Multi-Objective GA Optimization Using Reduced Models

Deepti Chafekar, Liang Shi, Khaled Rasheed, *Member, IEEE*, and Jiang Xuan*

***Abstract*—In this paper we propose a novel method for solving multi-objective optimization problems using reduced models. Our method called Objective Exchange Genetic Algorithm for Design Optimization (OEGADO) is intended for solving real-world application problems. For such problems the number of objective evaluations performed is a critical factor as a single objective evaluation can be quite expensive. The aim of our research is to reduce the number of objective evaluations needed to find a well-distributed sampling of the Pareto-optimal region by applying reduced models to steady state multi-objective GAs. OEGADO runs several GAs concurrently with each GA optimizing one objective and forming a reduced model of its objective. At regular intervals, each GA exchanges its reduced model with the others. The GAs use these reduced models to bias their search towards compromise solutions. Empirical results in several engineering and benchmark domains comparing OEGADO with two state-of-the-art Multi-Objective Evolutionary Algorithms show that OEGADO outperformed them for difficult problems.**

***Index Terms*—Genetic Algorithms, Multi-Objective optimization, Reduced models.**

I. Introduction

THIS paper concerns the application of reduced models for constrained multi-objective Genetic Algorithm (GA) optimization. The GA presented in this paper is mainly aimed at solving problems from realistic engineering design domains that usually involve simultaneous optimization of multiple and conflicting objectives with many constraints. In these problems instead of a single optimum there is usually a set of trade-off solutions called the non-dominated solutions or Pareto-optimal solutions (also called Pareto front). For such solutions no improvement in any objective is possible without sacrificing at least one of the other objectives. No other solutions in the search space are superior to these Pareto-optimal solutions when all objectives are considered.

Many challenges are faced in the application of GAs to engineering design domains. A large number of objective evaluations may be required in order to obtain trade-off Pareto-optimal solutions. Moreover, the search space can be complex with many constraints and a small feasible (physically realizable) region. However, determining the quality (fitness) of each point may involve the use of a simulator or an analysis code that takes a long time. Therefore it is impossible to be cavalier with the number of objective evaluations in an optimization.

For such problems multi-objective evolutionary algorithms are preferable as they can find the Pareto front in one run. Many Evolutionary algorithms for solving multi-objective optimization problems have been developed. The most recent ones are the ε-Multi-Objective Evolutionary Algorithm (ε-MOEA) [5], Non-Dominated Sorting Genetic Algorithm-II (NSGA-II) [3], Strength Pareto Evolutionary Algorithm-II (SPEA-II) [16], and Pareto Envelope based selection-II (PESA-II) [2]. Most of these approaches propose the use of a generational GA. The ε-MOEA proposed by Deb is a steady state MOEA based on the ε-dominance concept. The main aim of these methods is obtaining a well-converged and well-distributed Pareto front. There usually exists a trade-off in these methods between obtaining a well-distributed Pareto front and the number of objective evaluations performed. Many real world application problems are computationally complex and performing a large number of objective evaluations on these problems may be very difficult. The ε-MOEA method proposed by Deb is a fast multi-objective evolutionary algorithm in terms of computational time. The goal of our research however is the development of a method that: (i) converges close to the true Pareto front (ii) finds a well distributed Pareto front and (iii) Performs fewer objective evaluations.

In this paper we propose a novel method for multi-objective optimization based on the use of a steady state GA and reduced models. This method is relatively fast and practical. It is also easy to transform a single-objective GA to a multi-objective GA by using our method.

Our method can be viewed as a multi-objective transformation of GADO (Genetic Algorithm for Design Optimization) [9, 12], a GA that was designed with the goal of being suitable for use in engineering design. It uses new operators and search control strategies that target engineering domains [12]. GADO has been successfully applied to a variety of optimization tasks, which span many fields. It demonstrated a great deal of robustness and efficiency relative to competing methods [9].

In GADO, each individual in the GA population

* Manuscript received September 15, 2003. This research is sponsored by the US National Science Foundation under grant CTS-0121058.

D. Chafekar, L. Shi, and K. Rasheed (corresponding author) are with the Computer Science Department, University of Georgia, Athens, GA 30602 USA (emails: {chafekar,shi,khaled}@cs.uga.edu). J. Xuan is with the Computer Science Department, Stanford University, Stanford, CA 94305 USA (email: jakexuan@hotmail.com).

represents a parametric description of an artifact. The fitness of each individual is based on the sum of a proper measure of merit computed by a simulator or some analysis code, and a penalty function if relevant. A steady state model is used, in which several crossover and mutation operators are applied to two parents selected by linear rank based selection. One offspring point is produced, and then an existing point in the population is replaced by the newly generated point. The replacement strategy is a crowding technique, which takes into consideration both the fitness and the proximity of the points in the GA population. GADO monitors the degree of diversity of the GA population. If at any stage it is discovered that the individuals in the population became very similar to one another, the diversity maintenance module rebuilds the population using previously evaluated points in a way that restores diversity. Floating point representation is used. GADO also uses some search control strategies [12] such as a screening module that saves time by avoiding the full evaluation of points that are unlikely to correspond to good designs.

In the remainder of the paper, we provide a brief description of our proposed method along with a background study. We then briefly describe two existing competitive methods for multi-objective optimization and present results of the comparison of our method with them in several benchmark domains. Finally, we conclude the paper with a discussion of the results and future work.

II. THE PROPOSED METHOD

We propose OEGADO, a novel method for solving multi-objective optimization problems using reduced models. This section briefly describes the background behind OEGADO followed by its implementation details.

A. Background

As mentioned earlier, OEGADO can be viewed as a multi-objective transformation of GADO. It has therefore inherited many of the key features of GADO. We have previously extended GADO to incorporate reduced models [10] using informed operators [11] and a least squares approximation technique. We used these features as the basis for performing multi-objective optimizations. In this section we provide a brief description of informed operators and reduced model formation in GADO.

Informed Operators

Reduced models are fast but usually less accurate approximations of the actual fitness evaluation function. Reduced models can be physical, such as models relying on simpler physical equations, or numerical approximations such as response surfaces induced using some input-output pairs evaluated by the original expensive model. Reduced models can be used to speed up the GA optimization or other uses (such as multi-objective optimization as we propose in this paper). The use of reduced models to save time in evolutionary optimization has been extensively researched. A recent survey about fitness approximation in evolutionary computation can be found in [7]. Informed operators (IOs) [11] offer a very convenient way to use reduced models. The main idea behind informed operators is to replace pure randomness in traditional genetic operators with decisions informed by the reduced models.

The types of informed operators used in GADO include:

Informed initialization: For generating an individual in the initial population we generate a number of uniformly distributed random individuals in the design space and take the best according to the reduced model.

Informed mutation: To do mutation, several random mutations are generated of the base point. Each random mutation is generated according to the regular method used in GADO. The mutation that appears best according to the reduced model is returned as the result of the mutation.

Informed crossover: To do crossover two parents are selected at random according to the usual selection strategy in GADO. Several crossovers are conducted by randomly selecting a crossover method, randomly selecting its internal parameters and applying it to the two parents to generate a potential child. Informed mutation is applied to every potential child, and the best among the mutations is the outcome of the informed crossover.

Reduced model formation

Our reduced model formation method is based on maintaining a large sample of the points encountered during the optimization. When the sample reaches its maximum size, new points replace existing points at random.

We keep the sample divided into clusters. Starting with one cluster, we introduce one more cluster every specific number of iterations. The reason we introduce the clusters incrementally rather than from the beginning is that this way results in more uniform sized clusters. Every new point entering the sample either becomes a new cluster (if it is time to introduce a cluster) or joins one of the existing clusters. A point belongs to the cluster whose center is closest in Euclidean distance to the point at the time in which the point joined the sample. We use clustering because it makes it possible to fit discontinuous and complicated surfaces with simpler surfaces such as quadratic approximations.

We distinguish between the approximation functions for the measure of merit and those for the sum of constraints (for a detailed discussion please refer to [10]). We use quadratic approximation functions of the form:

$$\hat{F}(\overline{X}) = a_0 + \sum_{i=1}^{n} a_i x_i + \sum_{i=1,\, j=i}^{n,\, n} a_{ij} x_i x_j \qquad (1)$$

Where n is the dimension of the search space and x_i is design variable number i.

B. Objective Exchange Genetic Algorithm for Design Optimization

The main idea of OEGADO is to run several single objective GAs concurrently. Each GA 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. Each single objective GA in OEGADO uses least squares approximation to form a reduced model of its own objective. Every GA exchanges its own reduced model with 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, mutation and crossover are replaced by informed operators (IOs). As described above, 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 using the regular replacement strategy.
6. Steps 2 through 5 are repeated until the maximum number of objective evaluations is exhausted.

OEGADO can be extended for the general case of *n* objectives. We implemented (and used in some of the experiments described below) a three-objective version of OEGADO using a round-robin approach in which the exchange of the reduced models takes place as follows:

-Each GA forms its own reduced model as explained earlier.

-After a given interval of evaluations each GA offers its reduced model to one of the other two GAs and obtains one of their reduced models to be used by its informed operators.

-After the second interval each GA exchanges the reduced model with the other remaining GA.

-This process continues and the GAs continue to exchange their reduced models in a round-robin fashion.

It should be noted that OEGADO is not really a multi-objective GA, but several single objective GAs working concurrently to get the Pareto front. 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 other codes to evaluate the remaining objectives. Thus for OEGADO with two objectives:

Total number of objective evaluations = Sum of objective 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 GAs can run asynchronously which is better for objectives having different time complexities. If some objectives are fast, they are not slowed down by the slower objectives. A limitation of our method is that it is impractical when the evaluation of one objective is computationally comparable to the evaluation of all of them. However, we are targeting truly multi-disciplinary domains such as engineering design where each objective is computed by a different simulator or analysis code.

Usually for multi-objective GAs maintaining diversity is a key issue. However we did not need to take any extra measures for diversity maintenance, as the diversity maintenance module already present in GADO seemed to handle this issue effectively.

III. Experimental results

In this section, we first describe the competing methods used for comparison with OEGADO, namely ε-MOEA [5] and NSGA-II [3]. Finally, we discuss the results obtained for various test cases by these three methods.

A. Competing methods for comparison

We decided to compare our approach with two state-of-the-art methods. We describe the two methods briefly below. More detailed descriptions can be found in [5] and [3].

1) *ε-MOEA*

ε-MOEA is a steady-state MOEA developed for the purpose of achieving well-distributed Pareto-optimal solutions in a relatively short computational time. It is based on the ε-dominance concept that does not allow two solutions with a difference less that ε to be considered non-dominated solutions. This concept is the key feature in maintaining population diversity. ε-MOEA maintains two co-evolving populations, namely the EA population and the archive population. The archive population is based on the ε-dominance concept and the EA population is maintained using the usual dominance concept. In each iteration, two solutions (one from each population) are chosen for mating and two offspring solutions are created. Each of the offspring solutions is compared with the archive and the EA population for possible inclusion. Each offspring is compared with each member in the archive for ε-dominance and with the EA for regular dominance. If the offspring dominates any population member then the offspring replaces one of the members it dominates. Otherwise it is not accepted. As the optimization further continues, the final archive members are reported to be the obtained solutions. Thus the ε-MOEA demonstrates an elitist approach with good diversity maintenance.

2) NSGA-II

NSGA-II is a fast non-dominated sorting based multi-objective evolutionary algorithm. It can be viewed as an improved version of NSGA. It proposes a fast non-dominated sorting approach by incorporating a better book-keeping strategy that reduces the complexity involved in the non-dominated sorting procedure in every generation.

NSGA-II sorts a population of size N according to the level of non-domination. Every solution from the population P is checked against a partially filled population P' for domination. If a solution p from population P dominates any member q of P', then solution q is removed from P', else if a solution p is dominated by any member q of P' then p is ignored. A solution p enters P' only if p is not dominated by any member of P'. P' thus contains all the non-dominated solutions. For every subsequent generation the members of P' are discounted from P and the same procedure is reapplied. At the end of the optimization P' contains all the non-dominated solutions.

In addition to the fast non-dominated sorting, diversity maintenance in NSGA-II is carried out by the use of the crowded comparison operator. This operator guides the selection process at various stages by ranking the non-dominated solutions based on their crowding distances.

Due to its low computational requirements, elitist features and constraint handling capacity, NSGA-II has been successfully used in many applications. It proved to be better than many other multi-objective optimization GAs.

B. Test Problems

The test problems for evaluating the performance of our method were chosen based on significant past studies. We compared the performance of OEGADO, ε-MOEA and NSGA-II on a number of test problems having constraints with varying degrees of difficulty and having two or three objectives. We chose two two-objective and two three-objective problems from the benchmark domains commonly used in multi-objective GA research and two two-objective problems from the engineering domains for a total of six problems.

The problems chosen from the benchmark domains are TNK suggested by Tanaka [14], OSY used by Osyczka and Kundu [8], Rémy [13] used by Coello and Cortés[1] and DTLZ8 proposed by Deb[5]. The problems chosen from the engineering domains are Two-Bar Truss Design used by Deb [6] and Welded Beam design used by Deb [6]. All these problems are constrained multi-objective problems.

For OEGADO, the population size was set to the default value recommended in [9] which is ten times the dimension of the problem except for the DTLZ8 problem where the population size was set to half that value. For ε-MOEA the EA population size was set to 10 and the archive population was set 100 as recommended in [5]. For NSGA-II the population size was fixed to 100 as recommended in [4] except for the DTLZ8 problem where the population size was set to 150.

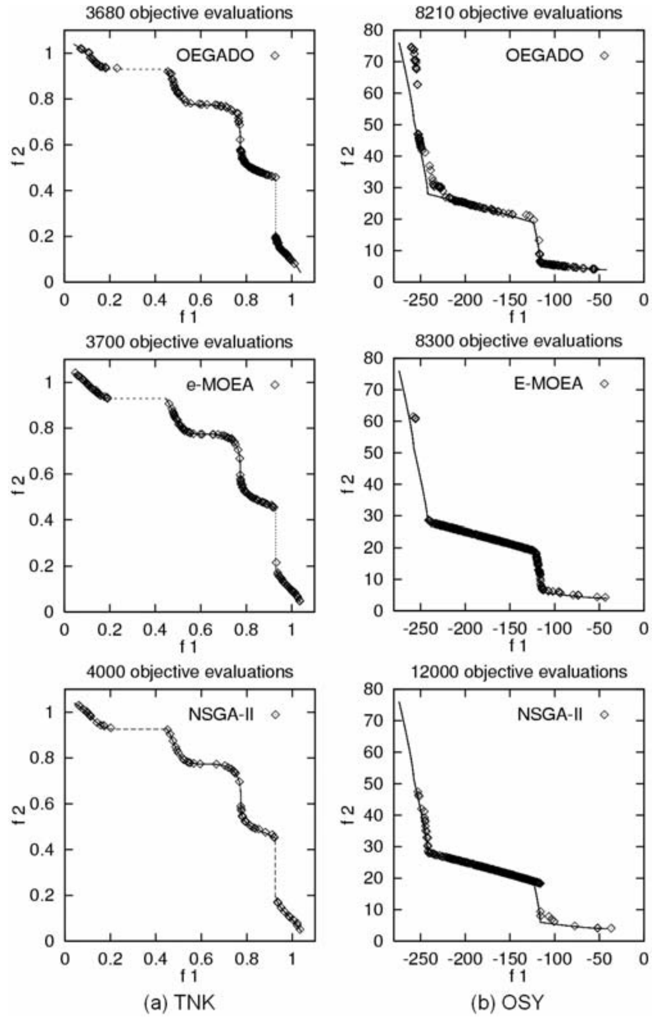

Fig. 1 Results of the TNK problem and the OSY problem

Since we did not know exactly how many evaluations would be required by OEGADO before hand (as it depends on the number of feasible points found), to give fair treatment to NSGA-II, we set the number of generations of NSGA-II liberally. In effect NSGA-II ended up doing significantly more evaluations than OEGADO. We however did not decrease the number of generations for NSGA-II and repeat the experiments as our method outperformed it in most domains anyway. We ran the ε-MOEA experiments last and gave it slightly more evaluations than OEGADO. No attempt was made to control or measure the actual CPU time of the different methods because in real-world applications the number of evaluations usually dominates any other book-keeping overhead [9, 12].

C. Results

Figures 1-3 present the graphical results of all three methods for all problems. Following the experimental methodology proposed in [15], the outcomes of five runs using different seeds were combined and then the non-dominated solutions were selected and plotted from the union set for each method. We used graphical representations of the Pareto fronts found by the three methods to compare their performance. We also indicated the average number of objective evaluations on each graph.

The TNK problem (Fig. 1a) and the OSY problem (Fig. 1b) 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 regions that can demand a GA to maintain its population at different intersections of the constraint boundaries. As it can be seen from the above graphs for the TNK problem, within a comparable number of fitness evaluations, the three methods performed well. OEGADO and ε-MOEA however gave a denser sampling near the Pareto-region than NSGA-II.

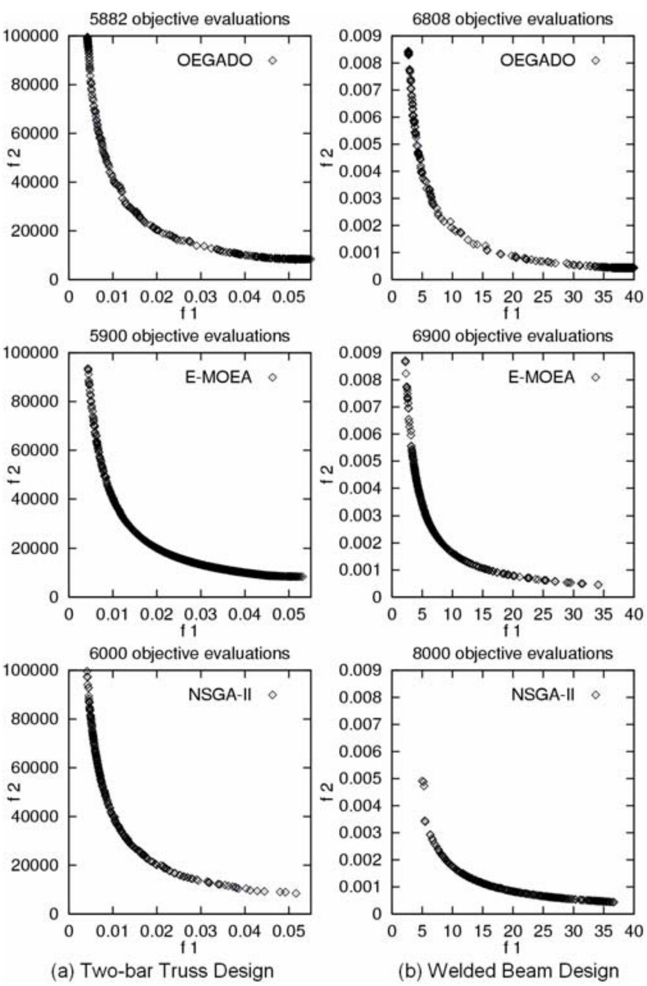

Fig. 2 Results for the Engineering design problems

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. The ε-MOEA gave a very good sampling at the mid-section of the curve but gave a very poor distribution at the extreme ends of the curve. NSGA-II did not give a good sampling of points at the extreme ends of the Pareto front and gave a poor distribution of the Pareto optimal solutions. In this problem OEGADO outperformed ε-MOEA and NSGA-II while running fewer objective evaluations.

For the Two-bar Truss design problem (Fig. 2a), ε-MOEA performed very well on the mid-section of the curve but not on the extremes. OEGADO however gave a good distribution of points on the whole curve especially on the extremes. NSGA-II performed well on one side of the curve but not the other.

In the Welded Beam design problem (Fig. 2b), the non-linear constraints can cause difficulties in finding the Pareto solutions. The figure shows that ε-MOEA performed better than the other methods at the mid section of the Pareto curve. ε-MOEA also found a good enough distribution of points along the Pareto front. OEGADO gave a good distribution of points at the Pareto region but did not give a very dense sampling of points. NSGA-II performed poorly with respect to finding a good distribution of points and only performed well on the second objective.

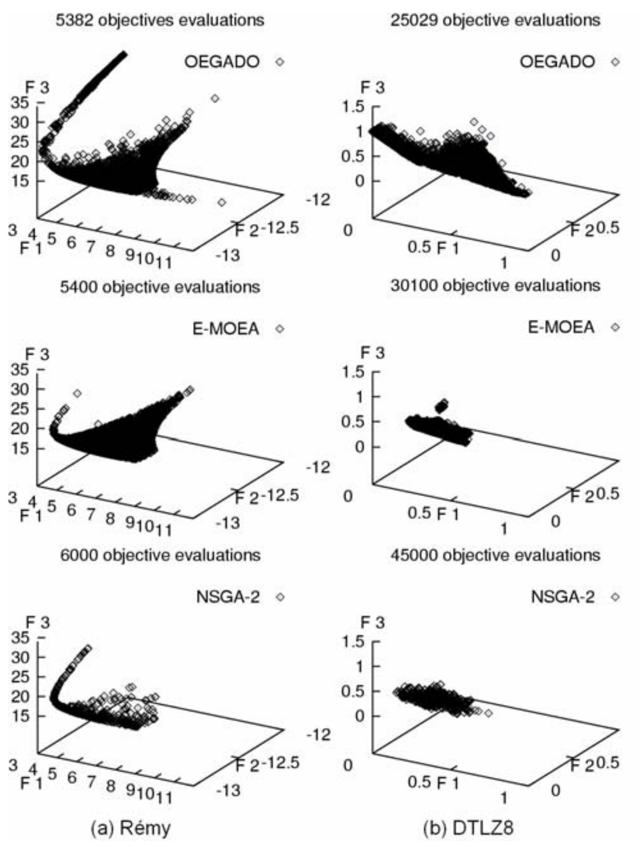

Fig. 3 Results for the three-objective problems

For the Rémy problem (Fig.3a) which is a three-objective problem, it can be seen that OEGADO not only performed well at the middle region of the Pareto-optimal curve in three dimensions but also performed well at the extreme regions. ε-MOEA performed very well only at the middle region. NSGA-II however performed poorly in both the middle and the extreme region. In this problem OEGADO outperformed ε-MOEA and NSGA-II.

The DTLZ8 problem (Fig.3b) is a three-objective, 30-variable problem. The overall Pareto front here is a combination of a straight line segment and a triangle in two different planes. It is a difficult task for any multi-objective evolutionary method to find solutions in both regions and also maintain a good distribution of solutions on the hyper-plane. As seen from Figure 3b, OEGADO found a good distribution of solutions near both the Pareto-optimal line and plane, while ε-MOEA did not give any distribution along the line and gave only a fair distribution on the plane. NSGA-II gave a very poor distribution of solutions on both the line and the plane.

IV. CONCLUSION AND FUTURE WORK

In this paper we presented a novel method for multi-objective GA optimization. We introduced the idea of using reduced models for multi-objective GA optimization wherein each GA runs concurrently and exchanges its reduced model with the other GAs as opposed to maintaining non-dominated solutions. The exchanged reduced models bias each GA towards the other objectives thus directing the optimization towards the Pareto front. Also since each GA runs independently, the method can easily find solutions in the extreme regions as well. In this way OEGADO is able to find a well-distributed Pareto front with fewer objective evaluations. OEGADO is based on concurrent execution of all the GAs which makes it fast and efficient in multi-processing environments. Thus it is a promising method for multi-objective optimization in real world application domains wherein the number of objective evaluations is a critical factor.

We presented a comparison of our method with two reliable, efficient and top of the line methods namely ε-MOEA and NSGA-II. The results show that for the simpler two-objective problems OEGADO performed as well as ε-MOEA and NSGA-II. For the difficult two-objective problems, OEGADO performed better than or as good as ε-MOEA and outperformed NSGA-II in most respects. For the three-objective problems OEGADO performed well in many respects. OEGADO produced a good approximation of the true Pareto front for the difficult three-objective problems, which the other two methods failed to produce. In general, OEGADO demonstrated robustness and efficiency in its performance. Moreover, OEGADO was able to find the Pareto-optimal solutions for all the problems in fewer objective evaluations than ε-MOEA and NSGA-II. For real-world problems, the number of objective evaluations performed can be critical as each objective evaluation takes a long time.

In the future, we would like to extend the implementation of OEGADO to handle more than three objectives by using a weighted sum approach rather than the round-robin approach. In the weighted sum approach we take a weighted sum of the reduced models of the other objectives (i.e. other than the objective being directly optimized by the respective GA). We would further like to explore the capabilities of OEGADO by challenging it with more complex problems with many variables and many objectives from real world application domains.

REFERENCES

[1] C. Coello Coello and N. Cruz Cortés. Solving Multiobjective Optimization Problems using an Artificial Immune system. Technical Report *EVOCINV-05-2002, Evolutionary Computation Group at CINVESTAV, Sección de Computación, Departamento de Ingeniería Eléctrica*, CINVESTAV-IPN, México, December 2002.

[2] D. Crone, J. Knowles and M. Oates (2000). The Pareto Envelope-based Selection Algorithm for Multi-objective Optimization. In *Proceedings of the Parallel Problem Solving from Nature VI Conference,* pp.839-848, Paris, France. Springer. Lecture Notes in Computer Science No. 1917.

[3] K. Deb, 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.

[4] K. Deb. S. Gulati (2001). Design of truss-structures for minimum weight using genetic algorithms, In *Journal of Finite Elements in Analysis and Design,* pp.447 – 465, 2001.

[5] K. Deb, M. Mohan, and S. Mishra. A Fast Multi-objective Evolutionary Algorithm for Finding Well-Spread Pareto-Optimal Solutions. *KanGAL Report No. 2003002*, 2003.

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

[7] Y. Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing Journal*, 2003 (in press).

[8] A. Osycza and S. Kundu (1995). A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm. *Structural Optimization* (10). 94-99.

[9] K. Rasheed. GADO: A genetic algorithm for continuous design optimization. Technical Report DCS-TR-352, Department of Computer Science, Rutgers University, 1998. Ph.D. Thesis, http://www.cs.uga.edu/~khaled/thesis.ps.

[10] K. Rasheed. An incremental-approximate-clustering approach for developing dynamic reduced models for design optimization. In *Proceedings of the Congress on Evolutionary Computation (CEC'2002)*, pp. 986—993, 2000.

[11] K. Rasheed and H. Hirsh. Informed operators: Speeding up genetic-algorithm-based design optimization using reduced models. *In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2000)*, pp. 628--635, 2000.

[12] K. Rasheed and H. Hirsh. Learning to be selective in genetic-algorithm-based design optimization. *Artificial Intelligence in Engineering, Design, Analysis and Manufacturing,* 13:157-169, 1999.

[13] V. Rémy, C. Fontiex and I. Marc. New Multicriteria Optimization Method Based on the Use of a Diploid Genetic Algorithm: Example of an Industrial Problem. In *Proceedings of Artificial Evolution (European Conference, selected papers),* pages 120-127, Brest, France, September 1995. Springer-Verlag.

[14] M. Tanaka (1995). GA-based decision support system for multi-criteria, optimization. In *Proceedings of the International Conference on Systems, Man and Cybernetics-2*, pp. 1556-1561.

[15] E. Zitzler, K. Deb and L. Thiele (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation Journal*, 8(2), 173--195.

[16] E. Zitzler, M. Lauman and L. Thiele (2001). SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, *Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH)* Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland.