Multiobjective Optimization Using a Pareto Differential Evolution Approach

Nateri K. Madavan

NASA Advanced Supercomputing Division
M/S T27A-1, NASA Ames Research Center
Moffett Field, CA 94035-1000, USA
email: madavan@nas.nasa.gov

Abstract - Differential Evolution is a simple, fast, and robust evolutionary algorithm that has proven effective in determining the global optimum for several difficult single-objective optimization problems. In this paper, the Differential Evolution algorithm is extended to multiobjective optimization problems by using a Pareto-based approach. The algorithm performs well when applied to several test optimization problems from the literature.

I. INTRODUCTION

Remarkable progress has been made in recent years in the development of evolutionary algorithms for multiobjective optimization problems (MOPs). MOPs are characterized by the presence of multiple conflicting objectives that must be optimized simultaneously and allow multiple optimal solutions. These multiple solutions are all optimal in the sense that there are no other solutions in the entire solution domain or search space that are superior to them when all objectives are considered simultaneously [1]. These "non-inferior" or nondominated solutions are referred to as Pareto-optimal solutions, and collectively represent the Pareto set or front. Multiobjective evolutionary algorithms (MOEAs) are population-based methods for solving MOPs that have been developed in recent years. These algorithms guide the search process toward the global Pareto-optimal region while maintaining adequate population diversity to capture as many solutions in the Pareto front as possible.

MOEAs can be classified broadly according to their approach as either non-Pareto or Pareto-based [2]. Non-Pareto methods are based on an aggregating approach where the multiple objective functions are combined into a single function allowing single-objective evolutionary algorithms to be then applied [2]. Methods based on various aggregating approaches, such as weighted sum, goal attainment, goal programming, and others, have been developed and applied successfully to several optimization problems. These methods require multiple single-objective optimization runs with different weights for the various objectives in order to find multiple Pareto-optimal solutions. Pareto-based approaches, on the other hand, offer the advantage of generating multiple Pareto solutions simultaneously. These methods are based on a suggestion by Goldberg [3] of using nondominated ranking and selection to evolve a population of solutions toward the Pareto front. Several Pareto-based methods have been developed in recent years (see, for example, [4]-[8]) and used in a variety of applications.

This paper considers the evolutionary strategy (ES) known as Differential Evolution (DE) (see Refs. [9], [10]) that has been applied successfully to a wide range of single-optimization problems [11]. This ES is extended to multiobjective problems using a Pareto-based approach in this paper. Most MOEAs in the literature are based on genetic algorithms (GAs). These include the nondominated sorting genetic algorithm (NSGA) of Srinivas and Deb [4], the FFES algorithm of Fonseca and Fleming [12], the NPGA algorithm of Horn et al. [6], the NSGA-II algorithm of Deb et al. [8], and the SPEA algorithm of Zitzler and Thiele [7], among others. There has been limited use of other evolutionary approaches such as evolutionary strategies for multiobjective optimization. The Pareto-archived evolutionary strategy (PAES) and its successors are among the few noteworthy ES-based approaches [5]. Thus, a need exists for more research in ES-based MOEAs such as the effort reported here. Although the lines between GAs and ESs have blurred in recent years as a result of a healthy cross-pollination of ideas between them and with the development of real-coded GAs, basic differences between the two approaches still remain. Perhaps the main ideological difference lies in the relative importance given to the two main evolutionary operators, recombination (crossover) and mutation, with GA-based approaches relying heavily on the former and ES-based approaches on the latter [14].

The Pareto Differential Evolution method described in this paper modifies the selection procedure in the basic DE algorithm by incorporating the fast nondominated sorting and ranking selection scheme of Deb et al. [8]. In effect, it combines the robust and effective DE strategy with key elements of the successful NSGA-II algorithm. The result is a simple and powerful evolutionary strategy that is self-adaptive and elitist, and can maintain diversity in the Pareto set. Results are presented for several difficult test problems from the literature to demonstrate the efficacy and capabilities of the method.

It should be mentioned that Abbass et al. [13] have also recently reported preliminary results from a Pareto-frontier Differential Evolution method. The basic idea in the present work and in [13] is similar in that both extend the basic DE algorithm to multi-objective problems by incorporating a nondominated sorting and ranking selection procedure. However, the manner in which this idea is implemented here differs from [13] and represents an alternate approach to multiobjective optimization using DE.

0-7803-7282-4/02/$10.00 ©2002 IEEE

II. DIFFERENTIAL EVOLUTION

Differential Evolution is an ES-based approach developed for single-objective optimization in continuous search spaces. It is conceptually simple and possesses good convergence properties that have been demonstrated in a variety of applications [11]. Details of the algorithm can be found elsewhere [9],[10]; only its main features are summarized here.

The approach uses a population $\boldsymbol{P}^Q$ that contains N K-dimensional real-valued parameter vectors in generation Q, where K is the number of parameters or decision variables:

$$\boldsymbol{P}^Q = P_i^Q = p_{i,k}^Q, \quad i = 1,N; \quad k = 1,K$$

The population is usually initialized in a random fashion and the population size N is maintained constant throughout the optimization process. Differential evolution is thus similar to a (μ, λ) ES [14] with μ and λ equal to N [15]. The method however differs from standard ES approaches in several respects as described below.

As with all ES-based approaches, mutation is the key ingredient of differential evolution. The basic idea is to generate new parameter vectors for the subsequent generation by using weighted differences between two (or more) parameter vectors selected randomly from the current population to provide appropriately scaled perturbations that modify another parameter vector (or, comparison vector) selected from the same population. This idea has been implemented in various forms but the form discussed and used here is the classical implementation where new trial parameter vectors $\tilde{P}_i^{Q+1}$ for the next generation $Q+1$ are generated according to the following mutation scheme:

$$\tilde{p}_{i,k}^{Q+1} = p_{i,k}^Q + F \cdot (p_{r_1,k}^Q - p_{r_2,k}^Q) \quad i = 1,N; \quad k = 1,K$$

where the integers r_1 and r_2 are chosen randomly in the range $[1,N]$. The random integers are pairwise different and differ from the running index i. The mutation parameter $F \in [0,2]$ is a real, constant, user-supplied parameter that controls the amplification of the differential variation. Other variants that either use the difference between more than two parameter vectors or keep track of the best parameter vector at each generation and use it in the mutation scheme have also been developed [9] and used with varying success in specific applications. Thus, differential evolution differs from other ESs most notably in that the mutation operator is derived from the current population and not by probability density functions that are defined separately [15].

DE is similar to other recombinative ES approaches in that it also uses discrete recombination. While various recombination strategies exist [14] the strategy adopted in differential evolution is to modify the trial parameter vectors $\tilde{P}_i^{Q+1}$ as follows:

$$\tilde{p}_{i,k}^{Q+1} = \tilde{p}_{i,k}^{Q+1} \quad if \quad rand(k) \le CR \quad or \quad k = rnbr(i)$$

and,

$$\tilde{p}_{i,k}^{Q+1} = p_{i,k}^Q \quad otherwise$$

In the above, $rand(k) \in [0,1]$ is the k th evaluation of a uniform random number generator, $CR \in [0,1]$ is the crossover parameter, and the value $rnbr(i)$ is a randomly chosen index in $\{1, 2, \ldots\ldots, K\}$ that ensures that $\tilde{p}_{i,k}^{Q+1}$ gets at least one parameter from the mutated $\tilde{p}_{i,k}^{Q+1}$ and not all from $p_{i,k}^Q$. Note also that the mutation and recombination operations described above can lead to new vectors that may fall outside the boundaries of the variables. Various repair rules can be used to ensure that these inadmissible vectors do not enter the population. A simple strategy, which is the one adopted here, is to delete these inadmissible vectors and form new ones until the population is filled.

The selection scheme used in DE is deterministic but differs from methods usually employed in standard ES approaches. Selection is based on local competition only, with the child trial vector (or, child) $\tilde{P}_i^{Q+1}$ competing against one population member P_i^Q (the comparison vector) and the survivor entering the new population P_i^{Q+1}. In other words, if $\tilde{P}_i^{Q+1}$ yields a better objective function value than P_i^Q then P_i^{Q+1} is set to $\tilde{P}_i^{Q+1}$. Otherwise, the old value P_i^Q is retained. This greedy selection criterion results in fast convergence; the adaptive nature of the mutation operator, in general, helps safeguard against premature convergence and allows the process to extricate itself from local optima.

II. PARETO-BASED DIFFERENTIAL EVOLUTION

The DE strategy can be extended to multiobjective optimization by adopting either a non-Pareto or Pareto-based approach. The non-Pareto approach is quite straightforward and requires no modifications to the basic algorithm. A single objective function needs to be formed from the multiple objectives and any of several aggregating approaches can be used. A Pareto-based approach such as the one adopted here, on the other hand, requires modifications to the basic algorithm that will allow a population of Pareto-optimal solutions to be determined simultaneously in a single simulation run.

The Pareto-based Differential Evolution algorithm developed here differs from the basic algorithm primarily in the selection procedure used to pick subsequent generations of the population. We have opted to use the nondominated sorting and ranking selection procedure developed by Deb et al. [8]. This procedure has been shown to be very effective in guiding the search toward the global Pareto front for several difficult optimization problems. It is implemented here in the following manner. Once new candidate parameter vectors are obtained using the DE mutation and recombination operators, the new population is combined with the existing parent population and a nondominated sorting and ranking of the combined population is performed. Note that the size of the combined population is $2N$. As noted in [8], this allows a global non-domination check among both the parent and offspring solutions although it requires additional computational effort in sorting the combined rather than only the offspring population as is done in other approaches. At the end of the nondominated sort all members of the combined population (both parent and new candidate parameter vectors) have been assigned a nondominated rank.

The NSGA-II algorithm incorporates both an elite-preserving and an explicit diversity-preserving strategy that are very effective. An explicit diversity-preservation or niching strategy is used to assign a diversity rank to all the members that are in the same nondominated front and thus have the same nondominated rank. The solutions within each nondominated front that reside in the least crowded region in that front are assigned a higher rank. A crowding distance metric is used to estimate the density of solutions surrounding a particular solution in the population and is obtained from the average distance of the two solutions on either side of the solution along each of the objectives. One of the reasons this particular niching strategy was chosen here is because it does not require any external parameters. Details can be found elsewhere [8].

Once the nondominated rank and diversity rank of all the combined population members was determined as above, different approaches to picking the members of the next generation were attempted. The first approach follows the lines of the basic DE algorithm where each child vector is compared with one of the parent population members (the comparison vector), P_i^Q , that was used to generate it. The child $\tilde{P}_i^{Q+1}$ replaces the parent P_i^Q if it has a higher nondominated rank, or, if it has the same nondominated rank and a higher diversity rank. The process is then repeated for subsequent generations. This is in a sense the crowded tournament selection operator in the NSGA-II procedure. When applied to several test problems this method worked well in terms of maintaining diversity among the solutions; however, the method was inefficient in that convergence was slow. This is to be expected, since the elitism in this approach is minimal and good nondominated solutions may be deleted in the selection process while poorer solutions survive. Some efficiency improvements were noted by increasing the elitism by restricting mating only to members of the highest nondominated set. The approach that worked best was to follow the NSGA-II procedure closely and pick the best N members out of the combined $2N$ population by filling the population from the best nondominated rank down. The diversity rank is used to pick the best members of the last nondominated rank that can be partially accommodated in the population of size N. This approach is more elitist and it is not surprising that it was found to be more efficient. An added advantage is that when the combined population includes several nondominated ranks, the diversity ranking needs to be computed only for the members of one nondominated rank, i.e., the last rank that can be partially accommodated [8]. This advantage usually only holds in the early phase of the simulation; in the latter phases, most population members belong to one or two nondominated sets.

The Pareto Differential Evolution method described above in effect combines the robust and effective DE mutation and crossover operators with the fast nondominated ranking scheme and diversity preservation strategy of the highly successful NSGA-II algorithm. The result is a simple and powerful evolutionary strategy that is self-adaptive and elitist, and can maintain diversity in the Pareto set. The results obtained here in the Results section demonstrate the capabilities of the method.

It is worth mentioning that elitism in the NSGA-II algorithm is emphasized in two ways. In addition to the crowding distance niching strategy, a crowded tournament selection operator is also used to ensure diversity on the current nondominated front. It was noted that this dual emphasis on elitist solutions could cause non-elitist solutions to be deleted rapidly resulting in poor convergence [16] and a method was suggested to control the extent of elitism by ensuring that individuals from several nondominated fronts are maintained in the population. In the present method elitism is introduced in only one way, since we do not use the tournament selection operator. Preliminary tests indicate that the elitism present in the method adequately provides diversity without causing convergence problems. Modifications to the present method to include tournament selection or controlled elitism can of course be done easily.

In contrast to the present method, the Pareto-frontier Differential Evolution method of Abbass et al. [13] uses a different approach to creating the offspring population from the parent. Only nondominated solutions are allowed to participate in reproduction. A nondomination check is first performed on the parent population. The nondominated solutions are retained and all the other solutions are removed from the population. If the number of nondominated solutions exceed a threshold number (e.g., half the total population) a distance metric relation is used to remove parents that are in close proximity to each other. The basic differential evolution procedure is used to generate the offspring from three parents chosen at random. The offspring enters the population only if it dominates the first selected (or comparison) parent; otherwise new parents are selected. This continues until the population is filled. The entire process is then repeated for subsequent generations. Preliminary results show the method to be effective. Detailed comparisons between this and the present approach need to be performed before their relative performance can be evaluated.

IV. RESULTS

The algorithm is tested on the test suite of problems (MOP1-7) described in Veldhuizen [17]. These test problems have been aggregated from the published literature and include various relevant features, such as, concave and convex Pareto-optimal fronts, continuous and discontinuous fronts, symmetric and unsymmetrical fronts, scalable decision variables, and multiple (more than two) objective functions. Many of these features may cause difficulties to an MOEA.

While a detailed evaluation of the algorithm is currently under investigation some preliminary results are presented here showing the ability of the algorithm to converge to the optimal Pareto solutions and maintain adequate diversity in the final population. Qualitative results from typical simulation runs on each of the test problems are shown. In order to obtain more quantitative measures of algorithm performance both in terms of the proximity to the true Pareto front that is achieved as well as in terms of the diversity of the optimal solutions we have used some of the methods described in [1]. The generational distance metric and the diversity metric measures are computed for some of the test problems.

For all the test problems a population of size 100 is used and each simulation was carried out to 250 generations. Based on various trial simulations, both the mutation factor and the crossover factor were chosen as 0.3. No effort was made to optimize these values for the various problems. It is interesting to note that, in general, lower values of the mutation and crossover parameter were found preferable. This is in contrast to the much higher values of these parameters that are recommended in the literature when using the DE algorithm for single-objective optimization. Similar findings were reported by [13].

MOP1 is based on Schaffer's objective function (see [17] or [1] and the references therein for details). This is a relatively easy problem because there is only one decision variable, and the Pareto front is convex. Here, following [17] and others, we increase the problem difficulty by using very large decision variable bounds (-10^5 to 10^5). Figure 1 shows qualitative results using the current algorithm. In this and subsequent figures, the axes f_1, f_2, f_3 denote the objective functions and the circular symbols represent the results obtained using the present method. The Pareto front is well predicted and a large number of optimal solutions are obtained that are spread out over the entire front. MOP2 is the two-variable formulation of Fonseca and Fleming (see [17]). This objective function is scalable in the decision variables. The Pareto front is a single concave curve and Fig. 2 shows that the present method effectively finds diverse solutions along this front. Unlike all the other test problems, MOP3 is a maximization problem. It is the two-objective function problem of Poloni (see [17]) with two non convex Pareto fronts that are disconnected in both the objective and decision variable spaces. Figure 3 shows that the present method is able to predict the two disconnected Pareto fronts that lie on the boundaries of the search space. MOP4 is Kursawe's two-objective function problem where the true Pareto front is made up of three disconnected curves. This is a particularly difficult problem and its solution mapping into dominated objective space is quite convoluted. The present method performs reasonably well as shown in Fig. 4 but required a larger population to converge. The method can capture the distinct optimal solution at (-20, 0) that corresponds to zero values for all the decision variables only sporadically, but otherwise performs well elsewhere along the Pareto front.

MOP5 is a tri-objective function problem, Viennet's third problem (see [17]). The actual Pareto front is a single convoluted three-dimensional curve Figure 5 shows the predicted Pareto front in three-dimensional objective space obtained by the present method. Figure 6 shows a two-dimensional plot in $f_1 - f_3$ space of the same results shown in Fig. 5. Both figures show good agreement with the results in [17] and [1], except for some details in the region around $f_1 = 8$. MOP6 is a problem constructed using the methodology of Deb [1], where the Pareto front is made up of four disconnected curves. The solution mapping of this problem into dominated objective space is not as convoluted as for MOP4, and the present method converges to the true Pareto front quite easily, as shown in Fig. 6.

MOP7 is another tri-objective function problem, Viennet's second problem, that complements MOP5. Its Pareto front is a disconnected surface. Figure 8 shows that the present method performs reasonably well, although close comparison with the results in [17] show that some optimal solutions are not captured in the present method. This issue is being investigated further.

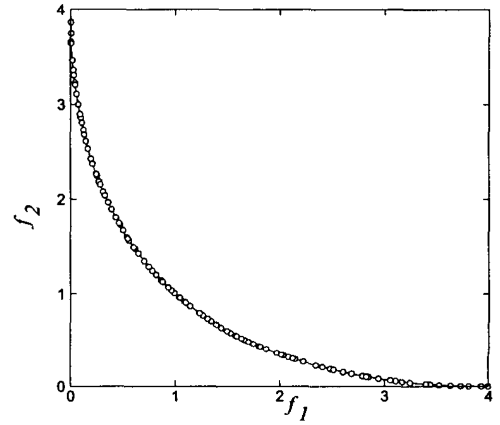

Figure 1. Pareto-optimal solutions in the objective space for test problem MOP1. The solid line marks the location of the true Pareto front.

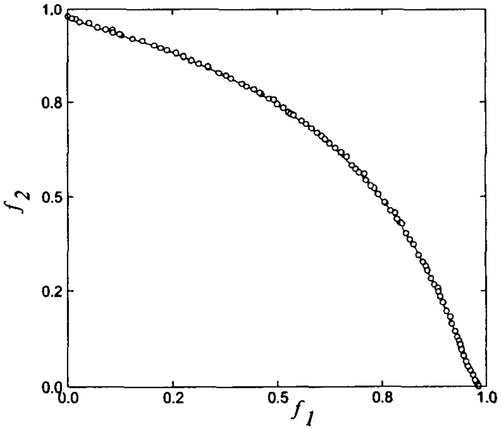

Figure 2. Pareto-optimal solutions in the objective space for test problem MOP2. The solid line marks the location of the true Pareto-optimal front.

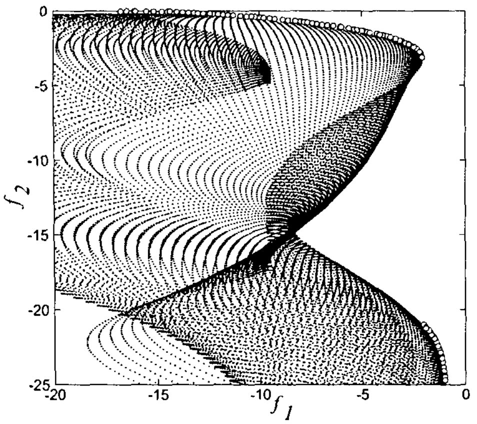

Figure 3. Pareto-optimal solutions in the three-dimensional objective space for test problem MOP3. The dots represent solutions obtained on a 200 x 200 grid of uniformly distributed points in parameter space.

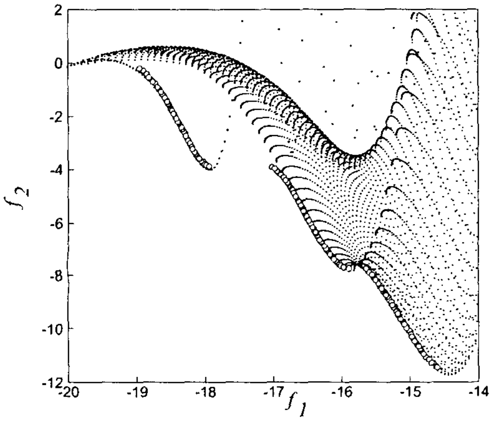

Figure 4. Pareto-optimal solutions in the objective space for test problem MOP4. The dots represent solutions obtained on a 200 x 200 grid of uniformly distributed points in parameter space.

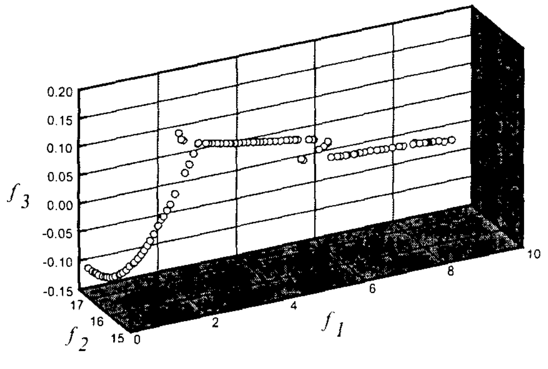

Figure 5. Pareto-optimal solutions in the three-dimensional objective space for test problem MOP5.

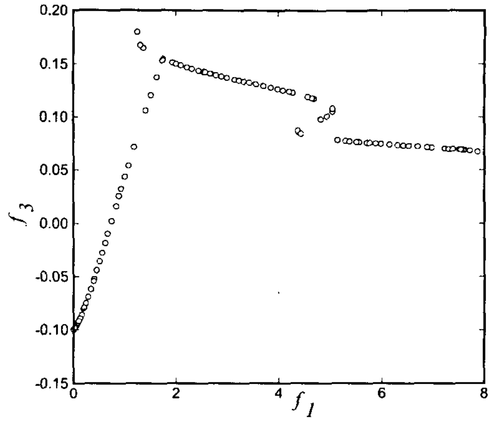

Figure 6. Pareto-optimal solutions in two dimensions (1 and 3) of the objective space for test problem MOP5.

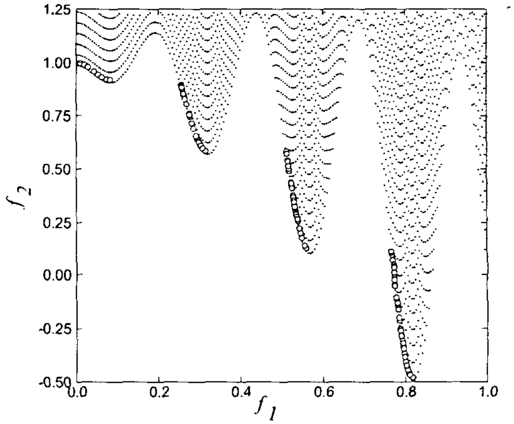

Figure 7. Pareto-optimal solutions in the three-dimensional objective space for test problem MOP6. The dots represent solutions obtained on a 200 x 200 grid of uniformly distributed points in parameter space.

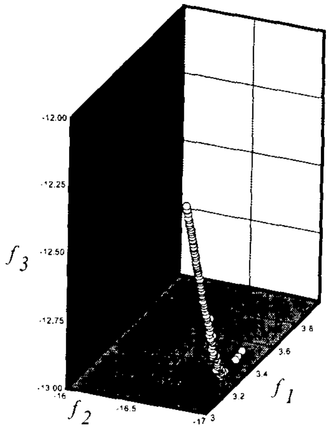

Figure 8. Pareto-optimal solutions in the three-dimensional objective space for test problem MOP7.

The results presented above demonstrate the qualitative performance of the present method on several difficult test problems. In order to obtain more quantitative measures of algorithm performance we have used some of the methods described in [1]. Algorithm performance is measured here both in terms of the proximity to the true Pareto front that is achieved as well as in terms of the diversity of the optimal solutions. The generational distance metric and the diversity metric measures are computed for some of the test problems. The means and variances of these measures are evaluated by conducting 10 distinct runs of each simulation. The results are tabulated in Table 1. In addition to the problems mentioned above, tests were performed for additional problems ZDT1-6 described in [1]. The results, when compared to similar studies of other algorithms [1], show the present algorithm is quite effective and performs well. Table 1 shows that the generational distance metric is large for both MOP4 and ZDT4, indicating that the algorithm has trouble converging to the true Pareto front for these two problems. More detailed comparisons with other algorithms are currently under investigation.

MOP	Generational Distance Metric, Mean	Generational Distance Metric, Variance	Diversity Metric, Mean	Diversity Metric, Variance
MOP1	0.0024544	0.0000001	0.561827	0.001250
MOP2	0.0003246	0.0000005	0.475401	0.000413
MOP3	0.0026627	0.0000008	0.483224	0.000364
MOP4	0.0314712	0.0064362	0.523455	0.008436
MOP6	0.0034270	0.0000084	0.276347	0.000620
ZDT1	0.0006151	0.0000007	0.298567	0.000742
ZDT2	0.0006516	0.0000006	0.317958	0.001389
ZDT3	0.0005631	0.0000003	0.623812	0.000225
ZDT4	0.6182576	0.8268814	0.840852	0.035741
ZDT6	0.0238862	0.0032943	0.473074	0.021721

Table 1. Mean and variances of the generational distance and diversity metrics for the present method on different MOPs.

SUMMARY AND CONCLUSIONS

The Differential Evolution method has been extended to tackle multi-objective optimization problems using a Pareto-based approach that incorporates a nondominated sorting and ranking selection procedure based on Deb et al. [8]. The approach described here in effect combines the robust and effective DE strategy with key elements of the NSGA-II algorithm. The result is a simple and powerful evolutionary strategy that is self-adaptive, elitist and can maintain diversity in the Pareto set. The mutation and crossover operator of the basic DE algorithm that has proven effective in single-objective optimization appears to work well for multi-objective problems also. However, preliminary studies conducted here indicate that the preferred values of the mutation and crossover parameters are much lower than those typically recommended in the DE literature for single-objective optimization. No attempts were made to optimize these parameters. The algorithm is applied to several difficult test problems from the published literature to demonstrate its capabilities and it is shown to perform very well. More detailed analyses of the results and comparisons to other methods are warranted and will be addressed in future work.

REFERENCES

[1] Kalyanmoy Deb, Multi-objective Optimization Using Evolutionary Algorithms, Wiley, 2001.

[2] C. A. C. Coello, "A Comprehensive Survey of evolutionary-based multi-objective optimization techniques," *Knowledge and Information Systems* Vol. 1:3, pp. 269-308, 1999.

[3] David E. Goldberg, Genetic Algorithms for Search, Optimization, and Machine Learning, Addison-Wesley, 1991.

[4] N. Srinivas and K. Deb, "Multi-objective Function Optimization Using Non-dominated Sorting Genetic Algorithms," *Evolutionary Computation*, Vol. 2:3, pp. 221-248, 1994.

[5] J. D. Knowles and D. W. Corne, "The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Multiobjective Optimization," Proceedings of the Congress on Evolutionary Computation 1999, pp. 98-105, Washington, D.C., July 1999.

[6] J. Horn, N. Nafploitis and D. E. Goldberg, "A Niched Pareto Genetic Algorithm for Multiobjective Optimization," Proceedings of the First IEEE Conference on Evolutionary Computation, Vol. 1, pp. 82-87, Piscataway, New Jersey, June 1994.

[7] E. Zitzler and L Thiele, "An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach," Tech. Report 43, Computer Engineering and Communication Networks Lab., Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, May 1998.

[8] K. Deb, S. Agrawal, A. Pratap and T. Meyarivan, "A Fast Elitist Non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," Proceedings of the Parallel Problem Solving from Nature VI Conference, pp. 849-858, Paris, France, 16-20 September, 2000.

[9] R. Storn, and K. Price, "Differential Evolution - A simple evolution strategy for fast optimization," *Dr. Dobb's Journal* Vol. 22:4, pp. 18-24, April 1997.

[10] K. V. Price, "Differential Evolution: A Fast and Simple Numerical Optimizer," In Biennial Conference of the North American Fuzzy Information Processing Society, (NAFIPS), pp. 525-527, M. Smith, M. Lee, J. Keller, J. Yen, Eds., pp. 524-527, IEEE Press, New York, June 1996.

[11] J. Lampinen, "A Bibliography of Differential Evolution Algorithm," Technical Report. Lappeenranta University of Technology, Department of Information Technology, Laboratory of Information Processing, Lappeenranta, Finland, 2001.(http://www.lut.fi/~jlampine/debiblio.htm).

[12] C. M. Fonseca and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evolutionary Computation*, Vol 3:1, pp. 1-16, 1995.

[13] H. A. Abbass, R. Sarker and C. Newton, "PDE: A Pareto-frontier Differential Evolution Approach for Multi-objective Optimization Problems," Proceedings of the Congress on Evolutionary Computation 2001, Vol. 2, pp. 971--978, Piscataway, New Jersey, May 2001.

[14] T. Back, U. Hammel, and H.-P. Schwefel, "Evolutionary Computation: Comments on the History and Current State," *IEEE Transactions on Evolutionary Computation,* Vol. 1:1, pp. 3-17, 1997.

[15] M. A. Shokrollahi and R. Storn,"Design of Efficient Erasure Codes with Differential Evolution," In: Proceedings of ISIT 2000, International Symposium on Information Theory, Sorrento, Italy, June 25-30, 2000.

[16] K. Deb and T. Goel, "Controlled Elitist Non-Dominated Sorting Genetic Algorithms for Better Convergence," Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO-2001), pp. 67-81, Zurich Switzerland, March 7-9. 2001.

[17] D. A. Van Veldhuizen, Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations. Ph.D. thesis, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Tech., Wright-Patterson AFB, Ohio, May 1999.