# Constrained Multi-Objective GA Optimization Using Reduced Models

**Xuan Jiang**
Dept. of Computer Science
University of Georgia
Athens, GA 30602
xuan@cs.uga.edu

**Deepti Chafekar**
Dept. of Computer Science
University of Georgia
Athens, GA 30602
chafekar@cs.uga.edu

**Khaled Rasheed**
Dept. of Computer Science
University of Georgia
Athens, GA 30602
khaled@cs.uga.edu

## Abstract

In this paper we propose a novel approach for solving constrained multi-objective optimization problems using a steady state GA and reduced models. Our method called Objective Exchange Genetic Algorithm for Design optimization (OEGADO) is intended for solving real-world application problems that have many constraints and very small feasible regions. OEGADO runs several GAs concurrently with each GA optimizing one objective and exchanging information about its objective with others. Empirical results in benchmark and engineering design domains are presented. A comparison between OEGADO and Non-Dominated Sorting Genetic Algorithm-II (NSGA-II) shows that OEGADO performed better than NSGA-II for difficult problems, and found Pareto-optimal solutions in fewer objective evaluations. The results suggest that our method may be better for solving real-world application problems wherein the objective computation time is large.

## 1 INTRODUCTION

This paper concerns the application of steady state Genetic Algorithms (GAs) in realistic engineering design domains which usually involve simultaneous optimization of multiple and conflicting objectives with many constraints. In these problems instead of a single optimum there usually exists a set of trade-off solutions called the non-dominated solutions or Pareto-optimal solutions. No other solutions in the search space are superior to these Pareto-optimal solutions when all objectives are considered.

Some of the challenges faced in applying GAs to engineering design domains are:

- The search space can be very complex with many constraints and the feasible (physically realizable) region in the search space can be very small.
- Determining the quality (fitness) of each point may involve the use of a simulator which takes a non-negligible amount of time. This simulation time can range from a fraction of a second to several days in some cases.

For such problems steady state GAs may perform better than generational GAs because they better retain the feasible points found in their population and may have higher selection pressure.

Most of the recent approaches in the area of multi-objective optimization using Evolutionary Algorithms (EAs) propose the use of a generational GA. Deb proposed an Elitist Steady State Multi-objective Evolutionary Algorithm (MOEA) (Deb, 2001) which attempts to maintain spread while attempting to converge to the true Pareto-optimal front. This algorithm requires sorting of the population for every new solution formed thereby increasing its time complexity. Very high time complexity makes the Elitist steady state MOEA impractical for some problems. To the best of our knowledge, apart from Elitist Steady State MOEA, the area of a steady state multi-objective GA has not been explored widely. Also constrained multi-objective optimization which is very important for real-world application problems has not received the deserved exposure. In this paper we propose a method for solving constrained multi-objective optimization using steady state GAs. This method is relatively faster, practical and has fairly low time complexity. It is also easy to transform a single-objective GA to a multi-objective GA by using our method.

Our method (OEGADO) uses reduced models for multi-objective optimization, as described in the following section. It can be viewed as a multi-objective transaformation of GADO (Genetic Algorithm for Design Optimization) (Rasheed, 1998). GADO is  a GA that was designed with the goal of being suitable for the use in engineering design. It has demonstrated a great deal of robustness and efficiency relative to competing methods.

We compared the results of the Objective Exchange Genetic Algorithm for Design Optimization (OEGADO) with the state-of-the-art Elitist Non-Dominated Sorting Algorithm (NSGA-II) (Deb, 2000a). NSGA-II is a non-dominated sorting based multi-objective evolutionary algorithm that incorporates an elitist approach, parameter-less niching approach and simple constraint handling strategy.

In the remainder of the paper, we provide a brief description of our proposed method. We then present results of the comparison of our method with NSGA-II. Finally, we conclude the paper with a discussion of the results and future work.

# 2 OBJECTIVE EXCHANGE GENETIC ALGORITHM FOR DESIGN OPTIMIZATION (OEGADO)

The main idea of OEGADO is to run several single objective GAs concurrently. Each of the GAs optimizes one of the objectives. All the GAs share the same representation and constraints, but have independent populations. They exchange information about their respective objectives every certain number of iterations.

In our implementation, we have used the idea of informed operators (IOs). The main idea of the IOs is to replace pure randomness used in the original operators with decisions that are guided by reduced models formed using the methods presented in (Rasheed, 2002). The reduced models are approximations of the fitness function, formed using some approximation techniques, such as least squares approximation (Rasheed, 2002). These functional approximations are then used to make the GA operators such as crossover and mutation more informed. These IOs generate multiple children and rank them using the approximate fitness obtained from the reduced model.

Every single objective GA in OEGADO uses least squares to form a reduced model of its own objective. Every GA exchanges its own reduced model with those of the other GAs. In effect, every GA, instead of using its own reduced model, uses other GAs' reduced models to compute the approximate fitness of potential individuals. Therefore each GA is informed about other GAs' objectives. As a result each GA not only focuses on its own objective, but also gets biased towards the objectives which the other GAs are optimizing.

The OEGADO algorithm for two objectives looks as follows:

1. Both the GAs are run concurrently for the same number of iterations, each GA optimizes one of the two objectives while also forming a reduced model of it.
2. At intervals equal to twice the population size, each GA exchanges its reduced model with the other GA.
3. The conventional GA operators such as initialization (only applied in the beginning), mutation and crossover are replaced by informed operators. The IOs generate multiple children and use the reduced model to compute the approximate fitness of these children. The best individual based on this approximate fitness is selected to be the newborn. It should be noted that the approximate fitness function used is of the other objective.
4. The true fitness function is then called to evaluate the actual fitness of the newborn corresponding to the current objective.
5. The individual is then added to the population according to the replacement strategy.
6. Steps 2 through 5 are repeated till the maximum number of evaluations is reached.

If both objectives have the same computational complexity, the two GAs can be synchronized. On the other hand, when objectives vary considerably in their time complexity, the GAs can be run asynchronously.

It should be noted that OEGADO is not really a multi-objective GA, but two single objective GAs working concurrently to get the Pareto-optimal region. Each GA finds its own feasible region, by evaluating its own objective. For the feasible points found by a single GA, we need to run the simulator to evaluate the remaining objectives. Thus for OEGADO with two objectives:

*Total number of objective evaluations = Sum of evaluations of each GA + Sum of the number of feasible points found by each GA*

A potential advantage of this method is speed, as the concurrent GAs can run in parallel. Therefore multiple objectives can be evaluated at the same time on different CPUs. Also the asynchronous OEGADO works better for objectives having different time complexities. If some objectives are fast, they are not slowed down by the slower objectives. It should be noted that because of the exchange of reduced models, each GA optimizes its own objective and also gives credit to the other objectives.

# 3 EXPERIMENTAL RESULTS

## 3.1 TEST PROBLEMS

The test problems for evaluating the performance of our method were chosen based on significant past studies. We chose four problems from the benchmark domains commonly used in past multi-objective GA research, and two problems from the engineering domains. The degree of difficulty of these problems varied from fairly simple to difficult.

The problems chosen from the benchmark domains are BNH used by Binh and Korn (Deb, 2001), SRN used by Srinivas, Deb (Deb, 2001), TNK suggested by Tanaka (Deb, 2001) and OSY used by Osyczka, Kundu (Deb, 2001). The problems chosen from the engineering domains are Two-Bar Truss Design used by Deb (Deb, 2000b) and Welded Beam design used by Deb (Deb, 2000b). All these problems are constrained multi-objective problems.

### 3.2 PARAMETER SETTINGS

Each optimization run used the following parameters for the two GAs:

Let *ndim* be equal to the number of dimensions of the problems.

1. Population size: For OEGADO the population size was set to 10**ndim*. For NSGA-II the population size was fixed to 100 as recommended in (Deb, 2000b).
2. Number of objective evaluations: Since the two methods work differently the number of objective evaluations is computed differently. The number of objective evaluations for OEGADO according to Section 2 is given as *Objective evaluations for OEGADO = 2*500*ndim + sum of feasible points found by each GA in OEGADO model*
   NSGA-II is a generational GA, therefore for a two-objective NSGA-II:
   *Total number of objective evaluations =2*population size * number of generations*

Since we did not know exactly how many evaluations would be required by OEGA before hand, to give fair treatment to NSGA-II, we set the number of generations of NSGA-II to be 10**ndim* giving it more evaluations than OEGADO for some cases.

### 3.3 RESULTS

Figures 1-4 present a comparison of the results of the two methods, OEGADO and NSGA-II, for all problems. The outcomes of five runs using different seeds were unified and then the non-dominated solutions were selected and plotted from the union set for each method. We are using graphical representations of the Pareto-optimal curve found by the two methods to compare their performance.

The BNH and SRN (Fig not shown due to space limitation) problems are fairly simple in that the constraints may not introduce additional difficulty in finding the Pareto-optimal solutions. It was observed that both methods performed equally well within comparable number of objective evaluations, and gave a dense sampling of solutions along the true Pareto-optimal curve.

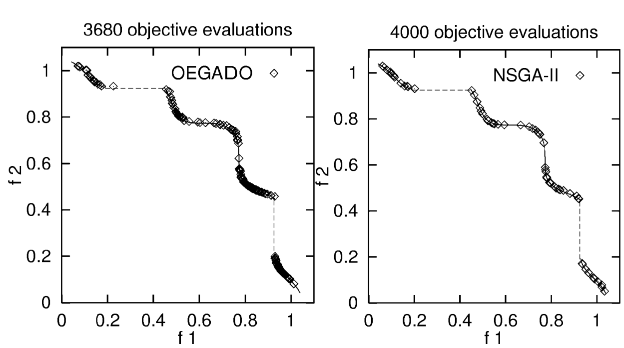


Fig. 1 Results for the benchmark problem TNK

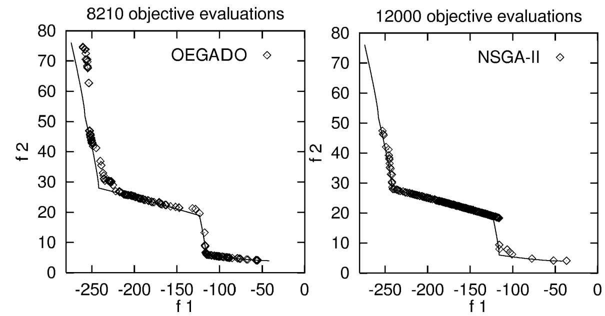


Fig. 2 Results for the benchmark problem OSY

The TNK (Fig. 1) and the OSY (Fig. 2) problems are relatively difficult. The constraints in the TNK problem make the Pareto-optimal set discontinuous. The constraints in the OSY problem divide the Pareto-optimal set into five. As it can be seen from the above graphs for the TNK problem, within comparable number of fitness evaluations, both methods performed equally well. For the OSY problem, it can be seen that OEGADO gave a good sampling of points at the mid-section of the curve and also found points at the extreme ends of the curve. NSGA-II however did not give a good sampling of points at the extreme ends of the Pareto-optimal curve and gave a poor distribution of the Pareto-optimal solutions. In this problem OEGADO outperformed NSGA-II while running for fewer objective evaluations.

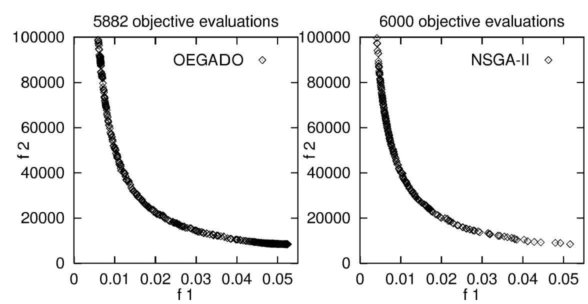


Fig. 3 Results for the Two-bar Truss design problem

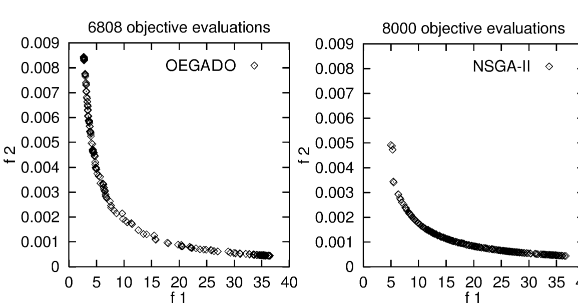


Fig. 4 Results for the Welded Beam design problem

For the Two-bar Truss design problem (Fig.3), within comparable fitness evaluations, NSGA-II performed slightly better than our method in the first objective. OEGADO showed a uniform distribution of the Pareto-optimal curve. In the Welded Beam design problem (Fig.4), the non-linear constraints can cause difficulties in finding the Pareto solutions. Within comparable fitness evaluations, OEGADO outperformed NSGA-II in both distribution and spread. OEGADO found the best minimum solution for $f_1$ with a value of 2.727 units. NSGA-II did not achieve a good distribution of the Pareto solutions at the extreme regions of the curve.

# 4 CONCLUSION AND FUTURE WORK

In this paper we presented a novel method for multi-objective optimization using reduced models, and compared our method with a reliable and efficient generational multi-objective GA called NSGA-II. The results show that a steady state GA can be used efficiently for constrained multi-objective optimization. For the simpler problems OEGADO performed equally well as NSGA-II. For the difficult problems, our method outperformed NSGA-II in most respects. Moreover, OEGADO was able to find the Pareto-optimal solutions for all problems in fewer objective evaluations than NSGA-II. Based on this study we believe that our method is very promising.

Currently, OEGADO does not have any explicit bias towards non-dominated solutions. In future we therefore intend to enhance it by giving credit to non-dominated solutions. We would like to extend OEGADO's implementation to handle more than two objectives and explore its capabilities for more complex real-world applications.

**Acknowledgments**

This research is sponsored by the US National Science Foundation under grant CTS-0121058. The program managers are Drs. Frederica Darema, C. F. Chen and Michael Plesniak.

**References**

(Deb, 2000a) Deb, K., S. Agrawal, A. Pratap, and T. Meyarivan (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *Proceedings of the Parallel Problem Solving from Nature VI*, pp.849-858.

(Deb ,2000b) Deb, K. Pratap, A. and Moitra, S. (2000). Mechanical Component Design for Multiple Objectives Using Elitist Non-Dominated Sorting GA. KanGAL Report No. 200002.

(Deb, 2001) K. Deb. Multi-objective optimization using evolutionary algorithms. Chichester, UK: John Wiley, 2001.

(Rasheed, 1998) Khaled Rasheed. GADO: A genetic algorithm for continuous design optimization. Technical Report DCS-TR-352, Department of Computer Science, Rutgers, The State University of New Jersey, New Brunswick, NJ, January 1998. Ph.D. Thesis.

(Rasheed, 2002) K. Rasheed., S. Vattam, X. Ni. Comparison of Methods for Using Reduced Models to Speed up Design Optimization. In Proceedings of the Genetic and Evolutionary Computation *Conference (GECCO'2002),* pp.1180-1187, 2002.