

Constraint Quadratic Approximation Operator for Treating Equality Constraints with Genetic Algorithms

Elizabeth F. Wanner, Frederico G. Guimarães, Rodney R. Saldanha
Department of Electrical Engineering
Universidade Federal de Minas Gerais
Belo Horizonte, Brazil
elizabeth@cpdee.ufmg.br

Ricardo H. C. Takahashi
Department of Mathematics
Universidade Federal de Minas Gerais
Belo Horizonte, Brazil
taka@mat.ufmg.br

Peter J. Fleming
Department of Automatic Control and
Systems Engineering
University of Sheffield
Sheffield, UK
p.fleming@sheffield.ac.uk

Abstract- This paper presents a new operator for genetic algorithms that enhances their convergence in the case of nonlinear problems with nonlinear equality constraints. The proposed operator, named CQA (Constraint Quadratic Approximation), can be interpreted as both a local search engine (that employs quadratic approximations of both objective and constraint functions for guessing a solution estimate) and a kind of elitism operator that plays the role of “fixing” the best estimate of the feasible set. The proposed operator has the advantage of not requiring any additional function evaluation per algorithm iteration, solely making use of the information that would be already obtained in the course of the usual Genetic Algorithm iterations. The test cases that were performed suggest that the new operator can enhance both the convergence speed (in terms of the number of function evaluations) and the accuracy of the final result.

1 Introduction

The most common way of incorporating constraints (both of inequality and equality types) into Genetic Algorithms has been the use of penalty functions [RPLH:89]. The constraints are placed into the objective function via a penalty parameter that measures violation of the constraints. The solution to the penalty problem can be made arbitrarily close to the optimal solution of the original problem by choosing a penalty parameter μ sufficiently large. Despite their simplicity, the penalty methods also have potential difficulties. If a very large μ is chosen to solve the penalty problem, some computational difficulties associated with ill-conditioning may appear. With a large μ , more emphasis is placed on feasibility and most procedures for unconstrained optimization will move quickly toward a feasible point. This point may be far from the constrained optimum, and premature termination can occur, especially in the presence of nonlinear equality constraints [BSS:79]. If μ is chosen too low, a lot of search time will be spent exploring the infeasible region, and termination will occur with large violation of the constraint.

Due to difficulties associated with the penalty methods, several researchers have developed alternative approaches to handle constraints [Coe:02], that go from variations of penalty functions (using, for instance, adaptive penalty factors) to more sophisticated methods such as dealing with

the constraints as if they were objectives of a multiobjective problem [FF:98]. These techniques, however, were developed mainly for the case of inequality constraints.

The equality constraints in Genetic Algorithms are difficult to manage. This is because equality constraints are more stringent than inequality ones, defining feasible sets of smaller dimension than the original optimization variable space. Genetic Algorithms, performing a search that is essentially based on sampling the full-dimensional space, are unlikely to find feasible solutions. We argue here that:

- The random nature of Genetic Algorithm operators is a fundamental feature that allows the search for optimal regions (basins of attraction) in the optimization space. These operators make the search become “spread” through the space, maximizing the chance of finding unknown basins. We call this the *volume search* property.
- This randomness of such operators, however, is conflicting with the need for performing a search in well-defined *zero-volume* objects, such as the feasible sets in equality-constrained problems. The *volume search* property will cause a kind of *random escape motion* from the feasible object, that will make the algorithm converge with slower rates.

In the case of unconstrained problems, this problem also occurs, leading to a *random escape motion* from the optimal point. However, for dealing with this problem, there is a well established procedure: the usage of an *elitism* operator. This operator avoids the *escape motion* of the best point obtained up to the current iteration.

In this paper, we devise a novel local search operator to tackle equality constraints in optimization problems. The main idea is to constrain the Genetic Algorithm to find solutions that are inside an object that has the same dimensionality as the feasible set. This object is a second-order approximation of the feasible surface. In this way, the optimization algorithm is divided into a two-stage methodology: (i) the second order approximation is updated in each generation, and (ii) the evolutionary search is performed such that some solutions are taken inside this approximated surface. An operator that performs such steps is included in the Genetic Algorithm, in this way opposing the *random escape motion* effect.

We suggest that this new operator, named as the *Constraint Quadratic Approximation* (CQA), can be interpreted

as an *elitism operator* specialized for enhancing, iteration after iteration, the estimate of the feasible set. The conventional elitism operator, that keeps a single point (or a set of points) in each iteration, does not apply to the search for the feasible set, since this object is an m -dimensional surface, and not a set of discrete points. The proposed CQA operator provides an approximation for this object in this sense: the best second-order surface that approximates the feasible set is kept and enhanced, as the algorithm evolves.

This operator can be used within the standard iterative cycle of Genetic Algorithms (GAs) in order to improve their convergence. In this manner, the proposed algorithm is in fact a special instance of a Memetic Algorithm (MA) or a Hybrid Genetic Algorithm (HGA).

The main advantage of the proposed CQA operator is that it does not require additional function evaluations in its operation. It employs only information already acquired by the usual execution of the GA. In the case when the objective function and/or the constraints are very expensive, from a computational point of view, this is a great advantage, since the additional computational cost required by the local operator is negligible.

The proposed CQA operator was coupled to the Real-Biased Genetic Algorithm (RBGA) [RSTM:03] to create the novel hybrid algorithm, RBGA-CQA Hybrid. The RBGA-CQA Hybrid was tested with some analytical problems and was compared with the RBGA algorithm. The obtained results support the conclusion that the hybrid algorithm can efficiently deal with nonlinear equality constrained problems.

2 Quadratic Approximations

We consider the nonlinear equality constrained problem of the form:

$$\begin{aligned} x^* &= \arg \min_x f(x) \\ \text{subject to: } g(x) &= 0 \end{aligned} \quad (1)$$

where f and g are real-valued nonlinear functions. We propose a methodology to solve the problem (1) using a quadratic approximation for both the objective and constraint functions.

Suppose that f and g can be locally approximated by quadratic functions of the form:

$$f(x) = (x - x_f)^T . H . (x - x_f) \quad (2)$$

and

$$g(x) = (x - x_g)^T . G . (x - x_g) - 1 \quad (3)$$

where x_f is the optimal point of the objective function approximation and x_g is the minimum point of the equality constraint function approximation. With these approximations, we have the essential information that is to be used into the methodology: the Hessian matrix and the unconstrained minima points of f and g .

A possible way to get the Hessian matrix of these functions would be to build an approximated matrix B_k through a *quasi-Newton* update formula [BSS:79]. However, in

quasi-Newton formulae, we have to evaluate the gradient vector of the function. It is sometimes inconvenient, difficult or impossible to calculate the derivatives of a function. Besides, the gradient evaluation would put a severe handicap on the new method, since the evaluation would require considerable computational effort.

Alternatively, with the goal of avoiding high computational costs, we use the current population and the available function evaluations for the population of the traditional Genetic Algorithm to fit a quadratic approximated function. The methodology applied is described as follows.

Let h be a real-valued function. Given distinct points z_1, z_2, \dots, z_n , we consider the problem of finding a quadratic real-valued function f such that

$$h(z_i) = f(z_i) \quad (4)$$

for $i = 1, 2, \dots, N$ where N is the number of available points.

The function f being quadratic, we can write it as

$$f(z) = z^T . H . z + r^T z + \gamma \quad (5)$$

for some suitable symmetric $n \times n$ matrix H , $n \times 1$ vector r and some scalar γ . Hence, the problem of finding f such that (4) holds can be restated as to find H , r and γ such that

$$z_i^T . Q . z_i + r^T z_i + \gamma = h(z_i) \quad (6)$$

for $i = 1, 2, \dots, N$.

This is a linear system of N equations in the unknown entries of H , r and γ . The number of unknowns in H is equal to $n + \frac{n^2 - n}{2}$, hence the total number of unknowns is given by

$$n + \frac{n^2 - n}{2} + n + 1 = \frac{(n + 1)(n + 2)}{2} \quad (7)$$

If

$$z_i^T . H . z_i + r^T z_i + \gamma = 0 \Rightarrow H = 0, r = 0, \gamma = 0 \quad (8)$$

for $i = 1, 2, \dots, N$, and $N = \frac{(n+1)(n+2)}{2}$ then there exists a unique quadratic function f such that (4) holds. This is an *interpolation case*. When $N > \frac{(n+1)(n+2)}{2}$, the linear system (6) is over-determined and we can find a least norm solution:

$$\min_{H, r, \gamma} \left\| \sum_i E_i \right\| \quad (9)$$

where E_i is the residual error given by

$$E_i = f(z_i) - h(z_i) \quad (10)$$

for $i = 1, 2, \dots, N$. If the norm is the Euclidean norm, then the function f is the quadratic least squares approximation. To obtain the unknown entries of H , r and γ we solve the over-determined linear system (6) through the Singular Value Decomposition (SVD) procedure.

Once we have obtained a quadratic approximation

$$f(z) = z^T . H . z + r^T z + \gamma \quad (11)$$

we can easily find the minimum point of f

$$z_f = -\frac{1}{2}H^{-1}r \quad (12)$$

and then we can rewrite the analytical expression of the function

$$f(z) = (z - z_f)^T.H.(z - z_f). \quad (13)$$

3 The Bisection Method for Solving the Approximated Optimization

With the local quadratic approximations

$$f(x) = (x - x_f)^T.H.(x - x_f) \quad (14)$$

and

$$g(x) = (x - x_g)^T.G.(x - x_g) - 1 \quad (15)$$

we can estimate a solution for (1) using the bisection method. In this section we present the mathematical formulation that permits the utilization of the bisection to solve (1).

By using a suitable change of coordinates, we can write the associated problem for the problem (1):

$$\begin{aligned} \min(z - z_f)^T.(z - z_f) \\ \text{subject to: } (z - z_g)^T.Q.(z - z_g) - 1 = 0 \end{aligned} \quad (16)$$

where z_f is the unconstrained minimum of the new objective function, z_g the center point of the quadratic constraint and Q the Hessian matrix associated with the constraint.

We know that

$$\nabla f(z) = 2(z - z_f) \quad (17)$$

is the analytical form of the gradient vector of f on the point z where z_f is the minimum point of f and

$$\nabla g(z) = 2Q(z - z_g) \quad (18)$$

is the analytical form of the gradient vector of the approximated constraint on z where z_g is the minimum point of the approximated constraint function.

Besides, through the Kuhn-Tucker conditions, we have

$$\nabla f = -\lambda.\nabla g \quad (19)$$

Then, using (17), (18) and (19), we can write

$$z - z_f = -\lambda.Q.(z - z_g) \quad (20)$$

and, finally,

$$(I + \lambda.Q).z = z_f + \lambda.Q.z_g \quad (21)$$

We can see that the equation (21) gives a parametric path joining z_f to z_g , where the gradient vectors of f and g are

linearly dependent. We search for the value, z^* , that is on the path and on the constraint

$$(z - z_g)^T.Q.(z - z_g) - 1 = 0 \quad (22)$$

This value is found through a bisection method.

The idea behind the bisection method is very simple. Over some interval the function is known to pass through zero because its sign changes. Evaluate the function at the interval's midpoint and examine its sign. Use the midpoint to replace whichever limit has the same sign. On the parametric path joining z_f and z_g , all values of z that are inside the interval $[z_f, z^*]$ have $g(z) = (z - z_g)^T.Q.(z - z_g) - 1 > 0$. In the same way, all z that are inside the interval $[z^*, z_g]$ have $g(z) = (z - z_g)^T.Q.(z - z_g) - 1 < 0$. Therefore, given any interval on this path, it is possible to ensure the existence of the point z^* which satisfies $g(z^*) = 0$.

Below we can see an algorithm based on the bisection method that can find this point z^* for a given precision ϵ .

The Bisection Method

Step 1. Specify $l_a \in l_b$;

Step 2. $x_a \leftarrow x(l_a)$;

Step 3. $x_b \leftarrow x(l_b)$;

Step 4. $f l_a \leftarrow f(x_a)$;

Step 5. $f l_b \leftarrow f(x_b)$;

Step 6. **while** non stop criterion

Step 6.1. $M \leftarrow f l_b$;

Step 6.2. $l_k \leftarrow \frac{l_a + l_b}{2}$

Step 6.3. $x_k \leftarrow x(l_k)$

Step 6.4. $f_k \leftarrow f(x_k)$

Step 6.5. **if** $M.f_k > 0$

Step 6.5.1. $l_b \leftarrow l_k$

Step 6.5.2. $x_b \leftarrow x_k$

Step 6.5.3. $f l_b \leftarrow f_k$

else

Step 6.5.1. $l_a \leftarrow l_k$

Step 6.5.2. $x_a \leftarrow x_k$

Step 6.5.3. $f l_a \leftarrow f_k$

If after n iterations the solution is known to be within an interval of size ϵ_n , then:

$$\epsilon_{n+1} = \frac{\epsilon_n}{2} \quad (23)$$

Thus, we know in advance the number of iterations that are required to achieve a given tolerance in the solution

$$n = \log_2 \frac{\epsilon_0}{\epsilon} \quad (24)$$

At the end of the bisection procedure, we must change the coordinates again, returning to the original problem coordinates of (1), in which the objective function is evaluated.

4 Hybridizing the Local Operator with GAs

In order to implement the proposed ideas, we use a version of a Genetic Algorithm that has been tested already in some related problems in the literature: the Real-Biased Genetic Algorithm (RBGA). The proposed new operator is only included in the pre-existing GA.

4.1 The Real-Biased GA

The Real-Biased Genetic Algorithm (RBGA) [RSTM:03] is a Genetic Algorithm with a different feature: the real biased crossover operator. The RBGA is defined as the successive application of the following operations: population evaluation and the fitness function computation; selection by roulette; real biased crossover; mutation; elitism.

The real biased crossover is defined as follows.

- The population with N individuals is randomly ordered in $\frac{N}{2}$ pairs of individuals. For each pair, the crossover will occur with probability pc .
- For each pair subjected to crossover, the fitness function $J(x)$ of the individuals is considered. The individuals which are n -vectors of real parameters are labeled x_1 and x_2 , such that $J(x_2) < J(x_1)$.
- The real biased crossover generates one offspring individual x_g as

$$x_g = \alpha \cdot x_1 + (1 - \alpha) \cdot x_2 \quad (25)$$

with α chosen in the interval $[-\xi; 1 + \xi]$, with ξ being an extrapolation factor selected by the user from the interval $[0; 1]$. α is selected according to the probability distribution defined by

$$\alpha = (1 + 2 \cdot \xi) \beta_1 \beta_2 - \xi \quad (26)$$

where β_1 and β_2 are random variables with uniform probability distribution inside the domain $[0; 1]$. These provide a quadratic probability distribution for α which makes the new individual x_g have a greater probability of being closer to x_1 (the best parent individual) than to x_2 (the worst parent individual).

- The other offspring is chosen without bias, i.e. α is chosen in the interval $[-\xi; 1 + \xi]$ with uniform probability.

The specific evaluation of the effect of the real biased crossover operator can be found in [TVRK:03].

In the case of one individual being out of the admissible range, the reflection method is applied to force the individual back inside the feasible region. For a reflection by the lower limit (x_L) the operation is defined as

$$x_r = x_L + |x - x_L| \quad (27)$$

where x is the individual outside of the admissible range and x_r represents the resulting individual after the reflection. For a reflection by the upper limit (x_U) the operation is defined in an analogous way as

$$x_r = x_U - |x_U - x| \quad (28)$$

with the same meaning for the other variables.

The mutation operator is defined as follows. Each individual in the population can be subjected to mutation, with probability of 0.03. If an individual x suffers mutation, the resulting individual x_m is defined by

$$x_m = x + \delta \quad (29)$$

with

$$\delta_i = 0.05 \beta_i (x_r)_i \quad (30)$$

where β_i is a random number with Gaussian distribution, zero mean and variance equal to one, and x_r is a range vector with lower and upper limits given by x_L and x_U respectively. The other operations in the RBGA are as usual.

The real-biased crossover resembles a search following the direction of a “tendency” with an information that is similar to the one that is given by a kind of “gradient vector” evaluation, but with possibly “long-range information” validity (instead of the only local validity that is associated with a gradient). This operator only evaluates the objective function without any calculation of function derivatives, in the same way as the GA. If the parents are located near each other, then a step that implicitly uses a directional derivative information is executed. This procedure speeds up the local convergence to the optimal point. If the parents are far away from each other (maybe in different attraction basins), this procedure can be interpreted as a “long-range trend” information. The offspring individual is created taking into account such information.

4.2 The RBGA-CQA Hybrid

A difference between global and local search procedures is that global techniques are largely independent on the initial conditions while local methods produce solutions that are strongly dependent on the starting point. Besides, local procedures tend to be coupled to the solution domain. The Genetic Algorithms were largely developed for the purpose of performing global searches. The conventional *genetic operators* were developed with the main purpose of enhancing the algorithm capability of finding global optima basins.

Memetic Algorithms (MA) or *Hybrid Genetic Algorithm* (HGA) denote the association of local and global search operators inside GA. This kind of strategy is used by many successful global optimization procedures with the goal of refining the solution of the problem and improving the speed of convergence to the actual optimum point (not only to its vicinity).

Richard Dawkins in his book *The selfish gene* has introduced the word *meme* to denote the idea of a unit of imitation in cultural transmission which in some aspects is analogous to the gene [Daw:76]. The first use of the name

Memetic Algorithm in the literature appeared in 1989 in the work of Pablo Moscato [Mos:89] *On Evolution, Search, Optimization, Genetic Algorithm and Martial Arts: Towards Memetic Algorithms*, for denoting algorithms that use some kind of structured information, that is obtained and refined as the algorithm evolves, and is “transmitted” from one generation to the another, for enhancing the search. Since then, this idea has gained wide acceptance in the computing community and has been successfully applied in a large class of problems [IM:96, Mos:99, LHKM:04, KC:04].

The main advantage obtained from the use of Memetic Algorithms is that the space of possible solutions is reduced to a “subspace” (or a lower-order set) of local optima. The introduction of local search in the traditional genetic algorithms has some computational cost, but this is compensated by the decrease in the search space that must be explored in order to find the solution.

The new methodology, RBGA-CQA Hybrid (RBGA with Constraint Quadratic Approximation), presented in this paper, deals with nonlinear equality constraint using the quadratic approximation technique and the bisection method as a local search operator in the real-biased Genetic Algorithm. This new local search operator improves the solution because it allows the equality constraint to be reached with increased precision. Furthermore, notice that this operator does not impose any significant additional computational cost on the traditional RBGA if the computational cost is measured by the number of calls of the objective and constraint functions. In this way, the new algorithm is a kind of Memetic Algorithm or Hybrid Genetic Algorithm for equality constrained problems. We show below the basic sketch of a Memetic Algorithm:

Step 1. Initialize parameters

Step 2. Initialize population

Step 3. WHILE no stop criterion

- selection
- crossover
- mutation
- local search

end

The RBGA-CQA fits in this sketch by using the quadratic approximation method as an operator for improving one or more individuals, providing solutions that satisfy the equality constraint with higher precision.

An actual implementation of the proposed scheme could be performed in several different ways. With the main purpose of showing the relative enhancement of solution that the proposed CQA operator can provide, we used a rather arbitrary choice of specific definitions that define how that operator links the pre-existing RBGA algorithm. For hybridizing the RBGA with the local search operator, we have established the following definitions:

- The RBGA is executed for the optimization of a modified objective function with a penalty term that takes into account the equality constraint:

$$F(x) = f(x) + \beta|g(x)|$$

- The local search CQA operator will be run every 5 generations. We performed experiments on executing the operator every 2, 5 and 10 generations, but no considerable variation in the results was perceived.

- As this operator is a local search one, only points in a neighborhood of the current best point will be used to build the quadratic approximations. This neighborhood is an ellipse whose axes correspond to 10 percent of the length of each parameter range. As a mathematical condition, the number of points inside this neighborhood must be higher or equal to

$$\frac{(n+1)(n+2)}{2}$$

where n is the problem dimension. The higher the number of points inside this neighborhood, the more accurate the quadratic approximation.

- Finally, the output point of the new operator will deterministically replace the worst point of the current population.

Notice that the output point also can be used as an additional stop criterion for the algorithm: the stabilization of such point can be interpreted as the algorithm finding the solution.

5 Results

The RBGA and RBGA-CQA-Hybrid algorithms have been tested with a set of analytical problems. The analytical test problems were chosen with different characteristics and degrees of difficulties. The problems are:

- Quadratic Problem:

$$x^* = \arg \min_x x_1^2 + x_2^2 + x_3^2$$

$$\text{subject to: } (x_1 - 2)^2 + (x_2 - 1)^2 + 4(x_3 - 1)^2 = 1 \quad (31)$$

- Non-quadratic Problem:

$$x^* = \arg \min_x (x_1 - 2)^4 + (x_1 - 2x_2)^2$$

$$\text{subject to: } x_1^2 - x_2 = 0 \quad (32)$$

- Multimodal Problem:

$$x^* = \arg \min_x x^T . A^T . A . x - 10[11] \cos(2\pi Ax)$$

$$\text{subject to: } (x_1 - 2)^2 + (x_2 - 2)^2 = 1 \quad (33)$$

where

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix}$$

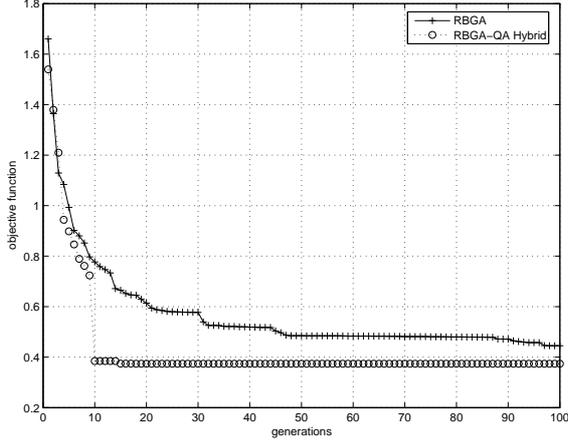


Figure 1: Convergence Line for the Quadratic Problem

In both algorithms, the equality constraint $g(x) = 0$ is replaced by two inequality constraints of the form

$$\begin{aligned} g_1(x) &= -g(x) - \epsilon \leq 0 \\ g_2(x) &= g(x) - \epsilon \leq 0 \end{aligned}$$

with $\epsilon = 0.001$. If any inequality constraint g_i is violated then the modified objective function is given by

$$f = f + 10 \cdot g_i(x).$$

Each algorithm was executed 30 times for each function and was started with the same basic parameters as listed below:

- Population size: 20 individuals
- Recombination Probability: 0.6
- Polarization Probability: 0.3
- Mutation Probability: 0.02
- Mutation Size: 0.05
- Dispersion Factor in Fitness Function: 1.8
- Extrapolation Factor in Recombination: 0.2
- Maximum Number of Generations: 100

For a discussion of the effect of these parameters over the RBGA performance, see [TVRK:03]. Since our objective is only coupling the CQA operator with an efficient and a previously tested Genetic Algorithm, we do not focus on this subject. An exhaustive investigation on this issue will be carried out and a parameter sensitivity analysis will be addressed in a future work.

The maximum number of generations was the only stopping criterion in both methods. At the end of 30 executions of RBGA and RBGA-CQA Hybrid, we obtained the mean convergence line which corresponds to the mean value of the best individual throughout the 100 generations.

Figures 1,2 and 3 show the convergence line for each problem. In the graphs, the x-axes represent the generation and the y-axes represent the base 10 logarithm of the objective function value of the best individual. The base 10 logarithm was used, except in the multimodal problem, only to enforce the difference between the lines.

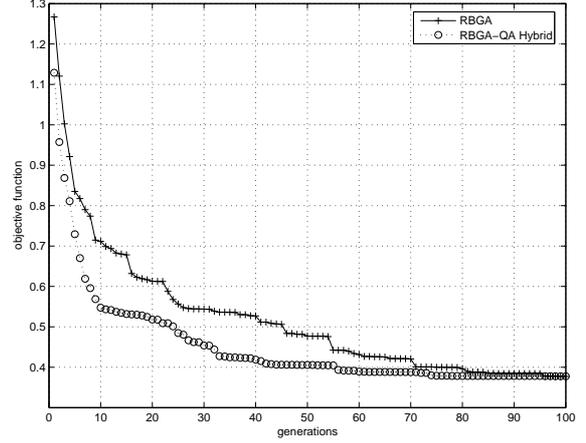


Figure 2: Convergence Line for the Non-Quadratic Problem

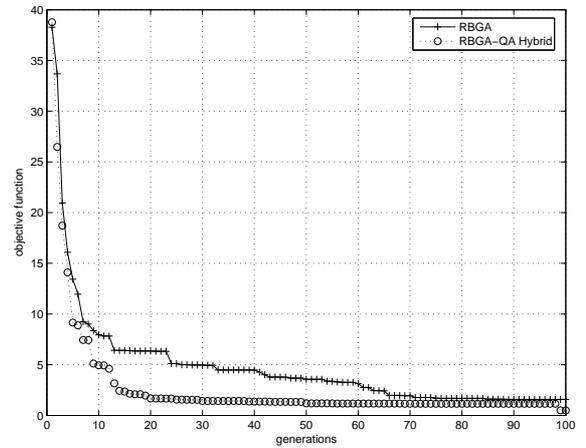


Figure 3: Convergence Line for the Multimodal Problem

In figure 1, we can see that the RBGA-CQA Hybrid has a higher speed of convergence compared with the RBGA. Around the 10th generation, the RBGA-CQA Hybrid found a solution which is a better value than the final solution found by the RBGA. A similar result can be seen when we observe the convergence line for figure 3. For the non-quadratic problem (fig.2), the convergence speed of the RBGA-CQA Hybrid is also higher than that of the RBGA.

5.1 Diversity Analysis

Given that the proposed local search, CQA operator, is substituting the worst individual by an improved one, an important issue that arises is the effect of this operator on the diversity of the population.

In order to perform a diversity analysis, we carried out the following experiment:

1. Define a radius r in the search space;
2. Execute the GA, returning the final population;
3. Place a sphere of radius r centered at the optimal point;
4. Initialize a sphere count as equal to one;

5. Exclude all individuals that are located within this sphere;
6. Among the remaining individuals, center the sphere at the individual that is closest to the previous center and increment the sphere counter;
7. Go to step 5 until there is no more remaining points in the population.

Through this experiment, it is possible to identify the population with greater diversity, which is simply that with more spheres. However, in order to obtain a relevant result, we need to perform this experiment over many executions of each algorithm and take the mean values. We executed the RBGA and RBGA-CQA, with the same basic parameters listed in the beginning of this section, 30 times over the following test problems:

- 2D Rotated Rastrigin

$$\begin{aligned}
 x^* &= \arg \min_x x^T \cdot A^T \cdot A \cdot x - 10[11] \cos(2\pi Ax) \\
 \text{subject to: } &(x_1 - 0.9)^2 + (x_2 - 0.9)^2 = 1
 \end{aligned} \tag{34}$$

where

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$$

- 3D Rastrigin

$$\begin{aligned}
 x^* &= \arg \min_x x^T \cdot x - 10[111] \cos(2\pi x) \\
 \text{subject to: } &(x_1 - 0.65)^2 + (x_2 - 0.65)^2 + (x_3 - 0.65)^2 = 1
 \end{aligned} \tag{35}$$

For the sphere radius, we set $r = 0.0001$. The final results are shown in Table 1. Figures 4 and 5 show the diversity analysis as described above, every 5 generations, for the problems 34 and 35.

Problem	RBGA	RBGA-CQA Hybrid
Rastrigin 2D	7.4333	6.4667
Rastrigin 3D	8.3333	7.6000

Table 1: Mean values of the number of spheres in the diversity analysis

As we can observe, in fact, the CQA operator implies a small reduction in the diversity of the population. This is expected, since the improvement of the best individual increases the possibility of extinction of the worst ones.

6 Conclusions

The RBGA-CQA Hybrid has presented a good performance in the equality constrained problem tests. The usage of quadratic approximations for both objective function and constraint and a kind of elitism feature in the proposed CQA operator enhance the convergence properties of the RBGA. The results confirm that the RBGA-CQA Hybrid converge to better solutions, while needing less function evaluations.

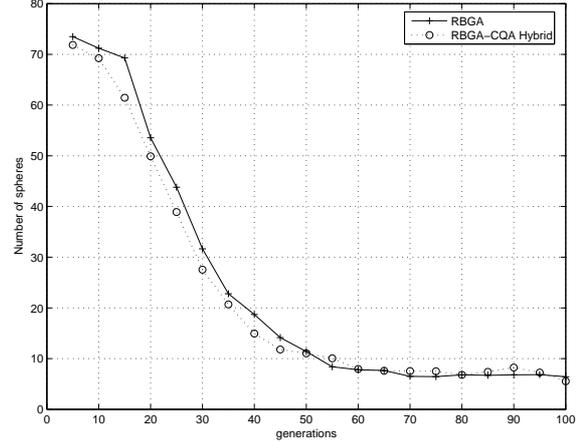


Figure 4: Number of spheres every 5 generations for the 2-D Rotated Rastrigin

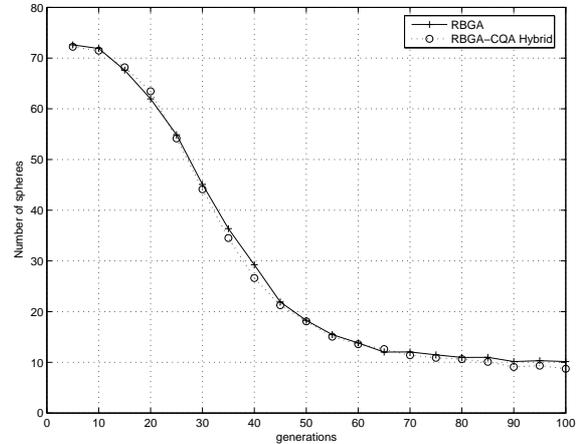


Figure 5: Number of spheres every 5 generations for the 3-D Rastrigin

The CQA operator does not require any additional function evaluation per algorithm iteration, which allows us to recommend that this operator should be included in any GA that is to be used in problems with equality constraints. This is especially emphasized in the case of problems with expensive-to-evaluate objective functions or constraints.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments. E.F.W. was supported by CAPES, Brazil and R.H.C.T, R.R.S. and F.G.G were supported by CNPQ, Brazil.

Bibliography

- [BSS:79] Bazaraa, M.S. and Sherali, H.D. and Shetty, C.M.(1979) "Nonlinear Programming: Theory and Algorithms," John Wiley and sons, Inc.
- [Coe:02] Coello Coello, C.A. "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*, 2002, vol 191, p 1245-1287.

- [Daw:76] Dawkins, R. (1976) "The selfish gene," Oxford Press University, Oxford.
- [FF:98] Fonseca, C. and Fleming, P. (1998) "Multiobjective optimization and multiple constraint handling with evolutionary algorithms I: a unified formulation," *IEEE Transactions on Systems, Man and Cybernetics - Part A*, 1998, vol 28(1), p 26-37.
- [IM:96] Ishibuchi, H. and Murata, T., "Multi-objective genetic local search algorithm," *Proceedings of the IEEE International Conference on Evolutionary Computation*, p. 119-124, May 20-22, 1996.
- [KC:04] Knowles, J. and Corne, D., "Memetic algorithms for multiobjective optimization: issues, methods and prospects," chapter in *Recent Advances in Memetic Algorithms*, Krasnogor, N., Smith, J.E., and Hart, W.E. (eds). pp. 313-352. Springer, 2004.
- [LHKM:04] Lozano, M. and Herrera, F. and Krasnogor, N. and Molina, D., "Real-coded memetic algorithms with crossover hill-climbing," *Evolutionary Computation Journal*, vol. 12, no. 3, p. 273-302, 2004.
- [MS:96] Michalewicz, Z. and Schoenauer, M. "Evolutionary algorithms for constrained parameter optimization problem," *Evolutionary Computation*, 1996.
- [Mos:89] Moscato, P. "On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms," *Caltech Concurrent Computation Program*, C3P Report 826, 1989.
- [Mos:99] Moscato, P., "Memetic algorithms: a short introduction," chapter in *New Ideas in Optimization*, D. Corne, M. Dorigo, F. Glover, McGraw-Hill, New York, 1999., pp. 219-234.
- [RSTM:03] Ramos, R.M. and Saldanha, R.R. and Takahashi, R.H.C. and Moreira, F.J.S. "The real-biased multiobjective genetic algorithm and its application to design of wire antennas," *IEEE Transactions on Magnetics*, 2003, vol 39(3), p 1329-1332.
- [RPLH:89] Richardson, J.T. and Palmer, M.R. and Liepins, G. and Hilliard, M. "Some guidelines for genetic algorithm with penalty function," *Proceedings of the Third International Conference on Genetic Algorithms*, 1989, p 191-197, Morgan Kaufmann.
- [TVRK:03] Takahashi, R.H.C. and Vasconcelos, J.A. and Ramirez, J.A. and Krahenbuhl, L. "A multiobjective methodology for evaluation genetic operators," *IEEE Transactions on Magnetics*, 2003, vol 39, p 1321-1324.