillips also compared their results with those produced by the methods contained in a software package called "Opti-Sep" (Siddall, 1972), 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 (Siddall, 1972), 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: $0.1 \le x_1 \le 2.0$, $0.1 \le x_2 \le 10.0$, $0.1 \le x_3 \le 10.0$ and $0.1 \le x_4 \le 2.0$. The mean from the 30 runs performed was $f(\bar{x}) = 1.792654$, with a standard deviation of 0.074713. The worst solution found was $f(\bar{x}) = 1.993408$, 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.

Design Variables	Best solution found			
	This paper	**GeneAS(1997)**	**Kannan(1994)**	**Sandgren(1988)**
$x_1(T_s)$	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
$g_1(\bar{x})$	-0.000020	-0.004750	0.000016	-0.204390
$g_2(\bar{x})$	-0.035891	-0.038941	-0.068904	-0.169942
$g_3(\bar{x})$	-27.886075	-3652.876838	-21.220104	54.226012
$g_4(\bar{x})$	-63.345953	-127.321000	-196.310000	-122.299000
$f(\bar{x})$	**6059.946341**	**6410.3811**	**7198.0428**	**8129.1036**

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

EXAMPLE 2

This problem was solved before by Deb (1997) using GeneAS (Genetic Adaptive Search), by Kannan and Kramer using an augmented Lagrangian

Multiplier approach (1994), and by Sandgren (1988) using a branch and bound technique.

Their results were compared against those produced by the approach proposed in 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: $1 \le x_1 \le 99$, $1 \le x_2 \le 99$, $10.0 \le x_3 \le 200.0$ and $10.0 \le x_4 \le 200.0$. 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 $f(\bar{x}) = 6177.253268$, with a standard deviation of 130.929702. The worst solution found was $f(\bar{x}) = 6469.322010$. 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.

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 two 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, 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.

ACKNOWLEDGEMENTS

The first author acknowledges partial support from CINVESTAV through project JIRA’2001/08 and from CONACyT through 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

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

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

Coello C. A., 2000, “Constraint-handling using an evolutionary multiobjective optimization technique”, *Civil Engineering and Environmental Systems*, 17:319—346.

Coello, C.A., 2001, “Theoretical and Numerical Constraint-Handling Techniques used with Evolutionary Algorithms: A Survey of the State of the Art”, *Computer Methods in Applied Mechanics and Engineering*, 2001 (in press).

Deb, K., 1991, “Optimal Design of a Welded Beam via Genetic Algorithms,” *AIAA Journal*, 29(11):2013—2015.

Deb K. 1997, “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.

Fonseca, C. M., and Fleming, P. J., 1995, “An overview of evolutionary algorithms in multiobjective. Optimization,” *Evolutionary Computation*, 3(1):1—16.

Goldberg D. E., 1989, “Genetic Algorithms in Search, Optimization and Machine Learning”. Addison-Wesley Publishing Co., Reading, Massachusetts.

Horn, J., Nafpliotis, N., and Goldberg, D. E., 1994, “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, IEEE Service Center.

Kannan B. K., and Kramer, S. N., 1994, “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.

Michalewicz, Z., and Schoenauer, M.., 1996, “Evolutionary Algorithms for Constrained Parameter Optimization Problems,” *Evolutionary Computation*, 4(1):1--32.

Parmee, I. C., and Purchase, G., 1994, “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, University of Plymouth.

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

Rao S. S. , 1996, “Engineering Optimization”. John Wiley and Sons, third edition.

Sandgren, E. ,1988, “Nonlinear integer and discrete programming in mechanical design.” In *Proceedings of the ASME Design Technology Conference*, pages 95--105, Kissimine, Florida.

Schaffer, J. D., 1985, “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.

Siddall, J. N., 1972, “Analytical Design-Making in Engineering Design”, Prentice-Hall.