

Handling Constraints in Genetic Algorithms using Dominance-Based Tournaments

Carlos A. Coello Coello and 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, MÉXICO
ccoello@cs.cinvestav.mx
emezura@computacion.cs.cinvestav.mx

Abstract

In this paper, we propose a constraint-handling approach for genetic algorithms which uses a dominance-based selection scheme. The proposed approach does not require the fine tuning of a penalty function and does not require extra mechanisms to maintain diversity in the population. The algorithm is validated using several test functions taken from the specialized literature on evolutionary optimization. The results obtained indicate that the approach can produce reasonable results at low computational costs.

1 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:

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

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

where \bar{x} is the solution vector $\bar{x} = [x_1, x_2, \dots, x_n]^T$, 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.

Although many constraint-handling methods have been developed in the last few years for genetic algorithms (see for example [11]), 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] is possible to derive new constraint-handling techniques that are not only easy to implement, but also computationally inexpensive.

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 [7] to the new vector $v=(f, f_1, \dots, f_m)$, where f_1, \dots, f_m are the original constraints of the problem. An ideal solution x would thus have $f_i(x)=0$ $1 \leq i \leq m$ and $f(x) \leq f(y)$ for all feasible y (assuming minimization). Surry et al. [15,14] proposed the use of Pareto ranking [6] and VEGA [13] 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 [15]. It should be added that COMOGA [15] required several extra parameters, although its authors argue [15] that the technique is not particularly sensitive to the values of such parameters. Parmee and Purchase [12] implemented a version of the *Vector Evaluated Genetic Algorithm* (VEGA) [13] 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 [12] 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:

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

where b 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 $d = (x_i - x_j) / (x_i - x_j)$ where $x_i \in S_i$, $x_j \in S_j$, and S_i and S_j are 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 fitted 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 it 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 is the impact of the segment chosen to search in the overall performance of the algorithm. Jiménez and Verdegay [9] 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 ($\max_j g_j(\bar{x})$, for $j = 1, \dots, m$, 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 [15]. 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 [12] 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 [13] 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 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]:

```

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

```

where $g_j(\bar{x})$ refers to the constraint 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 the best possible compromise, 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, 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 Niche-Pareto Genetic Algorithm (NPGA) [8] 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_c), which indicates the minimum number of individuals that will be selected through dominance-based tournament selection. The remainder will be selected using a purely probabilistic approach. In other words, $(1-S_c)$ individuals in the population are probabilistically selected. Each candidate has a probability of 50% of being 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_c will control the diversity of the population. For the experiments reported in this paper, a value close to one (≥ 0.8) was adopted.

The pseudocode is presented below where **oldpop** is the current population, t_{dom} is the size of the comparison set and **flip(P)** is a function that returns TRUE with probability **P**:

```

function select
begin
  candidate_1 = tournlist[0];
  candidate_2 = tournlist[1];
  if (flip(Sr)) /* fitness-feasibility-nondominance based tournament */
  begin
    if (oldpop[candidate_1]=feasible AND oldpop[candidate_2]=feasible)
      if (oldpop[candidate_1].fitness >= oldpop[candidate_2].fitness)
        winner=candidate_1; /* fitness checking */
      else
        winner=candidate_2;
    else /* feasibility checking */
      if (oldpop[candidate_1]=feasible AND oldpop[candidate_2]=nonfeasible)
        winner=candidate_1;
      else
        if (oldpop[candidate_1]=nonfeasible AND oldpop[candidate_2]=feasible)
          winner=candidate_2;
        else
          begin
            shuffle(tournlist); /* nondominance checking */
            candidate_1_dominated = FALSE;
            candidate_2_dominated = FALSE;
            for (i=2 to tdom+2)
              begin
                comparison_individual=tournlist[i];
                if (oldpop[comparison_individual] dominates oldpop[candidate_1])
                  candidate_1_dominated=TRUE;
                if (oldpop[comparison_individual] dominates oldpop[candidate_2])
                  candidate_2_dominated=TRUE;
              end
            if (candidate_1_dominated=TRUE AND candidate_2_dominated=FALSE)
              winner=candidate_2;
            else
              if (candidate_1_dominated=FALSE AND candidate_2_dominated=TRUE)
                winner=candidate_1;
              else /* tie break with accumulated constraint violation */
                if (oldpop[candidate_1].sumviol < oldpop[candidate_2].sumviol)
                  winner=candidate_1;
                else
                  winner=candidate_2;
              end
            end
          end
        end
      end
    else
      if (flip(0.5)) /* pure probabilistic selection */
        winner=candidate_1;
      else
        winner=candidate_2;
    end
  end
  return(winner);
end

```

4 Comparison of Results

To validate our approach, we have used the well-known benchmark proposed in [11]. The specific test functions used are the following:

1. g02

$$\text{Maximize: } f(\bar{x}) = \left| \frac{\sum_{i=1}^n \cos^4(x_i) - 2 \prod_{i=1}^n \cos^2(x_i)}{\sqrt{\sum_{i=1}^n i x_i^2}} \right| \quad (5)$$

Subject to:

$$g_1(\bar{x}) = 0.75 - \prod_{i=1}^n x_i \leq 0$$

$$g_2(\bar{x}) = \sum_{i=1}^n x_i - 0.75n \leq 0$$

where $n = 20$, $0 \leq x_i \leq 10$ ($i = 1, \dots, n$)

2. g04

Minimize:

$$f(\bar{x}) = 5.3578547 + 0.8356891 x_1 x_5 + 37.293239 x_1 - 40792.141 \quad (6)$$

Subject to:

$$g_1(\bar{x}) = 85.334407 + 0.0056858 x_2 x_5 + 0.0006262 x_1 x_4 - 0.0022053 x_3 x_5 - 92 \leq 0$$

$$g_2(\bar{x}) = -85.334407 - 0.0056858 x_2 x_5 - 0.0006262 x_1 x_4 + 0.0022053 x_3 x_5 \leq 0$$

$$g_3(\bar{x}) = 80.51249 + 0.0071317 x_2 x_5 + 0.0029955 x_1 x_2 + 0.0021813 x_3^2 - 110 \leq 0$$

$$g_4(\bar{x}) = -80.51249 - 0.0071317 x_2 x_5 - 0.0029955 x_1 x_2 - 0.0021813 x_3^2 + 90 \leq 0$$

$$g_5(\bar{x}) = 9.300961 + 0.0047026 x_3 x_5 + 0.0012547 x_1 x_3 + 0.0019085 x_3 x_4 - 25 \leq 0$$

$$g_6(\bar{x}) = -9.300961 - 0.0047026 x_3 x_5 - 0.0012547 x_1 x_3 - 0.0019085 x_3 x_4 + 20 \leq 0$$

where $78 \leq x_1 \leq 102$, $33 \leq x_2 \leq 45$, $27 \leq x_i \leq 45$, ($i = 3, 4, 5$)

3. g11

Minimize:

$$f(\bar{x}) = x_1^2 + (x_2 - 1)^2 \quad (7)$$

Subject to:

$$h(\bar{x}) = x_2 - x_1^2 = 0$$

where $-1 \leq x_1 \leq 1$, $-1 \leq x_2 \leq 1$

4. g12

$$\text{Maximize: } f(\bar{x}) = \frac{100 - (x_1 - 5)^2 - (x_2 - 5)^2 - (x_3 - 5)^2}{100} \quad (8)$$

$$\text{Subject to: } g(\bar{x}) = (x_1 - p)^2 + (x_2 - q)^2 + (x_3 - r)^2 - 0.0625 \leq 0 \quad (8)$$

where: $0 \leq x_i \leq 10$ ($i=1,2,3$), and $p, q, r = 1, 2, \dots, 9$. The feasible region of the search space consists of 9^3 disjointed spheres. A point (x_1, x_2, x_3) is feasible if and only if there exist p, q, r such that the above inequality holds.

In the experiments performed, 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, $S_i = 0.99$ (i.e., one out of every one hundred selections are done probabilistically, rather than in a deterministic way), tournament size= 10.

TF	optimal	Best Result		Mean Result		Worst Result	
		DBTS	KM	DBTS	KM	DBTS	KM
g02	0.803619	0.787933	0.79953	0.751301	0.79671	0.689747	0.79119
g04	-30665.539	-30659.997	-30664.5	-30630.670	-30655.3	-30551.493	-30645.9
g11	0.750	0.749001	0.75	0.75012467	0.75	0.760132	0.75
g12	1.000	1.000	0.99999857	1.000	0.999134613	1.000	0.991950498

Table 1. Comparison of the results for the test functions selected from [11]. DBTS stands for Dominance-Based Tournament Selection and KM stands for Koziel & Michalewicz's technique.

TF	optimal	Best Result		Mean Result		Worst Result	
		DBTS	DP	DBTS	DP	DBTS	DP
g02	0.803619	0.787933	0.653277	0.751301	0.490935	0.689747	0.369007
g04	-30665.539	-30659.997	-30653.27107	-30630.670	-30473.41012	-30551.493	-29973.77619
g11	0.750	0.749001	-	0.75012467	-	0.760132	-
g12	1.000	1.000	1.000000	1.000	1.000000	1.000	1.000000

Table 2. Comparison of the results for the test functions selected from [11]. DBTS stands for Dominance-Based Tournament Selection and DP stands for a Death Penalty technique. The symbol “-“ means that the approach could not find any feasible solution in generation 0.

TF	optimal	Best Result		Mean Result		Worst Result	
		DBTS	DYP	DBTS	DYP	DBTS	DYP
g02	0.803619	0.787933	0.627836	0.751301	0.480428	0.689747	0.376845
g04	-30665.539	-30659.997	-30612.62606	-30630.670	-30461.38812	-30551.493	-29833.09320
g11	0.750	0.749001	0.750430	0.75012467	0.763094	0.760132	0.934769
g12	1.000	1.000	1.000000	1.000	1.000000	1.000	1.000000

Table 3. Comparison of the results for the test functions selected from [11]. DBTS stands for Dominance-Based Tournament Selection and DYP stands for a Dynamic Penalty technique.

Our results are compared to the homomorphous maps of Koziel & Michalewicz [10] in Table 1. The results of Koziel and Michalewicz were obtained with 1,400,000 fitness function evaluations, whereas our approach required only 80,000 fitness function evaluations.

As can be seen in Table 1, our approach produces reasonable results with respect to the homomorphous maps (which is considerably more difficult to implement) at a fraction of its computational cost.

In table 2, DBTS finds better results in all the functions. Moreover, the consistency of the solutions are much better than the Death Penalty approach, because the mean and worst solution of DBTS are closer to the best than in the DP approach. Furthermore, in the function g11 DP could not find any feasible solution.

A similar behavior is shown in Table 3, where DBTS is compared with a Dynamic Penalty approach. DBTS has more consistency in its results.

The previous results indicate that the use of concepts borrowed from multiobjective optimization may be a viable alternative to handle constraints in evolutionary algorithms.

5 Discussion of results

Despite of the fact that there are other approaches that use either nondominance or the selection criteria shown before as DBTS, there are punctual differences. Coello [4] uses the concept of nondominance in his approach. However, his approach compares each individual with all the remaining population in order to obtain a ranking for it. Moreover, Stochastic Universal Sampling is used as selection technique.

Deb [5] uses similar criteria as DBTS in his technique. But he uses binary tournament. Also, his approach calculates the amount of unfeasibility of a solution in a different way.

Coello [4] argues that the use of DBTS should generate a high selection pressure. Thus, a procedure to maintain diversity must be implemented. It will increase the computational cost. However, the parameter S_r helps the search to get random selected individual (unfeasible, dominated, feasible with a low fitness) in every step of the process.

The consistency on our approach is shown in the comparison with Penalty function-based approaches. DBTS shows smaller differences between the best, mean and worst solutions than them.

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 technique approached well the optima of several (standard) test problems with a relatively low number of fitness function evaluations. However, more work in this area is still necessary in order to improve the quality of the results produced by the proposed approach.

Our future work involves the development of a self-adaptation mechanism that makes unnecessary the fine tuning of the parameters of the genetic algorithm (i.e., crossover rate, mutation rate and the selection ratio). Also, the development of alternative constraint-handling techniques based on multiobjective optimization concepts is under way.

Acknowledgments

The first author acknowledges support from CONACyT through the NSF-CONACyT project number 32999 A.

References

- 1 Camponogara E. and Talukdar S. (1997) A Genetic Algorithm for Constrained and Multiobjective Optimization. In JarmoT. Alander, editor, 3rd Nordic Workshop on Genetic Algorithms and Their Applications (3NWGA), pages 49--62, Vaasa, Finland. University of Vaasa.

- 2 Chankong V. and Haimes YY. (1983) Multiobjective Decision Making: Theory and Methodology. Systems Science and Engineering. North-Holland.
- 3 Coello C. (1999) A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. Knowledge and Information Systems. An International Journal, 1(3):269—308.
- 4 Coello C. (2000) Treating Constraints as Objectives for Single-Objective Evolutionary Optimization. Engineering Optimization, 32(3):275—308.
- 5 Deb K. and Agrawal S. (1999) A Niche-Penalty Approach for Constraint Handling in Genetic Algorithms. In Proceedings of the ICANNGA, Portoroz, Slovenia, 1999.
- 6 Fonseca C. and Fleming P. (1993) 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. University of Illinois at Urbana-Champaign, Morgan Kaufmann Publishers.
- 7 Fonseca C. and Fleming P. (1995) An overview of evolutionary algorithms in multiobjective optimization. Evolutionary Computation, 3(1):1--16.
- 8 Horn J., Nafpliotis N. and Goldberg D. (1994) A Niche 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.
- 9 Jiménez F. and Verdegay J. (1999) Evolutionary techniques for constrained optimization problems. In 7th European Congress on Intelligent Techniques and Soft Computing (EUFIT'99), Aachen, Germany. Springer-Verlag.
- 10 Koziel S. and Michalewicz Z. (1999). Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization. Evolutionary Computation, 7(1):19--44.
- 11 Michalewicz Z. and Schoenauer M. (1996) Evolutionary Algorithms for Constrained Parameter Optimization Problems. Evolutionary Computation, 4(1):1--32.
- 12 Parmee IC. 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.

- 13 Schaffer 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.
- 14 Surry P. and Radcliffe NJ. (1997) The COMOGA Method: Constrained Optimisation by Multiobjective Genetic Algorithms. Control and Cybernetics, 26(3).
- 15 Surry P., Radcliffe NJ. and Boyd ID (1995) 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.