A UNIVERSAL ECLECTIC GENETIC ALGORITHM FOR CONSTRAINED OPTIMIZATION

Angel Kuri Morales
Carlos Villegas Quezada
CENTRO DE INVESTIGACION EN COMPUTACION
INSTITUTO POLITECNICO NACIONAL
Juan de Dios Bátiz, S/N; Col. Lindavista; México, D.F. C.P. 07738
729-6000 ext. 56547
akuri@pollux.cic.ipn.mx

ABSTRACT. Several non-traditional methods of solving nonlinear constrained optimization problems have been introduced in the past: evolutionary programming, genetic algorithms and simulated annealing, among others. However these methods present some limitations. In this paper we describe a GA which departs from canonical GAs (CGA) and achieves most of the characteristics of what, in the literature, has been called an Idealized GA (IGA). This GA is designed to approach the IGA and we have it called an *Eclectic Genetic Algorithm* (EGA). We describe the application of an EGA to the solution of a set of six constrained nonlinear problems which have been recently discussed in the literature. Also the performance of an EGA is compared those of the CGA and eliTist GA (TGA).

1. INTRODUCTION

The traditional methods of linear and nonlinear programming are generally impractical when they are applied to constrained nonlinear programming problems. The general nonlinear problem is to find an extremum of an objective function subject to equality and/or inequality constraints, in general the constraints may be linear and/or nonlinear. The nonlinear programming problem can be formally stated as

$$\text{Minimize} \quad f(x) \quad \text{subject to :}$$
$$m \quad \text{linear and/ or nonlinear equality constraints}$$
$$h_j(x) = 0 \quad j = 1, \dots, m \quad \text{and}$$
$$(p-m) \quad \text{linear and/ or nonlinear inequality constraints}$$
$$g_j(x) \geq 0 \quad j = m+1, \dots p$$

Although in some special cases the equality constraints can be explicitly solved for selected variables and those variables eliminated from the problem as independent variables (reducing the problem to one with inequality constraints only) most often the equality constraints can be solved only implicitly and must be retained.
Several traditional algorithms have been developed to try to solve some class of problems of nonlinear programming (Unconstrained optimization, Quadratic programming, Convex programming, Geometric programming, etc.). It is not easy to determine which method one should select for a given nonlinear programming problem. Also, among the deficiencies that are present in the solution of this type of problems are the difficulty to be able to distinguish between a local maximum (or minimum) and a global maximum (or minimum). Constrained optimization problems often perform poorly when faced with nonconvex or disjoint feasible regions. Generally classic linear and nonlinear programming methods are often either unsuitable or impractical when applied to these constrained problems.
Some recent developments in constraint handling techniques have been described: penalty functions (Le Riche,1995), hybrid methods (Kim and Myung,1997), dynamic penalties and evolution strategies (Michalewicz,1995), fuzzy evolutionary programming (Van Le,1996), etc.
In this paper an Eclectic Genetic Algorithm (EGA) is presented. It is based on a set of strategies which tend to attain the characteristics that an idealized GA should present. The EGA is applied to the resolution of six problems of nonlinear optimization with restrictions proposed in the literature. The results of the application of an EGA are compared with the results obtained from the application of the Canonical GA (CGA), and the eliTist GA (TGA).

2. ALGORITHMS

In the past a vast number of different variations of what is called a GA have been introduced. Most of these variations have stemmed from a very simple form of a GA which is sometimes called canonical. In the CGA there are three operators: a) Selection, b) Crossover and c) Mutation. Selection is proportional, crossover is 1-point and mutation is

uniform (Holland, 1975).
One very interesting characteristic not present in the CGA is the so called elitism. Simply expressed, elitism means that certain individuals are artificially kept in the parent population from generation to generation to ensure that some desirable treats, once found by the algorithm, are not lost because of the stochastic nature of the selection operator. Typically, the best individual, up to the n-th generation, is sequestered to guarantee that random statistical fluctuations will not lead to irretrievable loss of desired characteristics. When such a mechanism is included in the GA, this GA is called an eliTist GA (TGA).

The characterization of CGAs was attempted, initially, by considering the so called schemas (Mitchell,1996). Schema analysis gives rise to the Schema Theorem (Goldberg,1989). Some authors have attempted to generalize this theorem. We stress the fact that the schema theorem and similar expressions are restricted to behaviors which depend, basically, on the CGA model. Also, analyzing a GA as a Markov chain, it is possible to show that a CGA will not converge to the best solution. It is also possible to show that a TGA will converge to the best solution (Rudolph,1997). Nevertheless, the theoretical conclusions just mentioned do not place a bound on time. That is, although we know that a TGA will converge to the best solution we do not know how much time the CGA will invest in this process.

An idealized GA (IGA) which works in accordance with the *Building Block Hipothesis* (BBH) has been defined (Mitchell,1996). This IGA works on one string at a time and, although there is no population defined, it still captures the essence of the BBH. A GA will approach an IGA if:

a) Samples are independent. The population has to be large enough, the selection process has to be slow enough and the mutation rate high enough so that no single locus is fixed at a single value in a large majority of the strings of the population.
b) Desired Schemas are Sequestered. Selection has to be strong enough to preserve desired schemas that have been discovered but also slow enough to prevent significant hitchhiking on some highly fit schemas.
c) Crossover is Instantaneous. The crossover rate has to be such that the time for crossover that combines two desired schemas is small with respect to the discovery time for such schemas.
d) The String is Large. The string has to be long so that the speedup factor is significant.

The above mechanisms are not mutually compatible in a CGA. We propose what we have called an Eclectic Genetic Algorithm (EGA) which seems to achieve all those goals. The term *eclectic* refers to our explicit intent of achieving an algorithm which takes the best out of all strategies, without regard for orthodoxy. In the EGA we incorporate the following (Kuri,1998):

a) Full elitism over a set of size *n* of the last population. Given that we have tested *nk* individuals by generation *k*, our population will consist of the best n up to that point.
b) A deterministic selection scheme (as opposed to the traditional proportional selection operator). We propose to emphasize genetic variety by imposing a strategy which enforces crossover of predefined individuals. In this scheme, the *i-th* individual is crossed with the *(n-i+1)-th* individual.
c) Annular crossover.
d) Selective invocation of a random mutation variation of a hill climber (RMHC).
e) Population self-adaptation of the following parameters: the number of offspring, crossover probability and mutation probability.

3. EXPERIMENTS

In order to test the behavior of the EGA and the differences with respect to CGA and TGA, we chose a set of 6 nonlinear optimization problems which have been discussed in the literature (Michalewicz, 1994). The problems which were solved are:

P1) Minimize $f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$, subject to
$c_1: x_1 + x_2^2 \geq 0;\; c_2: x_1^2 + x_2 \geq 0;\; -0.5 \leq x_1 \leq 0.5;\; x_2 \leq 1.0$

P2) Minimize $f(x) = -x_1 - x_2$, subject to $c_1: x_2 \leq 2x_1^4 - 8x_1^3 + 8x_1^2 + 2;$
$c_2: x_2 \leq 4x_1^4 - 32x_1^3 + 88x_1^2 - 96x_1 + 36;\;\; 0 \leq x_1 \leq 3$ and $0 \leq x_2 \leq 4$

P3) Minimize $f(x) = 0.01x_1^2 + x_2^2$, subject to $c_1: x_1 x_2 - 25 \geq 0;$
$c_2: x_1^2 + x_2^2 - 25 \geq 0;\;\; 2 \leq x_1 \leq 50$ and $0 \leq x_2 \leq 50$

P4) Minimize $f(x) = (x_1 - 2)^2 + (x_2 - 1)^2$, subject to $c_1: -x_1^2 + x_2 \geq 0$ and $x_1 + x_2 \leq 2$

P5) Minimize $f(x) = e^{x_1 x_2 x_3 x_4 x_5}$, subject to $\sum_{i=1}^{5} x_i^2 = 10;\ x_2 x_3 - 5 x_4 x_5 = 0;$

$x_1^3 x_2^3 = -1;\ -2.3 \le x_i \le +2.3, i = 1,2;\ \ -3.2 \le x_i \le +3.2, i = 3,4,5$

P6) Minimize $f(x) = -\mathrm{sgn}(x)$, subject to $-1 \le x \le 2$

For each problem and each one of the three methods (CGA, TGA, EGA), we performed computer simulations with different seeds for the random number generator, different population ranges, different number of generations and different probability of crossover and mutation. The EGA generates its own values of Pm, Pc and n.

The EGA maintains feasibility of all constraints using the following formula to evaluate the population:

$$eval(\overline{X}) = \begin{cases} f(\overline{X}), & if\ \overline{X} \in \mathbf{S} \cap \mathfrak{F} \\ K - \sum_{i=1}^{s} \frac{K}{m} \end{cases}$$

where $K \rightarrow \infty$

s = number of constraints satisfied

m = number of constratints

S defines the space of all values

F defines the space of feasible values.

In the case of the CGA we set a maximum of 2000 generations for each one of the problems and the different set of parameters (Pm, Pc, etc.). In none of the cases a satisfactory result was obtained. The behavior is erratic as may be observed in figure 1. Here we present the *Results Achievement* (RA) (defined as average of the maximum fitness of the different simulations) for the first 290 generations. In problem 5, the results were not meaningful.

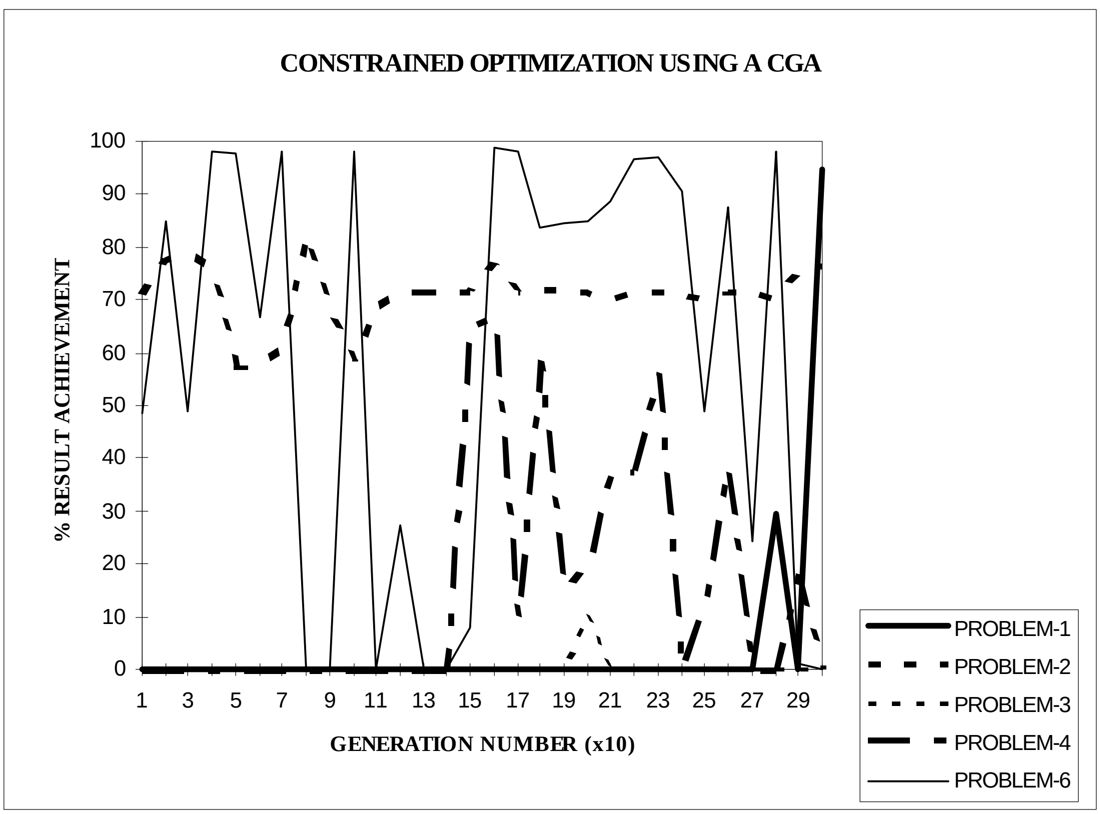

FIGURE 1. PERCENTUAL RA IN FIVE CONSTRAINED PROBLEMS USING CGA

In figure 2 the behavior of the EGA for the six problems is shown. Notice that a stable RA is reached quickly (on the average 40 generations). In six of the problems the required result was attained directly. Only in problem 6 the result was only approximate. The EGA reached 99.9% of the theoretical fitness (after only 280 generations).

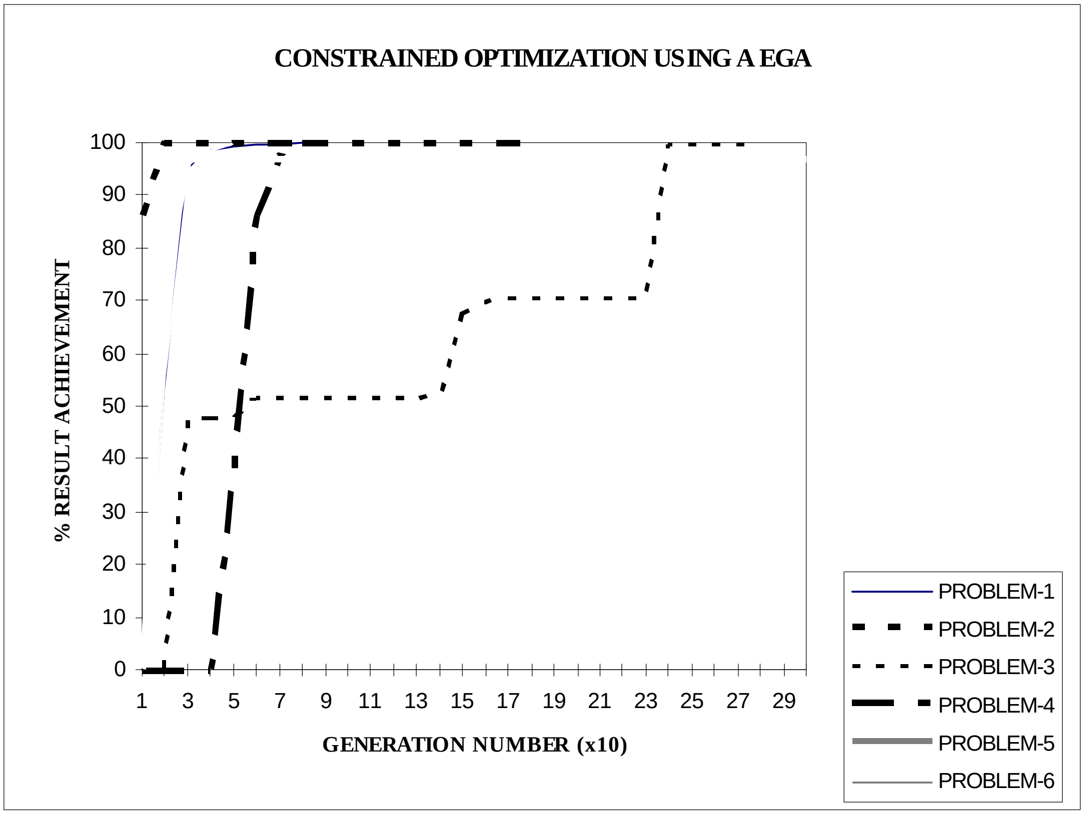

FIGURE 2. PERCENTUAL RA IN SIX CONSTRAINED PROBLEMS USING EGA

In the TGA, the results were obtained in between 400 and 2000 generations (average 1300 generations). In problem 5, the results were not meaningful. The performance of EGA is better than that of TGA.

The results obtained by EGA may be found in table 1. As may be seen from that table, EGA performed satisfactorily for all the six problems considered. These problems are considered to be GA-hard.

PROBLEM	FITNESS THEORETICAL	FITNESS OBSERVED	VECTOR SOLUTION THEORETICAL	VECTOR SOLUTION OBSERVED
P1	0.25	0.250047	(0.5,0.25)	(0.499952, 0.249969)
P2	-5.5079	-5.50270	(2.3295,3.1783)	(2.3309, 3.1718)
P3	5.0	5.00050	(15.8114,1.5811)	((15.7007, 1.5923)
P4	1.0	0.999996	(1.0, 1.0)	(1.00001, 0.999995)
P5	0.053949	0.055575	(-1.71714,1.59571, 1.82725,-0.76364, -0.76364)	(-1.69657, 1.57174, -1.85649, -0.67265, +0.86789)
P6	-2.0	-1.999995	(+2.0)	(+1.999995)

TABLE 1. EGA APPLIED TO CONSTRAINED OPTIMIZATION PROBLEMS

4. CONCLUSIONS

The experimental results seem to confirm our expectations about the EGA. The efficiency of this method surpasses that of CGA and TGA significantly. It is interesting to observe the ease with which we are able to solve the group of problems. There was no need for initial provisions to solve the problems. Except for problem 5, the initial population was generated randomly. In problem 5 the same EGA was used to find an initial group of 60 feasible individuals.
One should also notice the relatively short number of generations involved to obtain convergence. This contrasts sharply with other efforts at solving the same set reported in the past (Kim and Myung, 1997).
There, the authors go into considerable trouble to achieve convergence via a sophisticated double-chain evolutionary algorithm. We also point out that the self-adaptive nature of the EGA allowed it to perform with very slight parameter alterations throughout the test set. There remains to conduct a systematic analysis of its behavior in a more general environment. Informal tests conducted so far, however, seem to reinforce our present conclusions: the EGA has been used successfully to solve the iterated prisioner's dilemma, in genetic automata, and in unsupervised learning.

REFERENCES

Goldberg, D. 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley.

Holland, J.H. 1975. Adaptation in Natural and Artificial Systems. Ann Arbor, MI. University of Michigan Press.

Kim, Jong-Hwan; Myung, Hyun 1997. Evolutionary Programming Techniques for Constrained-Optimization Problems. IEEE Transactions on Evolutionary Computation Vol. 1, No. 2, July, pp. 129-140.

Kuri, Angel 1998. A Universal Eclectic Genetic Algorithm. Sixth Iberoamerican Conference on Artificial Intelligence. *to be published.*

Le Riche, Rodolphe G. 1995. A Segregated Genetic Algorithm for Constrained Structural Optimization. Proc. 6th Int. Conf. Genetic Algorithms. L. J. Eshelman (Ed), Los Altos, CA. Morgan Kaufmann, pp 558-565.

Michalewicz, Z.; Attia, N. 1994. Evolutionary optimization of constrained problems. Proc. 3rd Annu. Conf. Evolutionary Programming. A.V. Sebald and L.J. Fogel (Eds), River Edge, NJ, World Scientific, pp. 99-108.

Michalewicz, Z. 1995. Genetic algorithms, Numerical Optimization, and Constraints. Proc. 6th Int. Conf. Genetic Algorithms, L. J. Eshelman (Ed), Los Altos, CA, Morgan Kaufmann, pp. 151-158.

Mitchell, M. 1996. An Introduction to Genetic Algorithms. Cambridge, MA. MIT Press.

Rudolph, G. 1997. Convergence Analysis of Canonical Genetic Algorithms. IEEE Trans. Neural Networks. Special issue on Evolutionary Computation.

Van Le, T 1996. A Fuzzy Evolutionary Approach to Constrained Optimization Problems. IEEE Proceedings International Conference on Evolutionary Computation; Nagoya, Japan; pp. 274-278.