Classification-assisted Memetic Algorithms for Equality-constrained Optimization Problems

S.D. Handoko, C.K. Kwoh, and Y.S. Ong

School of Computer Engineering,
Nanyang Technological University, Singapore 639798
danielhandoko@pmail.ntu.edu.sg

Abstract. *Regressions* has successfully been incorporated into memetic algorithm (MA) to build surrogate models for the objective or constraint landscape of optimization problems. This helps to alleviate the needs for expensive fitness function evaluations by performing local refinements on the approximated landscape. *Classifications* can alternatively be used to assist MA on the choice of individuals that would experience refinements. Support-vector-assisted MA were recently proposed to alleviate needs for function evaluations in the inequality-constrained optimization problems by distinguishing regions of feasible solutions from those of the infeasible ones based on some past solutions such that search efforts can be focussed on some potential regions only. For problems having equality constraints, however, the feasible space would obviously be extremely small. It is thus extremely difficult for the global search component of the MA to produce feasible solutions. Hence, the classification of feasible and infeasible space would become ineffective. In this paper, a novel strategy to overcome such limitation is proposed, particularly for problems having one and only one equality constraint. The raw constraint value of an individual, instead of its feasibility class, is utilized in this work.

1 Introduction

Real-world optimization problems are often constrained. Generally, they can be formulated as finding some vector $\mathbf{x}$ of n real-valued independent variables that minimizes

$$f(\mathbf{x}) \quad (1)$$

subject to

$$\mathbf{g}(\mathbf{x}) \leq \mathbf{0} \quad (2)$$

$$\mathbf{h}(\mathbf{x}) = \mathbf{0} \quad (3)$$

where $\mathbf{x} \in \Re^n$ is often referred to as the *solution*, while $f : \Re^n \rightarrow \Re$ the *objective*, whereas $\mathbf{g} : \Re^n \rightarrow \Re^{n_g}$ and $\mathbf{h} : \Re^n \rightarrow \Re^{n_h}$ the *inequality* and *equality constraints*, respectively. There may also be bound constraints of the form $\mathbf{x}_\ell \leq \mathbf{x} \leq \mathbf{x}_u$ with $\mathbf{x}_\ell$ being the *lower* and $\mathbf{x}_u$ being the *upper bound*.

In addition, real-world problems often involve expensive computation of their objective/constraint functions. Potential energy minimization in computational molecular chemistry or biology, for example, demands minutes to hours in each function evaluation depending on the size of the molecule as well as the fidelity of the model being used. The number of function evaluations required to solve problems in this category is therefore a significant issue.

One method to deal with such situation when only the inequality constraints are present were proposed in [1]. Five benchmark problems experimented with, all of which have sufficiently reasonable ratio of the feasible to the whole search space, were solved within less amount of function evaluations using the proposed method. Dealing with the equality-constrained optimization problems, however, extremely small ratio of the feasible to the whole search space poses challenges to the global search algorithm to find a feasible solution, deeming the proposed method that separates the regions of feasible from infeasible solutions unsuited. In this paper, a novel strategy designed for the equality-constrained problems is proposed with a primary focus on problems with single equality constraint only.

2 Literature Review

2.1 Deterministic Algorithms

Methods of feasible directions is a class of deterministic algorithms that proceed from one feasible solution to another in order to solve constrained optimization problems [2]. *Zoutendijk* algorithm [3] and *sequential linear programming* (SLP) approaches [4][5][6] employ first-order approximation to both the objective and the constraints and are consequently prone to slow convergence. By employing second-order functional approximation, *sequential quadratic programming* (SQP) technique [7] enjoys quadratic rate of convergence and is the state-of-the-art of nonlinear programming solvers [8].

The following quadratic program is solved for direction $\mathbf{d}$ at the i-th major iteration of the SQP.

$$f(\mathbf{x}^{(i)}) + \nabla f(\mathbf{x}^{(i)})^T \mathbf{d} + \frac{1}{2}\mathbf{d}^T \nabla^2 L(\mathbf{x}^{(i)})\mathbf{d} \tag{4}$$

subject to

$$g_j(\mathbf{x}^{(i)}) + \nabla g_j(\mathbf{x}^{(i)})^T \mathbf{d} \leq 0 \qquad j = 1, \ldots, n_g \tag{5}$$

$$h_j(\mathbf{x}^{(i)}) + \nabla h_j(\mathbf{x}^{(i)})^T \mathbf{d} = 0 \qquad j = 1, \ldots, n_h \tag{6}$$

where

$$\nabla^2 L(\mathbf{x}^{(i)}) = \nabla^2 f(\mathbf{x}^{(i)}) + \sum_{i=1}^{n_g} \mu_j^{(i)} \nabla^2 g_j(\mathbf{x}^{(i)}) + \sum_{i=1}^{n_h} \nu_j^{(i)} \nabla^2 h_j(\mathbf{x}^{(i)}) \tag{7}$$

Throughout this work, the gradient vectors are assumed to be readily available while the Hessian matrices are updated using the quasi-Newton approximation.

Although quadratic rate of convergence is achievable, it is well known that deterministic optimization algorithms may not converge to the global optimum. Constrained optimization problems with nonlinear objective or constraints are in general intractable. It is impossible to design a deterministic algorithm that would outperform the exhaustive search in assuring global convergence [9].

2.2 Randomized Algorithms

Genetic algorithm (GA) [10] is a randomized algorithm with ability to overcome the drawback of deterministic optimization algorithms. Belonging to the class of *evolutionary computing*, GA is motivated by the natural inheritance of genes and the natural selection in the course of biological evolution [11] with the crossover, the mutation, and the survival-of-the-fittest being at its very heart. The simplest form of the algorithm assumes only one population evolved from one generation to the next. In dealing with constraints, the ranking scheme in [12] is often used. Summarized in the following three points, it is employed throughout this work.

- The feasible solution is preferred to the infeasible one.
- Between two feasible solutions, the one having better objective is preferred.
- Between two infeasible solutions, the one having less amount of violation to the constraints is preferred.

Research works on constrained evolutionary computing over the last decade include Homomorphous Mapping [13], Stochastic Ranking [14], the ASCHEA [15], Simple Multimembered Evolution Strategy (SMES) [16] that is known to have used the smallest number of fitness function evaluations (FFEs) so far, and some others [17][18][19][20]. Even though specially-designed operators accelerate the search for the global optimum to certain extent, it is a consensus that GA may suffer from excessively slow convergence trying to locate the optimum with sufficient precision because of its failure in exploiting local information [21].

2.3 Hybrid Algorithms

When hybridizing optimization methods, two central yet competing goals meet: *exploration* and *exploitation* [22]. The exploration provides reliable estimates of the global optimum by surveying the search space using global search methods, which are accommodated by the randomized algorithms. The exploitation then enhances each estimate by focussing the search efforts on its local neighborhood in order to produce a sufficiently accurate global optimum. This is accomplished using local search methods, which are facilitated by the deterministic algorithms. Motivated by Dawkins' notion of meme [23] (unit of cultural evolution capable of local refinements), a *memetic algorithm* (MA) exhibits this particular behavior. As the simplest variant, the simple MA simply interleaves global and local search methods one after the other. When compared to its conventional counterparts, the simple MA performs better by converging to a high quality global optimum and searching more efficiently [21].

Local refinements for each individual in the population, unfortunately, need not necessarily be the most efficient strategy. Local refinements of solutions at different locations may end with the same local optimum. Local search methods, such as the SQP, are known to converge quickly only when initialized with an approximate solution close enough to the optimal solution. Thus, the choice of individuals that should undergo local refinements becomes a critical issue in MA.

3 Proposed Approach

3.1 The Global Optimum

The optimization problem constrained by one and only one equality constraint always has global optimum situated at some particular location along the curve defined by $h(\mathbf{x}) = 0$. This curve is the feasible space of the problem. Illustrated in Fig. 1 is the constraint space of benchmark problem **g11**—the objective and constraint functions of which can be found in [24]. The solid curve represents the feasible space and the dot the global optimum of the problem. For this type of problems, the feasible space sets the solutions with positive constraint values apart from those with negative ones.

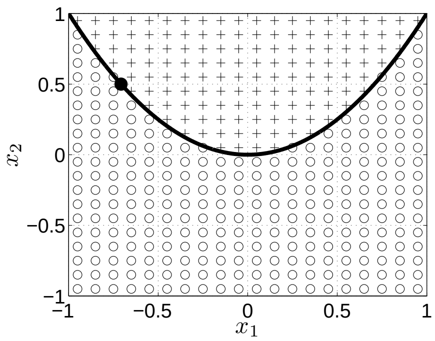

Fig. 1. Constraint Space of Benchmark Problem **g11**
$\circ$: constraint space where $h(\mathbf{x}) < 0$; $+$: constraint space where $h(\mathbf{x}) > 0$

It is understood that local search methods, such as the SQP, converge quickly to local optimum when they are initialized with an approximate solution that is close enough to the optimal solution [2]. Because one of the possibly many local optima must be the global optimum, focussing the search efforts on the regions nearby the feasible space will definitely increase the odds of being more efficient in locating the global optimum of the problem. This is achieved in this work by utilizing one fact that the feasible space of an optimization problem with single equality constraint is the zero-crossing of the constraint value.

3.2 The Neighborhood

Similar to [1], neighborhood $\mathcal{N}$ of the individual $\mathbf{x}$ is defined as the collection of k nearest solutions (to the individual) obtained from database of past solutions. The Euclidean distance in (8) is used throughout as a sparsity measure between any two n-dimensional solutions $\mathbf{p}$ and $\mathbf{q}$.

$$d_{\mathbf{pq}} = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \tag{8}$$

For the k neighbors are infallibly past solutions, no additional FFE is necessary to know quantities associated with these solutions. Freely accessible information include the constraint values based on which two classes can be derived. Should a neighbor and its corresponding class be represented as $\mathbf{x}_i$ and y_i, respectively, $\mathcal{N} = \{(\mathbf{x}_i, y_i) : i = 1, 2, \ldots, k\} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_k, y_k)\}$ defines the information contained within this neighborhood with

$$y_i = sign(h(\mathbf{x}_i)) = \begin{cases} +1 & h(\mathbf{x}_i) > 0 \\ -1 & h(\mathbf{x}_i) < 0 \end{cases} \tag{9}$$

Different from [1], the neighbors in this work are restricted to past solutions $\mathbf{x}_i$ for which $h(\mathbf{x}_i) \neq 0$. This means all the neighbors shall be infeasible—which is indeed desirable as the regions surrounding the feasible space must be infeasible. By making use of the signs of the constraint values of an individual's neighbors, it is demanded that the individual relative position can be predicted such that local search will only be executed if the individual is nearby the feasible space of the optimization problem.

Mixed Neighborhood This type of neighborhood consists of members having positive and negative constraint values. Neighborhood of this type is absolutely of significant interest and importance. A two-class classification subproblem can be formulated out of this scenario. The decision boundary produced as the result of solving the classification subproblem does not only distinguish the regions of positive constraint values from those of the negative ones, but also approximates the feasible space of the optimization problem locally. *Support Vector Machine* (SVM) [25] will be described in the next subsection to serve this purpose.

Positive-only Neighborhood A positive-only neighborhood, as indicated by its name, consists of members having only positive constraint values. This is of little or no importance as there is no clue about the feasible space of the problem that can be deduced from this type of neighborhood.

Negative-only Neighborhood A negative-only neighborhood, as revealed by its name, consists of members having only negative constraint values. Similarly, no clue about the feasible space of the problem can be mined from this type of neighborhood, making it of little or no significance.

3.3 The Support Vector Machine (SVM)

SVM is a machine-learning technique initially proposed as a two-class classifier. It is well-known as being characterized by its ability to maximize the geometric margin between the two classes, and simultaneously, minimize the classification error. Upon provision of k training data instances $(\mathbf{x}_i, y_i)$ where $y_i \in \{-1, +1\}$ for all $i = 1, 2, \ldots, k$, the SVM needs to maximize the quadratic program below.

$$\sum_{i=1}^{k} \alpha_i - \frac{1}{2} \sum_{i=1}^{k} \sum_{j=1}^{k} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \tag{10}$$

subject to

$$\sum_{i=1}^{k} y_i \alpha_i = 0 \tag{11}$$

$$\forall i \ \alpha_i \geq 0 \tag{12}$$

Collection of training data instances having $\alpha > 0$ defines the support vectors. Every one of them is situated at the decision surface $D(\mathbf{x}) = +1$ or $D(\mathbf{x}) = -1$ depending on which class it belongs to. Weight vector $\mathbf{w}$ and bias w_0 are hence computed using (13) and (14) in which SV is the set of support vectors indices.

$$\mathbf{w} = \sum_{i=1}^{k} \alpha_i y_i \mathbf{x}_i = \sum_{i \in SV} \alpha_i y_i \mathbf{x}_i \tag{13}$$

$$w_0 = \frac{1}{|SV|} \sum_{i \in SV} \left(y_i - \sum_{j=1}^{k} \alpha_j y_j (\mathbf{x}_j \cdot \mathbf{x}_i) \right) \tag{14}$$

Upon encountering a mixed neighborhood, the SVM needs training based on the k instances of $\mathcal{N}$. Subsequently, the SVM can be used to predict the relative position of the individual $\mathbf{x}$ with respect to neighborhood $\mathcal{N}$, producing one of the following three possible outcomes.

1. $|D(\mathbf{x})| \leq +1$
   The individual $\mathbf{x}$ has been estimated to be located nearby the feasible space. With no neighbors found within this region, furthermore, local refinement is obviously necessary to exploit this seemingly unexplored search space.
2. $D(\mathbf{x}) > +1$
   Depending on the actual value of $D(\mathbf{x})$, the individual $\mathbf{x}$ may be close enough to the feasible space. With neighbors around, there may not be further need to exploit this previously explored region of the search space.
3. $D(\mathbf{x}) < -1$
   Similar to case 2, the individual $\mathbf{x}$ may be located close to the feasible space depending on the value of $D(\mathbf{x})$. With neighbors around, no exploitation of this previously explored search space would be necessary.

3.4 The Complete Algorithm

Algorithm 1 presents the complete algorithm of the proposed approach in pseudocode form. An important point worth noting is that the size of the neighborhood is recommended to be some multiple of the dimensionality of the problem being solved such that it would be large enough to capture important information yet small enough to ensure locality and allow the SVM to perform reasonably fast as its complexity depends largely on the number of training data instances. While the cost of running the SVM may not be inexpensive, the efforts required for evaluating the objective and constraint functions may be magnitudes greater for many practical optimization problems. Thus, the additional budget incurred by the SVM will become insignificant when dealing with some computationally-expensive optimization problems.

Algorithm 1 Classification-assisted MA (CaMA)

```
Initialize a population
Evaluate the population
while no stopping criteria have been fulfilled do
  for each individual x in the population do
    if past solutions are of negative only or positive only constraint values then
      Refine x using local search
    else
      if neighborhood 𝒩 of x is a mixed neighborhood then
        Train SVM based on 𝒩 to obtain decision function D(·)
        if |D(x)| ≤ 1 then
          Refine x using local search
        end if
      end if
    end if
  end for
  Evolve the population through crossover, mutation, and elitism
  Evaluate the population
end while
```

4 Results and Discussions

Using GA as the global and SQP as the local search method, an empirical study was carried out with a population size of 100 individuals and a maximum of $2n$ fitness function evaluations (FFEs) for each individual refinement with n being the dimensionality of the problem. Experimented with are benchmark problem **g03** for $n = 2, \ldots, 10$ and **g11**, the objective and constraint functions of which can be found in [24]. These are the only two benchmark problems having single equality constraints among the 24 problems in [24]. When the proposed method was used, a neighborhood size of $2n$ was assumed.

Table 1. Number of FFEs Required by SMA and CaMA to Locate the Global Optimum

Problem	Statistics	SMA	CaMA	Saving
g03 ($n = 2$)	best average worst	284 362 471	124 152 187	58.01%
g03 ($n = 3$)	best average worst	935 1,116 1,244	220 307 411	72.49%
g03 ($n = 4$)	best average worst	1,543 1,681 1,812	284 434 552	74.18%
g03 ($n = 5$)	best average worst	1,842 2,083 2,258	258 394 536	81.08%
g03 ($n = 6$)	best average worst	2,212 2,389 2,639	228 337 642	85.89%
g03 ($n = 7$)	best average worst	2,499 2,714 2,955	148 407 2,763	85.00%
g03 ($n = 8$)	best average worst	2,763 3,572 5,747	180 1,175 3,499	67.11%
g03 ($n = 9$)	best average worst	3,001 4,330 6,257	157 2,351 4,441	45.70%
g03 ($n = 10$)	best average worst	3,429 5,539 6,477	401 2,945 5,788	46.83%
g11 ($n = 2$)	best average worst	523 606 712	116 154 225	74.59%

Table 1 tabulates the performance of 30 independent runs of the simple MA (SMA) as well as the classification-assisted MA (CaMA) proposed in this paper. As the simplest variant of MA, the simple MA simply interleaves the global with the local search methods one after the other. In other words, each individual in the population would experience local refinements in the context of simple MA. The percentage of saving achievable by the CaMA with respect to the SMA on the average is then calculated as follows.

$$\text{saving} = \frac{\#\text{FFEs}_{(\text{SMA})} - \#\text{FFEs}_{(\text{CaMA})}}{\#\text{FFEs}_{(\text{SMA})}} \times 100\% \qquad (15)$$

It is clear from the table that the CaMA consistently outperforms the SMA in the best, worst, and average cases. With average savings ranging from about 45% to 85%, the CaMA shall undoubtedly bring great advantage when solving computationally-expensive optimization problems. For this category of problems, reductions of several hundreds to several thousands of FFEs as exhibited over the benchmark problems could easily translate to savings of days to months of computational time. All these are made possible as search efforts were focussed around the regions that surround the feasible space of the optimization problems, thanks to the classification algorithms, such as the SVM, that enable prediction of local feasible space of the problems. As unnecessary refinements initiated with solutions relatively far away from the feasible space are eliminated, less number of FFEs are required in locating the global optimum of the problems.

5 Conclusion

Raw constraint values—rather than feasibility classes—are utilized in this work to focus search efforts on the regions that surround the minute feasible space of optimization problems with single equality constraint. Savings of up to 85% are achievable in term of the number of fitness function evaluations needed to solve benchmark problem **g03** with dimensionality varied from 2 to 10 as well as **g11**. In the optimization of computationally-expensive problems, such amount could bring significant time reduction. Thus, generalization to problems with multiple equality constraints and possibly some inequality constraints shall be addressed in immediate future.

References

1. Handoko, S.D., et. al., "Using Classification for Constrained Memetic Algorithm: A New Paradigm," in *Proceedings of the 2008 IEEE International Conference on Systems, Man and Cybernetics*, 2008.
2. Bazaraa, M.S., H.D. Sherali, and C.M. Shetty, *Nonlinear Programming: Theory and Algorithms*, Wiley-Interscience, 2006.
3. Zoutendijk, G., *Methods of Feasible Directions*, Elsevier, 1960.
4. Baker, T.E. and R. Ventker, "Successive Linear Programming in Refinery Logistic Models," presented at the ORSA/TIMS Joint National Meeting, 1980.
5. Baker, T.E. and L.S. Lasdon, "Successive Linear Programming at Exxon," *Management Science*, 31(3), pp. 264–274, 1985.
6. Zhang, J.Z., N.H. Kim, and L.S. Lasdon, "An Improved Successive Linear Programming Algorithm," *Management Science*, 31(10), pp. 1312–1331, 1985.
7. Wilson, R.B., "A Simplicial Algorithm for Convex Programming," Ph.D. Thesis, Harvard University, 1963.
8. Schittkowski, K., "NLPQL: A FORTRAN Subroutine Solving Constrained Nonlinear Programming Problems," *Annals of Operations Research*, 5(2), pp. 485–500, 1986.
9. Wright, S.J., "Nonlinear and Semidefinite Programming," in *Proceedings of Symposia in Applied Mathematics*, 61, pp. 115–138, 2004.

10. Holland, J.H., *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, 1975.
11. Darwin, C., *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*, John Murray, 1859.
12. Deb, K., "An Efficient Constraint Handling Method for Genetic Algorithms," *Computer Methods in Applied Mechanics and Engineering*, 186(2–4), pp. 311–338, 2000.
13. Koziel, S. and Z. Michalewicz, "Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization," *Evolutionary Computation*, 7(1), pp. 19–44, 1999.
14. Runarsson, T.P. and Y. Xin, "Stochastic Ranking for Constrained Evolutionary Optimization," *IEEE Transactions on Evolutionary Computation*, 4(3), pp. 284–294, 2000.
15. Hamida, S.B. and M. Schoenauer, "ASCHEA: New Results Using Adaptive Segregational Constraint Handling," in *Proceedings of the 2002 Congress on Evolutionary Computation*, pp. 884–889, 2002.
16. Mezura-Montes, E. and C.A.C. Coello, "A Simple Multi-Membered Evolution Strategy to Solve Constrained Optimization Problems," *IEEE Transactions on Evolutionary Computation*, 9(1), pp. 1–17, 2005.
17. Barbosa, H.J.C. and A.C.C. Lemonge, "A New Adaptive Penalty Scheme for Genetic Algorithms," *Information Science*, 156(3–4), pp. 215–251, 2003.
18. Farmani, R. and J.A. Wright, "Self-Adaptive Fitness Formulation for Constrained Optimization," *IEEE Transactions on Evolutionary Computation*, 7(5), pp. 445–455, 2003.
19. Chootinan, P. and A. Chen, "Constraint Handling in Genetic Algorithms Using a Gradient-based Repair Method," *Computers and Operation Research*, 33(8), pp. 2263–2281, 2006.
20. Elfeky, E.Z., R.A. Sarker, and D.L. Essam, "A Simple Ranking and Selection for Constrained Evolutionary Optimization," in *Lecture Notes in Computer Science*, 4247, Springer-Verlag, 2006.
21. Tang, J., M.H. Lim, and Y.S. Ong, "Diversity-Adaptive Parallel Memetic Algorithm for Solving Large Scale Combinatorial Optimization Problems," *Soft Computing: A Fusion of Foundations, Methodologies, and Applications*, 11(9), pp. 873–888, 2007.
22. Torn, A. and A. Zilinskas, *Global Optimization*, Springer-Verlag, 1989.
23. Dawkins, R., *The Selfish Gene*, Oxford University Press, 1976.
24. Liang, J.J., et al., *Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-Parameter Optimization*, Technical Report, 2006.
25. Vapnik, V.N., *The Nature of Statistical Learning Theory*, Springer, 1995.