EVOLUTIONARY OPTIMIZATION OF CONSTRAINED PROBLEMS

Zbigniew Michalewicz
Department of Computer Science, University of North Carolina
Charlotte, NC 28223, USA

and

Naguib F. Attia
Department of Mathematics and Computer Science
Johnson C. Smith University, Charlotte, NC 28223, USA

ABSTRACT

In this paper we discuss a construction of Genocop II, a hybrid optimization system for general nonlinear programming problems. We present the first experimental results of the system on five test cases. These include a variety of objective functions with nonlinear constraints. The results are encouraging.

1. Introduction

The general nonlinear programming problem $\mathcal{NP}$ is to find $\overline{X}$ so as to

optimize $f(\overline{X})$, $\overline{X} = (x_1, \ldots, x_q) \in R^q$,

subject to $p \geq 0$ equations:

$$c_i(\overline{X}) = 0, \ i = 0, \ldots, p,$$

and $m - p \geq 0$ inequalities:

$$c_i(\overline{X}) \leq 0, \ i = p+1, \ldots, m.$$

In this paper we discuss a new hybrid system, Genocop II, to solve optimization problems in this class. The name Genocop is taken from the expression "GEnetic algorithm for Numerical Optimization for COnstrainted Problems." The first version of the Genocop system, Genocop I, handled only linear constraints[17,18,19], whereas the current version, Genocop II, handles any set of constraints.

The concept of the presented system is based on the ideas taken from the recent developments in area of optimization[3] combined with iterative execution of the Genocop I; these executions are controlled by a temperature of the system (which can be also interpreted as a variable penalty coefficient).

The prototype of the new system Genocop II was run on several test-cases; the results of experiments are presented and compared with known optima.

The paper is organized as follows. The next section summarizes one traditional approach to solve nonlinear programming problems, which was adopted as a basis for Genocop II. Section 3 introduces the system Genocop II, and the following section discusses several nonlinear test cases together with the experimental results of the Genocop II. The final section contains conclusions and some directions for future work.

2. A Traditional Calculus Based Method

Calculus-based methods assume that the objective function $f(\overline{X})$ and all constraints are twice continuously differentiable functions of $\overline{X}$. The general approach of most methods is to transform the nonlinear problem $\mathcal{NP}$ into a sequence of solvable subproblems. The amount of work involved in a subproblem varies considerably among methods. These methods require explicit (or implicit) second derivative calculations of the objective (or transformed) function, which in some methods can be ill-conditioned and cause the algorithm to fail.

During the last 30 years there has been considerable research directed toward the nonlinear optimization problems and progress has been made in theory and practice[12]. Several approaches have been developed in this area, among these are: the sequential quadratic penalty function[7,3], recursive quadratic programming method[6], penalty trajectory methods[20], and the SOLVER method[10]. In this section, we discuss briefly one of these approaches, the sequential quadratic penalty function method.

The method replaces a problem $\mathcal{NP}$ by the problem $\mathcal{NP}'$:

optimize $F(\overline{X}, r) = f(\overline{X}) + \frac{1}{2r}\overline{C}^T\overline{C}$,

where $r > 0$ and $\overline{C}$ is a vector of all active constraints $c_1, \ldots, c_\ell$.

Fiacco and McCormick[15] have shown that the solutions of $\mathcal{NP}$ and $\mathcal{NP}'$ are identical in the limit as $r \longrightarrow 0$. It was thought that $\mathcal{NP}'$ could then be solved simply by minimizing $F(\overline{X}, r)$ for a sequence of decreasing positive values of r by Newton's method[9]. This hope, however, was short-lived, because minimizing $F(\overline{X}, r)$ proved to be extremely inefficient for the smaller values of r; it was shown by Murray[20] that this was due to the Hessian matrix of $F(\overline{X}, r)$ becoming increasingly ill-conditioned as $r \longrightarrow 0$. As there seemed to be no obvious way of overcoming this problem, the method gradually fell into disuse. More recently, Broyden and Attia[8,7] offered a method that overcomes the numerical difficulties associated with the simple quadratic penalty function. The computation of the search direction does not require the solution of any system of linear equations, and can thus be expected to require less work than is needed for some other algorithms. The method also provides an automatic technique for calculating the initial value for the parameter r and its successive values[7].

3. The System Genocop II

The technique discussed in the previous section, together with the existing system Genocop I, was used in construction of a new system, Genocop II. The structure of the Genocop II is given in Figure 1. We discuss the steps of this algorithm in the remaining part of this section.

```
procedure Genocop II
begin
  t ⟵ 0
  split the set of constraints C into
     C = L ∪ N_e ∪ N_i
  select a starting point X̄_s
  set the set of active constraints, A to
     A ⟵ N_e ∪ V
  set temperature τ ⟵ τ_0
  while (not termination-condition) do
  begin
    t ⟵ t + 1
    execute Genocop I for the function
      F(X̄, τ) = f(X̄) + (1/2τ) Ā^T Ā
      with linear constraints L
      and the starting point X̄_s
    save the best individual X̄*:
      X̄_s ⟵ X̄*
    update A:
      A ← A − S ∪ V,
    decrease temperature τ:
      τ ← g(τ, t)
  end
end
```

Figure 1: The structure of Genocop II.

There are several steps of the algorithm in the first phase of its execution (before it enters the *while* loop). The parameter t (which counts the number of iterations of the algorithm, i.e., the number of times the algorithm Genocop I is applied) is initialized to zero. The set of all constraints C is divided into three subsets: linear constraints L, nonlinear equations N_e and nonlinear inequalities N_i. A starting point $\overline{X}_s$ (which need not be feasible) for the following optimization process is selected (or a user is prompted for it). The set of active constraints A consists initially of elements of N_e and set $V \subseteq N_i$ of violated constraints from N_i. A constraint $c_j \in N_i$ is violated at point $\overline{X}$ iff $c_j(\overline{X}) > \delta$ ($j = p+1, \ldots, m$), where δ is a parameter of the method. Finally, the initial temperature of the system τ is set to τ_0 (a parameter of the method).

In the main loop of the algorithm we apply Genocop I to optimize a modified function

$F(\overline{X}, \tau) = f(\overline{X}) + \frac{1}{2\tau}\overline{A}^T\overline{A}$

with linear constraints L. Note that the initial population for Genocop I consists of *pop_size* identical copies (of the initial point for the first iteration and of the best saved point for subsequent ones); several mutation operators introduce diversity in the population at early stages of the process. When Genocop I converges, its best individual $\overline{X}^*$ is saved and used later as the starting point $\overline{X}_s$ for the next iteration. However, the next iteration is executed with a decreased value of the temperature parameter ($\tau \leftarrow g(\tau, t)$) and a new set of active constraints A:

$A \leftarrow A - S \cup V$,

where S and V are subsets of N_i satisfied and violated by $\overline{X}^*$, respectively.

The mechanism of the algorithm is illustrated on the following example. The problem is to

minimize $f(\overline{X}) = x_1 \cdot x_2^2$,

subject to one nonlinear constraint:

$c_1 \; : \; 2 - x_1^2 - x_2^2 \geq 0.$

The known global solution is $\overline{X}^* = (-0.816497, -1.154701)$, and $f(\overline{X}^*) = -1.088662$. The starting feasible point is $\overline{X}_0 = (-0.99, -0.99)$. After the first iteration of Genocop II (A is empty) the system converged to $\overline{X}_1 = (-1.5, -1.5)$, $f(\overline{X}_1) = -3.375$. The point $\overline{X}_1$ violates the constraint c_1, which becomes active. The point $\overline{X}_1$ is used as the starting point for the second iteration. The second iteration ($\tau = 10^{-1}$, $A = \{c_1\}$) resulted in $\overline{X}_2 = (-0.831595, -1.179690)$, $f(\overline{X}_2) = -1.122678$. The point $\overline{X}_2$ is used as the starting point for the third iteration. The third iteration ($\tau = 10^{-2}$, $A = \{c_1\}$) resulted in $\overline{X}_3 = (-0.815862, -1.158801)$, $f(\overline{X}_3) = -1.09985$. The sequence of points $\overline{X}_t$ (where $t = 4, 5, \ldots$ is the iteration number of the algorithm) approaches the optimum.

4. Test Cases

In order to evaluate the method of Genocop II, a set of test problems has been carefully selected to indicate the performance of the algorithm and to illustrate that it has been successful in practice. The five test cases include quadratic, nonlinear, and discontinuous functions with several nonlinear constraints.

All runs of the system were performed on SUN SPARC station 2. We used the following parameters for Genocop I in all experiments:

> *pop_size* $= 70$, $k = 28$ (number of parents in each generation), $b = 2$ (coefficient for non-uniform mutation), $a = 0.25$ (parameter of arithmetical crossover), $\delta = 0.01$ (parameter which determines whether or not a constraint is active).

In most cases, the initial temperature τ_0 was set at 1 (i.e., $g(\tau, 0) = 1$); additionally, $g(\tau, t) = 10^{-1} \cdot g(\tau, t-1)$.

Genocop II was executed ten times for each test case. For most problems, the number of generations necessary for Genocop I to converge was 1000 (more difficult problems required larger number of iterations). We did not report the computational times for these test cases, since we do not have full implementation of Genocop II yet. The actions of the system were simulated by executing its external loop in manual fashion: when Genocop I converges, the best point is incorporated as the starting point for the next iteration, the constraints are checked for their activity status, and the evaluation function is adjusted accordingly.

All test cases and the results of the Genocop II system are reported in the following subsections.

4.1. Test Case #1

The problem[14] is

$$\text{minimize } f(\overline{X}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2,$$

subject to nonlinear constraints:

$$\begin{aligned} c_1 &: \ x_1 + x_2^2 \geq 0, \\ c_2 &: \ x_1^2 + x_2 \geq 0, \end{aligned}$$

and bounds:

$$-0.5 \leq x_1 \leq 0.5, \text{ and } x_2 \leq 1.0.$$

The known global solution is $\overline{X}^* = (0.5, 0.25)$, and $f(\overline{X}^*) = 0.25$. The starting feasible point is $\overline{X}_0 = (0, 0)$.

Genocop II found the exact optimum in one iteration, since none of the nonlinear constraints are active at the optimum.

4.2. Test Case #2

The problem[11] is

$$\text{minimize } f(x, y) = -x - y,$$

subject to nonlinear constraints:

$$\begin{aligned} c_1 &: \ y \leq 2x^4 - 8x^3 + 8x^2 + 2, \\ c_2 &: \ y \leq 4x^4 - 32x^3 + 88x^2 - 96x + 36, \end{aligned}$$

and bounds:

$$0 \leq x \leq 3 \text{ and } 0 \leq y \leq 4.$$

Iteration number	The best point	Active constraints
0	(0,0)	none
1	(3,4)	c_2
2	(2.06, 3.98)	c_1, c_2
3	(2.3298, 3.1839)	c_1, c_2
4	(2.3295, 3.1790)	c_1, c_2

Table 1: Progress of Genocop II on test case #2; for iteration 0 the best point is the starting point.

The known global solution is $\overline{X}^* = (2.3295, 3.1783)$, and $f(\overline{X}^*) = -5.5079$. The starting feasible point is $\overline{X}_0 = (0, 0)$. The feasible region is almost disconnected.

Genocop II approached the optimum very closely at the 4th iteration. The progress of the system is reported in the table 1.

4.3. Test Case #3

The problem[11] is

$$\text{minimize } f(\overline{X}) = (x_1 - 10)^3 + (x_2 - 20)^3,$$

subject to nonlinear constraints:

$$\begin{aligned} c_1 &: \; (x_1 - 5)^2 + (x_2 - 5)^2 - 100 \geq 0, \\ c_2 &: \; -(x_1 - 6)^2 - (x_2 - 5)^2 + 82.81 \geq 0, \end{aligned}$$

and bounds:

$$13 \leq x_1 \leq 100 \text{ and } 0 \leq x_2 \leq 100.$$

The known global solution is $\overline{X}^* = (14.095, 0.84296)$, and $f(\overline{X}^*) = -6961.81381$ (see figure 2). The starting point, which is not feasible, is $\overline{X}_0 = (20.1, 5.84)$.

Genocop II approached the optimum very closely at the 12th iteration. The progress of the system is reported in the table 2.

4.4. Test Case #4

The problem[5] is

$$\text{minimize } f(x_1, x_2) = 0.01x_1^2 + x_2^2,$$

subject to nonlinear constraints:

$$\begin{aligned} c_1 &: \; x_1 x_2 - 25 \geq 0, \\ c_2 &: \; x_1^2 + x_2^2 - 25 \geq 0, \end{aligned}$$

Figure 2: A feasible space for test case #3.

and bounds:

$$2 \le x_1 \le 50 \text{ and } 0 \le x_2 \le 50.$$

The global solution is $\overline{X}^* = (\sqrt{250}, \sqrt{2.5}) = (15.811388, 1.581139)$, and $f(\overline{X}^*) = 5.0$. The starting point (not feasible) is $\overline{X}_0 = (2, 2)$.

It is interesting to note that the standard cooling scheme (i.e., $g(\tau, t) = 10^{-1} \cdot g(\tau, t-1)$) did not produce good results, however, when the cooling process was slowed down (i.e., $g(\tau, 0) = 5$ and $g(\tau, t) = 2^{-1} \cdot g(\tau, t-1)$), the system approached optimum easily (table 3). This, of course, leeds to some questions about how to control the temperature for a given problem: this is one of the topics for future research.

4.5. Test Case #5

The final test problem[14] is

$$\text{minimize } f(\overline{X}) = (x_1 - 2)^2 + (x_2 - 1)^2,$$

subject to a nonlinear constraint:

$$c_1 \; : \; -x_1^2 + x_2 \ge 0,$$

and a linear constraint:

$$x_1 + x_2 \le 2.$$

Iteration number	The best point	Active constraints
0	(20.1, 5.84)	c_1, c_2
1	(13.0, 0.0)	c_1, c_2
2	(13.63, 0.0)	c_1, c_2
3	(13.63, 0.0)	c_1, c_2
4	(13.73, 0.16)	c_1, c_2
5	(13.92, 0.50)	c_1, c_2
6	(14.05, 0.75)	c_1, c_2
7	(14.05, 0.76)	c_1, c_2
8	(14.05, 0.76)	c_1, c_2
9	(14.10, 0.87)	c_1, c_2
10	(14.10, 0.86)	c_1, c_2
11	(14.10, 0.85)	c_1, c_2
12	(14.098, 0.849)	c_1, c_2

Table 2: Progress of Genocop II on test case #3; for iteration 0 the best point is the starting point.

Iteration number	The best point	Active constraints
0	(2,2)	c_1, c_2
1	(3.884181, 3.854748)	c_1
2	(15.805878, 1.581057)	c_1
3	(15.811537, 1.580996)	c_1

Table 3: Progress of Genocop II on test case #4; for iteration 0 the best point is the starting point.

The global solution is $\overline{X}^* = (1,1)$ and $f(\overline{X}^*) = 1$. The starting (feasible) point is $\overline{X}_0 = (0,0)$.

Genocop II approached the optimum very closely at the 6th iteration. The progress of the system is reported in the table 4.

5. Conclusions

There are several interesting points connected with the above method. First, as with any other method based on genetic algorithms, it does not require any implicit (or explicit) calculations of gradient or Hessian matrix of the objective function and constraints. Consequently, the method does not suffer from the ill-conditioned Hessian problem usually associated with some calculus-based methods.

Any genetic algorithm can be used in place of Genocop I for the inner loop of

Iteration number	The best point	Active constraints
0	(0, 0)	c_1
1	(1.496072, 0.503928)	c_1
2	(1.020873, 0.979127)	c_1
3	(1.013524, 0.986476)	c_1
4	(1.002243, 0.997757)	c_1
5	(1.000217, 0.999442)	c_1
6	(1.000029, 0.999971)	c_1

Table 4: Progress of Genocop II on test case #5; for iteration 0 the best point is the starting point.

Genocop II. In such a case all constraints (linear and nonlinear) should be considered for placement in the set of active constraints A (the elements of L should be distributed between N_e and N_i). However, such method is much slower and less effective: for efficiency reasons, it is much better to process linear constraints separately (as it is done in Genocop I).

Further research includes experimenting with (1) problems of higher dimensions, (2) different cooling schemes and (3) different values of parameter δ which decides whether a constraint is active or not.

It might be worthwhile to search for other constraint handling paradigms as well. Recently, two interesting new approaches were proposed based on the behavioural memory paradigm and the re-mapping the fitness measures. The first method [22] acts in two stages: (1) a population evolves with some standard GA, where the fitness function is related to the constraint satisfaction, and (2) the final population from the previous stage (viewed as a memory containing some essential information about the constraints) is accepted as initial population for a GA with the objective function as fitness function, which is overridden by assigning zero fitness whenever the constraints are not satisfied. The second method [21] re-maps the fitness measures so that all feasible points have higher fitness than infeasible points (which ensures that feasible points are always preferred).

Also, it might be interesting to experiment with "adaptive penalty functions". After all, probabilities of applied operators might be adaptive (as in evolution strategies [23]); some initial experiments indicate that adaptive population sizes may have some merit [2]; so the idea of adaptive penalty functions may deserve some attention. In its simplest version, a penalty coefficient would be part of the solution vector and undergo all opertors (as opposed the idea of Genocop II, where such penalty coefficient is changed on regular basis).

Currently, a complete version of GENOCOP II is being implemented, which would include (as options) all above constraint handling methods. Interesting comparisons between these approaches should be reported shortly.

Acknowledgements:

The author wishes to thank David Fogel for his constructive comments.

References:

1. Aarts, E.H.L. and Korst, J., *Simulated Annealing and Boltzmann Machines*, Wiley, Chichester, UK, 1989.
2. Arabas, J., Michalewicz, Z., and Mulawka, J., *GAVaPS — a Genetic algorithm with Varying Population Size*, Technical Report 001-1994, Department of Computer Science, University of North Carolina, Charlotte, 1994.
3. Attia, N.F., *New Methods of Constrained Optimization using Penalty Functions*, Ph.D. Thesis, Essex University, England, 1985.
4. Belew, R. and Booker, L. (Editors), *Proceedings of the Fourth International Conference on Genetic Algorithms*, R.K. Belew and L. Booker (Eds.), Morgan Kaufmann Publishers, Los Altos, CA, 1991.
5. Betts, J.T., *An Accelarated Multiplier Method for Nonlinear Programming*, Journal of Optimization Theory and Applications, **21**, No.2, pp.137–174, 1977.
6. Biggs, M.C., *Constrained Minimization using Recursive Quadratic Programming: Some Alternative Subproblem Formulations*, Towards Global Optimization, L.C.W. Dixon and G.P. Szego (Eds.), North-Holand, 1975.
7. Broyden, C.G. and Attia, N.F., *A Smooth Sequential Penalty Function Method for Solving Nonlinear Programming Problem*, Lecture Notes in Control and Information Science 59, System Modelling and Optimization, Proceedings of the 11th IFIP Conference, July 1983. Springer-Verlag, Berlin and New York, 1983.
8. Broyden, C.G. and Attia, N.F., *Penalty Functions, Newton's Method, and Quadratic Programming*, Journal of Optimization Theory and Applications, **58**, No.3., 1988.
9. Fiacco, A.V., and McCormick, G.P., *Nonlinear Programming*, John Wiley, New York, 1968.
10. Fletcher, R., *Practical Methods of Optimization*, Vol.2, of *Constrained Optimization*, John Wiley and Sons, Chichester and New York, 1981.
11. Floudas, C.A. and Pardalos, P.M., *A Collection of Test Problems for Constrained Global Optimization Algorithms*, Springer-Verlag, Lecture Notes in Computer Science, **455**, 1987.
12. Floudas, C.A. and Pardalos, P.M., *Recent Advances in Global Optimization*, Princeton Series in Computer Science, Princeton University Press, Princeton, NJ, 1992.
13. Goldberg, D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, Reading, MA, 1989.
14. Hock, W. and Schittkowski K., *Test Examples for Nonlinear Programming Codes*, Lecture Notes in Economics and Mathematical Systems, Vol.187, Springer-Verlag, 1981.

15. McCormick, G.P., *Computability of Global Solutions to Factorable Nonconvex Programs*, Part I: Convex Underestimating Problems, Mathematical Programming, **10**, No.2 (1976), pp.147–175.
16. Michalewicz, Z. and Janikow, C., *Genetic Algorithms for Numerical Optimization*, Statistics and Computing, **1**, No.2, pp.75–91, 1991.
17. Michalewicz, Z. and Janikow, C., *Handling Constraints in Genetic Algorithms*, Proceedings of the Fourth International Conference on Genetic Algorithms, R.K. Belew and L. Booker (Eds.), Morgan Kaufmann Publishers, Los Altos, CA, 1991, pp.151–157.
18. Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, AI Series, New York, 1992.
19. Michalewicz, Z. and Swaminathan, S., *Evolutionary Operators for Continuous Convex Parameter Spaces*, in this volume.
20. Murray, W., *An Algorithm for Constrained Minimization*, Optimization, (Ed: R. Fletcher), pp.247–258, Academic Press, London and New York, 1969.
21. Powell, D. and Skolnick, M.M., *Using Genetic Algorithms in Engineering Design Optimization with Non-linear Constraints*, Proceedings of the Fifth International Conference on Genetic Algorithms, S. Forrest (Ed.), Morgan Kaufmann Publishers, Los Altos, CA, 1993, pp.424–430.
22. Schoenauer, M., and Xanthakis, S., *Constrained GA Optimization*, Proceedings of the Fifth International Conference on Genetic Algorithms, S. Forrest (Ed.), Morgan Kaufmann Publishers, Los Altos, CA, 1993, pp.573–580.
23. Schwefel, H.-P., *Numerical Optimization for Computer Models*, Wiley, Chichester, UK, 1981.