Projection-Based Local Search Operator for Multiple Equality Constraints within Genetic Algorithms

Gustavo Peconick, Elizabeth F. Wanner (Member, IEEE), Ricardo H. C. Takahashi (Member, IEEE)

Abstract—This paper presents a new operator for genetic algorithms that enhances convergence in the case of multiple nonlinear equality constraints. The proposed operator, named CQA-MEC (Constraint Quadratic Approximation for Multiple Equality Constraints), performs the steps: (i) the approximation of the non-linear constraints via quadratic functions; (ii) the determination of exact equality-constrained projections of some points onto the approximated constraint surface, via an iterative projection algorithm; and (iii) the re-insertion of the constraint-satisfying points in the genetic algorithm population. This operator can be interpreted both as a local search engine (that employs local approximations of constraint functions for correcting the feasibility) and a kind of elitism operator for equality constrained problems that plays the role of “fixing” the best estimates of the feasible set. The proposed operator has the advantage of not requiring any additional function evaluation per algorithm iteration, solely making usage of the information that is 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.

I. INTRODUCTION

The most common way of incorporating constraints (both of inequality and equality types) into genetic algorithms has been the use of penalty functions [1]. 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 $\mu$ sufficiently large. Despite their simplicity, the penalty methods also have potential difficulties. If a very large $\mu$ is chosen to solve the penalty problem, some computational difficulties associated with ill-conditioning may appear. With a large $\mu$, 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 [2]. If $\mu$ 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 [3], 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 [4]. 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. The 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 [5], the authors have presented a methodology for solving mono-objective problems with only one equality constraint. The idea was to perform a quadratic approximation of both the objective function and the constraint function, followed by the application of the Karush-Kuhn-Tucker condition in the auxiliary quadratic problem for finding an estimate of the constrained optimum. That idea, although leading to good computational results, was not applicable to the case of several equality constraints.

In this paper, we devise a novel methodology to tackle multiple equality constraints. This methodology also uses quadratic approximations for the equality constraints. A quadratic approximation is the simplest nonlinear model, thus needing less information to be generated while keeping a good description of the local behavior of the function. The
procedure here uses only the constraint function samples that have already been evaluated by the usual steps of the evolutionary algorithm. These samples are employed for constructing quadratic function approximations of the constraint non-linear functions. In this work, these quadratic approximations define an associated local search problem that is solved using a semi-analytical formulation (a formulation that can be solved exactly and reliably via a low-cost numerical method). The key novelty that allows dealing with several equality constraints is an iterative projection algorithm that finds points that satisfy the approximated quadratic constraints with low computational overhead. Different constraint-satisfying solutions can be found for different local quadratic approximations that describe the constraint functions nearby different points. Such high-quality constraint-satisfying solutions, being re-introduced in the GA population, lead the algorithm to a faster convergence, and also allow a much tighter satisfaction of problem constraints. An operator with these features 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.

Previous works have employed the strategy of using local approximations to enhance the performance of stochastic methods (none of them for equality-constrained problems, however). In [6], the authors use local approximations, including quadratic ones, to estimate the fitness of the offspring population, in order to reduce the cost of the evolutionary process. The quadratic approximation is a very popular response surface methodology [7], [8], and is also used in electromagnetic design [9], [10]. In [7], the quadratic approximations are used to improve genetic operators. Some main differences of the approach proposed in this paper, in relation to these previous works, are:

- There is no need of taking additional function samples. The only information that is employed for building the quadratic approximations is the sample set that is already available from the normal iterations of the genetic algorithm; not requiring additional function evaluations in its operation. In the case when 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 constraint-satisfying estimates are re-introduced in the genetic algorithm population for being iterated inside the usual genetic algorithm. There is no step in which the population is globally driven by the approximations. This makes the proposed approach more robust against eventual approximation errors.

Such features have already been successfully incorporated, using local quadratic approximations as a specialized local search operator, in [11], [12], for dealing with inequality-constrained problems.

The proposed CQA operator was coupled to the Real-Biased Genetic Algorithm (RBGA) [14], [15], for evaluating its effect. The RBGA-CQA-MEC Hybrid was tested with some analytical problems and was compared with the RBGA algorithm. The results that are obtained support the conclusion that the hybrid algorithm can efficiently deal with nonlinear equality constrained problems.

This paper is organized as following. The local search operator is detailed in Section II. Section III describes its hybridization with GAs. Section IV presents the results and the conclusions are given in Section V.

II. LOCAL SEARCH OPERATOR

We consider the nonlinear equality constrained problem of the form:

$$x^* = \arg \min_x f(x)$$

subject to: $g_i(x) = 0$

$$i = 1, \ldots, k$$

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

The local search operator transforms the constraints of the original problem (1) into local quadratic constraints.

In the first place, we need to show how to construct the quadratic approximation model for each function $g_i$, for $i = 1, \ldots, k$.

A. Quadratic Approximations

If a point, $x_n$, is selected for the local search phase, we select all the points inside a neighborhood of $x_n$, defined by

$$N(x_n) = \{ x : (x - x_n)^T R (x - x_n) \leq 1 \}$$

and

$$R_{ij} = \begin{cases} 0.1(L_i^+ - L_i^-) \gamma_1, & i = j \\ 0, & i \neq j \end{cases}$$

where $L_i^-$ and $L_i^+$ are, respectively, the minimum and maximum value for the $i$th variable.

Let $g$ be a real-valued function. Given the distinct points $z_1, z_2, \cdots, z_n$ inside the neighborhood, we consider the problem of finding a convex quadratic real-valued function

$$\hat{g}(z) = z^T H z + r^T z + \gamma$$

for some suitable symmetric $n \times n$ matrix $H$, $n \times 1$ vector $r$ and some scalar $\gamma$, such that

$$g(z_i) \equiv \hat{g}(z_i)$$
for \( i = 1, 2, \cdots, N \) where \( N \) is the number of available points.

Hence, the problem of finding \( \hat{g} \) such that (5) holds can be restated as to find \( H, r \) and \( \gamma \) such that

\[
\begin{align*}
E_i &= \hat{g}(z_i) - g(z_i) \\
E_i &= z_i^T H z_i + r^T z_i + \gamma - g(z_i)
\end{align*}
\]

for \( i = 1, 2, \cdots, N \).

This is a linear system of \( N \) equations in the unknown entries of \( H, r \) and \( \gamma \). The total number of unknowns is given by \( \frac{(n+1)(n+2)}{2} \).

When \( N > \frac{(n+1)(n+2)}{2} \), the linear system (6) is overdetermined and we can find a least norm solution:

\[
\min_{H, r, \gamma} \| E \|.
\]

If the norm is the Euclidean norm, then the function \( \hat{g} \) is the quadratic least squares approximation.

In this way, \( \hat{g} \) can be found by solving

\[
\min \left( t : \sqrt{\sum_{i} E_i^2} \leq t, H > 0 \right)
\]

where \( H > 0 \) means that the matrix \( H \) is semidefinite positive. Since the constraint is a Lorentz cone constraint, the semidefinite problem (8) can be efficiently solved using SeDuMi [13].

For more detailed information of this procedure, see [5]. It is worthwhile to notice that the construction of the quadratic models does not impose any additional extra function evaluation since the process uses the available samples gathered during the optimization process.

### B. Iterative Projection Algorithm

Consider the local quadratic approximations

\[
\hat{g}_i(x) = (x - x_{o_i})^T G_i (x - x_{o_i}) - 1
\]

for \( i = 1, \cdots, k \). The constraint functions have been normalized in (9). The aim here is to estimate constraint-satisfying solutions for (1), replacing the original constraint functions by the approximated ones. This subsection is devoted to solving this auxiliary approximated problem.

Consider \( z_0 \in \mathbb{R}^n \) being the point selected for starting the local search phase. Consider, without loss of generality, the first quadratic equality constraint, \( \hat{g}_1 \). We perform the following steps:

1) Set the counters \( j = 0 \) and \( i = 1 \).

2) Find the coordinate change \( \mathcal{L}_1 \) which makes \( x_{o_1} \) becomes located at the origin, and in which the ellipsoid defined by \( \hat{g}_1(x) = 0 \) becomes a sphere with unitary radius, for all \( i = 1, \cdots, k \).

3) While not (stop criterion)
   a) Find \( w_j \) such that
   \[
   w_j = \frac{\mathcal{L}_1(z_j)}{\| \mathcal{L}_1(z_j) \|}.
   \]

   b) Find now \( z_{j+1} \) as:

\[
z_{j+1} = \mathcal{L}_1^{-1}(w_j).
\]

   c) Update the counters: \( j = j + 1, \) and \( i = i + 1; \) if \( i > k \), make \( i = 1 \).

Each point \( z_{j+1} \) corresponds to a projection of \( z_j \) onto the \( i \)-th constraint surface. Applying this procedure for all quadratic equality constraints in sequence and after several iterations, the point selected for the local search phase \( x_{s} \) will move to a point which satisfies all quadratic equality constraints simultaneously.

Figure 1 shows, in a few iterations, a sequence of points that approximates to the point that is on all the equality constraints, in a bi-dimensional example. This very simple procedure can converge or not, depending on the relative position of \( z_0 \) in relation to the constraint surfaces. It should be noticed that the evaluation of the quadratic functions and all other numerical operations involved in this algorithm are computationally non-expensive.

The Constraint Quadratic Approximation for Multiple Equality Constraints (CQA-MEC) local search operator is defined as an operation of quadratic approximation followed by the application of the iterative projection algorithm, as defined above.

### III. HYBRIDIZING THE GA WITH THE LOCAL SEARCH OPERATOR

The basic scheme for using the CQA-MEC operator inside a genetic algorithm is:

Step 1. Initialize parameters
Step 2. Initialize population
Step 3. WHILE no stop criterion
The local search phase receives a small number of points (we have employed from one to five points) to be locally enhanced. Each point is submitted to the CQA-MEC operator. Each resulting enhanced point is put inside the population, just before the operation of selection that will start the next generation. A possible variation of this scheme is: the local search (CQA-MEC) operator is run each \( p \) generations. In examples presented in this paper, the operator has been executed each \( 5 \) generations.

A key observation that allows the development of the approach proposed here is: population-based methods, in their normal execution, perform a number of samplings of the objective and constraint functions, whose information may be used to fit local approximations of these functions. In fact, the usual heuristic operators of a genetic algorithm perform an intensive sampling that becomes increasingly concentrated around the most promising regions of the search space. It should be noticed that the quadratic approximations are the parametric models that need less information for being built, while keeping a description of the function non-linear behavior. This means that even for relatively high-dimensional problems the construction of such approximations remains feasible, and that the approximations become progressively better, as the algorithm evolves, and a more dense sample set is obtained nearby the optimum.

Notice that, in the proposed methodology, the local search procedure with quadratic approximation does not replace the normal genetic algorithm functioning. As the GA finds better candidate solutions, the samples around such solution, produced by the usual operations inside the GA, are used for fitting locally-valid quadratic approximations. The enhanced-feasibility candidate solution resulting from the approximation-based local search procedure is calculated with a semi-analytical procedure, allowed by the quadratic function structure, and then returned to the GA normal population, leading to an overall more accurate search.

In some cases, the application of CQA-MEC operator, in the initial generations of a genetic algorithm, does not converge to the solution of the quadratic auxiliary problem (the feasibility is not attained). This occurs because the point that is to be locally enhanced is far from the solution, leading to divergence. However, in such cases, as the GA evolves, it produces better estimates of the original problem solution, due to GA conventional operations only. This leads to three combined effects: (i) the quadratic approximation of the problem functions becomes better, as more function samples become available; (ii) the initial point to be submitted to CQA becomes closer to the constraint surface; and (iii) the problem optimum is also approached, with the objective function value being enhanced due to the selective pressure established by the genetic algorithm selection. This means that the CQA-MEC operator eventually becomes functional – when this occurs, the local search operation makes the search to become much faster and reliable. The final convergence to the problem optimum is due to the combined effects of constraint satisfaction, associated to CQA-MEC operator, and of fitness enhancement, associated to the genetic algorithm selection operation.

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

### A. The Real-Biased GA

The Real-Biased Genetic Algorithm (RBGA) \([14, 15]\) is a real-coded 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 \( p_c \).
- For each pair subjected to crossover, the fitness function \( J(x) \) of the individuals is considered. The individuals (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 \( x_g = \alpha x_1 + (1 - \alpha) x_2 \) with \( \alpha \) chosen in the interval \( [-\xi; 1 + \xi] \), with \( \xi \) being an extrapolation factor selected by the user from the interval \( [0; 1] \). \( \alpha \) is selected according to the probability distribution defined by \( \alpha = (1 + 2 \xi) \beta_1 \beta_2 - \xi \) where \( \beta_1 \) and \( \beta_2 \) are random variables with uniform probability distribution inside the domain \( [0; 1] \). This provides a quadratic probability distribution for \( \alpha \) which makes the new individual \( x_g \) to 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, \( \alpha \) is chosen in the interval \( [-\xi; 1 + \xi] \) with uniform probability.

Notice that if the real-biased crossover is applied just after the selection, this crossover will not incur in any additional cost of function evaluation. The specific evaluation of the effect of the real biased crossover operator can be found in \([14]\).

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 is created taking into account such information.

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
\[
x_r = x_L + |x - x_L|
\]
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
\[
x_r = x_U - |x_U - x|
\]
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
\[
x_m = x + \delta
\]
with \( \delta_i = 0.05\beta_i(x_r)_i \), where \( \beta_i \) 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.

B. The RBGA-CQA-MEC Hybrid

The specific implementation of hybridization of RBGA with the CQA-MEC operator has been performed as follows.

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

\[
F(x) = f(x) + 100 \sum_{i=1}^{k} |g_i(x)|
\]

- The local search CQA-MEC operator will be run every 5 generations.

- 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 precise the quadratic approximation.

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

IV. RESULTS

The RBGA and RBGA-CQA-MEC 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:

**Problem 1**
\[
x^* = \arg \min_x (x_1 - 4)^2 + (x_2 - 6)^2 + (x_3 - 1)^2 + (x_4 - 2)^2 + (x_5 - 1)^2
\]
subject to:
\[
\begin{align*}
g_1(x) &= x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0 \\
g_2(x) &= x_2 x_3 - 5 x_4 x_5 = 0 \\
g_3(x) &= x_1^2 + 3 x_2 + 1 = 0 \\
-2.3 &\leq x_i \leq 2.3, \quad i = 1, \cdots, 4 \\
-3.2 &\leq x_5 \leq 3.2
\end{align*}
\]

**Problem 2:**
\[
x^* = \arg \min_x (x_1^2 + x_2^2 + x_3^2 - 14 x_1)_i - 10 x_2 + (x_3 - 10)^2 + 4 (x_4 - 10)^2 + 2 (x_5 - 10)^2 + 7 (x_6 - 11)^2 + 2 (x_7 - 10)^2 + (x_10 - 7)^2 + 45;
\]
subject to:
\[
\begin{align*}
g_1(x) &= 3 (x_1 - 1)^2 + 4 (x_2 - 3)^2 + 2 x_3^2 - 7 x_4 + 12 = 0 \\
g_2(x) &= 5 x_1^2 + 8 x_2 + (x_3 - 6)^2 - 2 x_4 - 40 = 0 \\
g_3(x) &= x_1^2 + 2 (x_2 - 2)^2 - 2 x_1 x_2 + 14 x_5 - 6 x_6 = 0 \\
-10 &\leq x_i \leq 10, \quad i = 1, \cdots, 10
\end{align*}
\]

**Problem 3:**
\[
x^* = \arg \min_x (x_1 - 10)^2 + 5 (x_2 - 12)^2 + x_3 x_1^2 - 4 x_4 x_7 - 10 x_6 - 8 x_7
\]
subject to:
\[
\begin{align*}
g_1(x) &= -7 + 2 x_1^2 + 3 x_2^2 + x_3 + 4 x_4^2 + 5 x_5 = 0 \\
g_2(x) &= 4 x_1^2 + x_2 - 3 x_4 x_5 - 2 x_1^2 + 5 x_6 - 11 x_7 = 0 \\
-10 &\leq x_i \leq 10, \quad i = 1, \cdots, 7
\end{align*}
\]

In both algorithms, each equality constraint \( g(x) = 0 \) is replaced by two inequality constraints of the form
\[
\begin{align*}
g_1(x) &= -g(x) - \epsilon \leq 0 \\
g_2(x) &= g(x) - \epsilon \leq 0
\end{align*}
\]
with \( \epsilon = 0.001 \), for the purpose of performing fitness assignment.

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

2007 IEEE Congress on Evolutionary Computation (CEC 2007) 3047
V. CONCLUSIONS

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

The CQA-MEC 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 multiple equality constraints. This is especially emphasized in the case of problems with expensive-to-evaluate functions.

ACKNOWLEDGMENT

The authors acknowledge the support of the Brazilian agencies CAPES, CNPq and FAPEMIG.

REFERENCES


