Incorporating Directional Information within a Differential Evolution Algorithm for Multi-objective Optimization

Antony W. Iorio
School of Computer Science and I.T.,
RMIT University
Melbourne VIC 3000 Australia
iantony@cs.rmit.edu.au

Xiaodong Li
School of Computer Science and I.T.,
RMITUniversity
Melbourne VIC 3000 Australia
xiaodong@cs.rmit.edu.au

ABSTRACT

The field of Differential Evolution (DE) has demonstrated important advantages in single objective optimization. To date, no previous research has explored how the unique characteristics of DE can be applied to multi-objective optimization. This paper explains and demonstrates how DE can provide advantages in multi-objective optimization using directional information. We present three novel DE variants for multi-objective optimization, and a report of their performance on four multi-objective problems with different characteristics. The DE variants are compared with the NSGA-II (Non-dominated Sorting Genetic Algorithm). The results suggest that directional information yields improvements in convergence speed and spread of solutions.

Categories and Subject Descriptors

G.1.6 [**Mathematics of Computing**]: Numerical Analysis—*Optimization*

General Terms

Algorithms

Keywords

Differential Evolution, Multi-objective Optimization

1. INTRODUCTION

DE is a population based optimization algorithm, which has globally and locally correlated step-sizes. The step-sizes self-adapt over time in relation to the location of the population of individuals in the search space. This results in an extremely efficient search. DE is also invariant under a rotation transformation of the decision space [15]. As a result, it can manage problems with non-separable parameters. Many real-world problems have such parameter interactions, and are not aligned with the principle coordinate system in the

decision space [18]. This makes DE an ideal choice for such problems. Evolutionary Strategies have these properties as well, but unlike DE, have to learn the optimal step sizes and the correlation matrix for aligning offspring in the search space. This can be computationally expensive as the dimensionality of the decision space grows [15]. DE doesn't have to learn these properties because they are a property of the differential calculation.

DE has been successfully applied to multi-objective problems, but the application of the actual DE scheme in these cases has been the same as it has been for single objective problems, without leveraging aspects of the multi-objective domain [6, 1, 4, 3, 2, 22, 10, 16]. For instance, with an evolutionary multi-objective algorithm, we are interested in finding a diverse set of non-dominated solutions as close to the Pareto-optimal solutions as possible. With a single objective evolutionary algorithm, we are primarily concerned with finding a single global optimum. The multi-objective problem introduces more complexity to the optimization procedure, because one must find a set of well distributed non-dominated solutions close to the Pareto-optimal front. This presents a challenge for a multi-objective algorithm because it must converge to the Pareto-optimal region, and also maintain diverse solutions.

In this paper, we have proposed three new schemes for DE, which incorporate directional information. They have demonstrated improved performance over a basic DE algorithm on four multi-objective test problems. These new approaches to DE incorporate directional information by selecting parents for the DE calculation according to two measures. These measures are the ranking of individuals within a population by a non-dominated sorting procedure, and a crowding distance calculation which measures the closeness of individuals to each other in the objective space. The ranking of individuals provides an indication as to what direction the search should head in, and the crowding distances suggest what direction is suitable in order to maintain diversity. This provides the directional information to the proposed DE algorithms that will be discussed shortly. Just like DE, the new approaches can be applied to a wide class of problems, and provide a very simple way to direct the search towards the Pareto-optimal set, while also maintaining a diverse set of non-dominated solutions. Previous work has also reported on the use of directional information in a multi-objective evolutionary optimization algorithm [5]. It demonstrated that directional information can improve the performance of an optimization procedure. The estimation

based approach demonstrated in [5] is limited to differentiable optimization problems. The proposed directional information DE approaches in this paper have no such limitations.

In the following section we introduce multi-objective optimization, and discuss DE and previous work in the area of DE multi-objective optimization. In Section 3 we will introduce the proposed multi-objective DE Algorithm which incorporates directional information and discuss the motivation for this approach. The experiments, problems, and parameter settings are described in Section 4, followed by a description of the evaluation criteria in Section 5. Finally, a discussion of the results and concluding remarks are presented in Section 6.

2. BACKGROUND

In this section we will first introduce multi-objective optimization, and the concept of non-domination. This is followed by a description of DE and its application to multi-objective problems.

2.1 Multi-objective Optimization

Multi-objective optimization deals with optimization problems which are formulated with some or possibly all of the objective functions in conflict with each other. Such problems can be formulated as a vector of objective functions $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), .., f_m(\mathbf{x}))$ subject to a vector of input parameters $\mathbf{x} = (x_1, x_2, ..., x_d)$, where m is the number of objectives, and d is the number of parameters. Multi-objective problems have a set of trade-off solutions, where a solution may be better on objective f_1 but worse on objective f_2, whilst other solutions may be worse on objective f_1 but better on objective f_2. Evolutionary multi-objective (EMO) optimization is typically concerned with finding a diverse set of solutions, which is close to the Pareto-optimal solution set. The Pareto-optimal set is the set of solutions which are globally non-dominated with respect to all other feasible solutions.

The criteria for evaluating the performance of a multi-objective evolutionary algorithm are different from those for assessing the performance of single objective algorithms. In this study the non-dominated solution sets resulting from a run of an EMO algorithm are assessed with a metric described in Section 5. This metric can be used to provide an indication of the convergence to the Pareto-optimal set, as well as the diversity of the non-dominated solution set.

2.2 Differential Evolution

DE is a population-based direct-search algorithm for global optimization [14]. It has demonstrated its robustness and power in a variety of applications, such as neural network learning [9], IIR-filter design [19], and the optimization of aerodynamic shapes [17]. It has a number of important characteristics which make it attractive as a global optimization technique, and the reader is referred to [15] for an excellent introduction to DE which covers this in more detail.

DE differs from other EAs in the mutation and recombination phase. Unlike stochastic techniques such as Genetic Algorithms and Evolutionary Strategies, where perturbation occurs in accordance with a random quantity, DE uses weighted differences between decision space vectors to perturb the population.

The DE outlined in Algorithm 1 was adapted for our purposes because it was deemed to be the most suitable, although a large variety of DE approaches exist. Firstly, it has the property of rotational invariance, and secondly it provides two differentials, which are $K(\mathbf{x}_{r3,G} - \mathbf{x}_{i,G})$ and $F(\mathbf{x}_{r1,G} - \mathbf{x}_{r2,G})$. The differentials can be conveniently adapted for generating vectors which point towards the Pareto-optimal set, and generating a diverse set of solutions. This will be elaborated upon further in Section 3.

Algorithm 1 Differential Evolution

Step 1. $i = 1$

Step 2. Randomly select $r_1, r_2, r_3 \in \{1, 2, \cdots, N\}$ such that $r_1 \neq r_2 \neq r_3 \neq i$ where i is the index of the currently selected individual in the population.

Step 3. Generate an offspring $\mathbf{u}_{i,G+1}$ from the selected parents $\mathbf{x}_{r3,G}$, $\mathbf{x}_{r1,G}$, and $\mathbf{x}_{r2,G}$ and current individual $\mathbf{x}_{i,G}$ at generation G, using $\mathbf{u}_{i,G+1} = \mathbf{x}_{i,G} + K(\mathbf{x}_{r3,G} - \mathbf{x}_{i,G}) + F(\mathbf{x}_{r1,G} - \mathbf{x}_{r2,G})$ where K and F are control parameters.

Step 4. $i = i + 1$

Step 5. Repeat from step 2 if the desired number of offspring has not been reached and $i \neq N$.

In the basic DE scheme (Algorithm 1), the population is typically randomly initialised within the parameter bounds for each decision variable. At each generation G, the population undergoes perturbation. The first step initialises the index, i, of the current individual $\mathbf{x}_{i,G}$, to 1. The second step involves three unique individuals denoted by $\mathbf{x}$, with population indices r_1, r_2, and r_3, being randomly selected from a population of size N. The coefficient K is responsible for the level of combination that occurs between $\mathbf{x}_{r3,G}$ and the current individual $\mathbf{x}_{i,G}$. The coefficient F is responsible for scaling the step size resulting from the vector subtraction $\mathbf{x}_{r1,G} - \mathbf{x}_{r2,G}$. At the third step these individuals participate in a vector calculation to generate an offspring, $\mathbf{u}_{i,G+1}$, for the following generation, $G + 1$.

Typically in the single-objective case, if the new individual $\mathbf{u}_{i,G+1}$, evaluates better than the currently selected individual $\mathbf{x}_{i,G}$, then the current individual is replaced with the new one. At the fifth step, a condition is checked to determine if the algorithm should continue to iterate over i from 1 to N, in order to generate the required number of offspring. In the multi-objective DE case, individuals cannot directly replace the parents without either a dominance comparison with the current parent, or a sort of all the offspring with all the parents, with respect to their dominance levels.

2.2.1 Multi-objective Differential Evolution

DE has been applied within a variety of multi-objective algorithms in order to solve multi-objective optimization problems. One of the first applications was to tune a fuzzy controller for the automatic operation of a train, although the cost function transformed the objectives of punctuality, comfort, and energy usage into the degenerate case of a single objective [6]. The Pareto Differential Evolutionary Algorithm (PDE) uses non-dominated solutions for reproduction, and places offspring back into the population if they dominate the current parent [4]. This PDE was also extended into a variant with self-adaptive crossover and mutation [2]. Multi-objective DE has also been applied to minimize the error and the number of hidden units in neural network training. The resulting Pareto-optimal set is the

tradeoff between these two objectives [1]. Another approach involving Pareto-optimal based evaluation has also been applied to an Enterprise Planning problem with the two objectives of cycle time and cost [22]. Preliminary work has also reported on the behaviour of a non-dominated sorting DE multi-objective algorithm [10], demonstrating its potential worth as a rotationally invariant optimizer for problems with non-separable parameters. The use of DE on non-separable multi-objective problems has also been reported in [16].

3. DIRECTIONAL INFORMATION

There are two desirable features of any population based multi-objective optimization procedure. Firstly, it must find a non-dominated solution set which dominates much of the feasible region of the search space and is close to the Pareto-optimal set, and secondly, it must maintain a diverse non-dominated solution set. Successful algorithms, such as the NSGA-II, address these two criteria by ranking solutions in terms of non-domination, and providing each individual with a crowding distance metric which measures how much each individual contributes to diversity within a dominance rank. For the purposes of this study, we have employed NSGA-II as the basis of each of the algorithms we have proposed. Our directional information DE schemes incorporate the NSGA-II measures of dominance ranking and crowding distance in order to achieve suitable directions for guiding the search.

3.1 The Proposed Framework

Before we describe our approach, it is necessary to describe the NSGA-II [8], which is the basis of the DE variants incorporating directional information. In each of these variants we have replaced the crossover operation within the NSGA-II with a DE operation.

The NSGA-II produces N offspring from a parent population of size N. The combined population of size $2N$ is sorted into separate non-domination levels. Individuals are selected from this combined population to be inserted into the new population, based on their non-domination level and how closely individuals are crowded together. If there are more individuals in the last front than there are slots remaining in the new population of size N, a diversity preserving mechanism is used. Individuals from this last front are placed in the new population based on how much they contribute to diversity in that front. The algorithm then iterates until a termination condition is met (In our application, the maximum number of generations is the specified termination criteria). The NSGA-II uses a real-coded crossover and mutation operator but in the proposed variants, these operators were replaced with a DE scheme.

In the NSGA-II we do not know which individuals are better until all candidates are sorted and assigned a crowding distance and non-domination level. Parents are selected for mating using a tournament selection operator which uses the rank and crowding distance of individuals. Within the NSGA-II framework, the DE variants are used to generate N offspring from the selected parents. The offspring individuals are then evaluated on the objective functions. Following this, they are combined with the parent generation. The combined population is then sorted into dominance ranks, as was mentioned previously. Each individual also has a crowding distance associated with it.
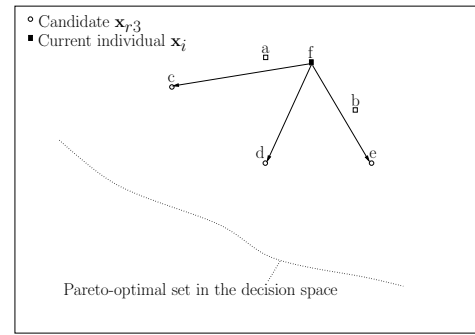

Figure 1: The current individual, f, is ranked worse than the candidates for $\mathbf{x}_{r3,G}$ labelled c, d, and e. The resulting vectors are used to generate offspring in the direction of better ranked solutions.

3.2 Directional Convergence

It is possible to efficiently converge towards the Pareto-optimal set by calculating a differential which creates a vector that will point in the general direction of the Pareto-optimal set, and contribute to the DE calculation. In the basic DE scheme, which does not incorporate such direc-

---

Algorithm 2 NSDE-DC

Step 1. $i = 1$

Step 2. If there exists an r_1 and r_2 such that $r_1 \neq r_2 \neq i$, where $r_1, r_2 \in \{1, 2, \cdots, N\}$, select $\mathbf{x}_{r1,G}$ and $\mathbf{x}_{r2,G}$ for the Differential Evolution calculation. If there exists an r_3 such that the rank of $\mathbf{x}_{r3,G}$ is less than the rank of $\mathbf{x}_{i,G}$, and $r_3 \neq r_2 \neq r_1 \neq i$, where $r_3 \in \{1, 2, \cdots, N\}$, select $\mathbf{x}_{r3,G}$ for the DE calculation. i is the index of the currently selected individual in the population.

Step 3-5 Same as Algorithm 1.

---

tional information, the vector generated can produce offspring which are dominated by individuals in the current population. This can occur because the generated vector could point towards the Pareto-optimal set, or away from the Pareto-optimal set. This wastes valuable objective function evaluations.

Figure 1 shows a number of individuals in the decision space. Individuals c, d, and e are candidates for $\mathbf{x}_{r3,G}$, and are ranked better than the current individual f. As a result of this, the calculation $K(\mathbf{x}_{r3,G} - \mathbf{x}_{i,G})$ generates *convergence vectors* which point towards regions where better ranked individuals are located. K is a scaling factor controlling the magnitude of the *convergence vector*. This figure details how vectors can be suitably generated in order to point towards the Pareto-optimal set.

The procedure of the NSDE-DC (Non-dominated Sorting Differential Evolution with Directional Convergence) approach is outlined in Algorithm 2. It is similar to the basic DE scheme previously described, except there are further constraints on the selection of the parent individuals. In later generations, when the entire population is of the same rank, there is no difference with the basic DE scheme.

3.3 Directional Spread

Directional information for generating offspring along a non-dominated front is possible from the NSDE-DS (Non-
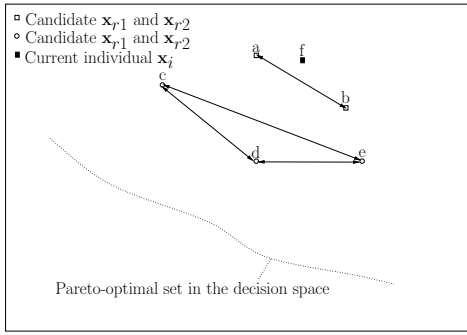
Figure 2: Individuals a and b are of the same rank. Individuals c, d, and e are of the same rank as well. The vectors resulting from taking the difference of individuals of the same rank helps to spread offspring solutions.

Algorithm 3 NSDE-DS

Step 1. $i = 1$
Step 2. If there exists an r_1 and r_2 such that the ranks of $\mathbf{x}_{r1,G}$ and $\mathbf{x}_{r2,G}$ are equal, and $r_1 \neq r_2 \neq i$, where $r_1, r_2 \in \{1, 2, \cdots, N\}$, select $\mathbf{x}_{r1,G}$ and $\mathbf{x}_{r2,G}$ for the DE calculation. If there exists an r_3 such that $r_3 \neq r_2 \neq r_1 \neq i$, where $r_3 \in \{1, 2, \cdots, N\}$, select $\mathbf{x}_{r3,G}$ for the DE calculation. i is the index of the currently selected individual in the population.
Step 3-5 Same as Algorithm 1.

---

dominated Sorting Differential Evolution with Directional Spread) variant. The algorithm attempts to select $\mathbf{x}_{r1,G}$ and $\mathbf{x}_{r2,G}$ from the same rank. The differential *spread vector* resulting from $F(\mathbf{x}_{r1,G} - \mathbf{x}_{r2,G})$, should contribute to the generation of offspring which are spread out across a non-dominated front. Figure 2 shows individuals a and b, which are of the same rank. Individuals c, d, and e are also of an equal, but different rank. Either group can contribute candidates for $\mathbf{x}_{r1,G}$ and $\mathbf{x}_{r2,G}$. F is a scaling factor responsible for controlling the magnitude of the *spread vectors* which are generated.

This component of the DE calculation should in principle help to maintain diversity in the population. In later generations, when the entire population is of the same rank, there is no difference with the basic DE scheme.

The NSDE-DS is also similar to the basic DE approach except for a number of constraints similar to those associated with the NSDE-DC. The procedure for generating offspring from the mating pool is described in Algorithm 3.

3.4 Directional Convergence and Spread

The NSDE-DCS combines both of the features of the NSDE-DC and NSDE-DS variants. Figure 3 shows the possible offspring vectors that could be generated from the current individual f, and the parent candidates, a, b, c, d, and e. From this plot, it is apparent that offspring solutions have a good spread and coverage, and are located in regions of the search space which are in the general direction of the Pareto-optimal set.

As in the previous variants, this is a constrained version of the basic DE scheme described in Section 1. The steps described in Algorithm 4 are followed in order to generate a candidate offspring.
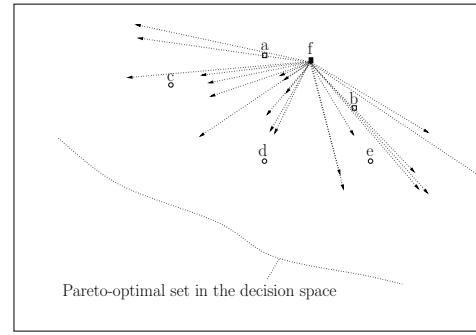

Figure 3: Offspring candidates are generated which are the result of the addition of spread and convergence vectors, to the current individual f.

Algorithm 4 NSDE-DCS

Step 1. $i = 1$
Step 2. If there exists an r_1 and r_2 such that the ranks of $\mathbf{x}_{r1,G}$ and $\mathbf{x}_{r2,G}$ are equal, and $r_1 \neq r_2 \neq i$, where $r_1, r_2 \in \{1, 2, \cdots, N\}$, select $\mathbf{x}_{r1,G}$ and $\mathbf{x}_{r2,G}$ for the DE calculation. If there exists an r_3 such that the rank of $\mathbf{x}_{r3,G}$ is less than the rank of $\mathbf{x}_{i,G}$, and $r_3 \neq r_2 \neq r_1 \neq i$, where $r_3 \in \{1, 2, \cdots, N\}$, select $\mathbf{x}_{r3,G}$ for the DE calculation. i is the index of the currently selected individual in the population.
Step 3-5 Same as Algorithm 1.

---

4. EXPERIMENTS

A population size of 100 individuals was used for each of the algorithms on each of the test problems. For the DE variants, F was set to 0.8 and K was set to 0.4. Suggestions from the literature helped guide our choice of parameter values for the DE variants [15]. As well as the new DE variants incorporating directional information, an implementation of a basic DE with non-dominated sorting (NSDE) [10] is also evaluated for comparative purposes. A mutation rate of 0.1 and crossover rate of 0.9 were also used with the NSGA-II[1]. η_c and η_m are parameters within the NSGA-II which control the distribution of the crossover and mutation probabilities and were assigned values of 10 and 50 respectively.

Experiments were conducted on four test problems. These test problems are described in more detail in [11]. The problems are 10-dimensional in the decision space. Problem R1 is unimodal, originally proposed by Deb [7]. Problem R2 is discontinuous, Problem R3 has a non-uniform mapping between the decision and objective space. Problem R4 is based upon the Schwefel function and is highly deceptive. A new random uniform rotation matrix was generated for each run of each algorithm. These matrices were used to rotate each of the problems in the decision space so that there is no bias for any particular coordinate system [11]. Such a rotation introduces non-separability between decision variables. Each algorithm was run 50 times on each test problem.

---

[1]The variant of NSGA-II used in this study is the original NSGA-II available from the Kanpur Genetic Algorithms Laboratory site at `http://www.iitk.ac.in/kangal/`. It formed the basis of the DE variants studied.

5. EVALUATION CRITERIA

The set of solutions generated by a multi-objective evolutionary algorithm cannot be evaluated with respect to a single measure of performance. For instance, one algorithm may generate a set of solutions which covers the Pareto-optimal solutions well, but has slightly worse convergence to the Pareto-optimal solutions compared with another set which has slightly worse coverage of the Pareto-optimal solutions. A large number of performance metrics have been proposed in the literature, and the reader is referred to a recent comparative survey [24].

In the assessment of the non-dominated sets generated by the algorithms investigated, the Generational Distance metric proposed by [21] is used. This metric, in Equation (1) and (2), measures the average distance of the solutions in the non-dominated solution set Q, to a Pareto-optimal set P^* (Figure 4(a)). The set P^* is a sample of 500 uniformly spaced Pareto-optimal solutions in the objective space. $f_j^{*(k)}$ is the j-th objective function value of the k-th member of the Pareto-optimal set, P^*, and $f_j^{(i)}$ is the j-th objective function value of the i-th member of the set, Q.

$$GD(Q, P^*) = \frac{\sum_{i=1}^{|Q|} d_i}{|Q|} \tag{1}$$

$$where, \; d_i = \min_{k=1}^{|P^*|} \sqrt{\sum_{j=1}^{m}(f_j^{(i)} - f_j^{*(k)})^2} \tag{2}$$

One of the motivations for using this metric, is that it has the elegant property of being able to measure convergence to the Pareto-optimal set, as well as the coverage of a non-dominated solution set over the Pareto-optimal set on the same scale [12]. This can be achieved by swapping the order in which one measures distances between the two sets. With the metric in Equation (3) and (4), we measure the average distance of the solutions in the Pareto-optimal set P^*, to a non-dominated solution set Q. $f_j^{(k)}$ is the j-th objective function value of the k-th member of the non-dominated set, Q, and $f_j^{*(i)}$ is the j-th objective function value of the i-th member of the Pareto-optimal set, P^*. Figure 4(a) demonstrates a relatively good convergence of Q towards P^* with respect to the $GD(Q, P^*)$ metric, where as Figure 4(b), which shows the same points in Q, suggests a poor coverage by set Q over P^* with respect to the $GD(P^*, Q)$ metric. If both measures approach 0 it suggests that the non-dominated set has both a good coverage and convergence to the Pareto-optimal set.

$$GD(P^*, Q) = \frac{\sum_{i=1}^{|P^*|} d_i}{|P^*|} \tag{3}$$

$$where, \; d_i = \min_{k=1}^{|Q|} \sqrt{\sum_{j=1}^{m}(f_j^{*(i)} - f_j^{(k)})^2} \tag{4}$$

6. DISCUSSION AND CONCLUSION

As shown in Figure 6 and 7, at least one of the DE variants incorporating directional information has superior performance to the baseline Differential Evolution algorithm
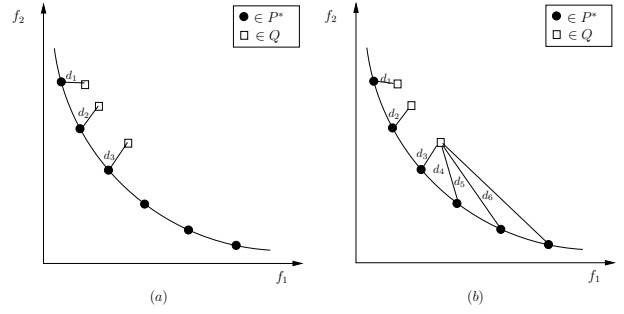

Figure 4: (a) $GD(Q, P^*)$ is a measure of convergence of set Q towards set P^*. (b) $GD(P^*, Q)$ gives a larger value than $GD(Q, P^*)$ because it considers the coverage of the set Q over the set P^* in this specific example.

(NSDE). Secondly, all of the directional information DE approaches outperformed the NSGA-II, with respect to speed of convergence, and spread of solutions across the Pareto-optimal set.

The variant incorporating directional spread information (NSDE-DS) had similar performance with respect to convergence and diversity, to the variant incorporating both directional spread and convergence information (NSDE-DCS). The variant that incorporates only convergence information
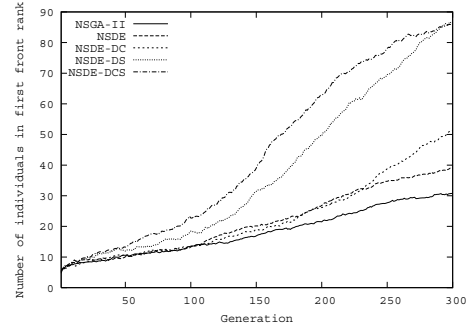

Figure 5: Average number of individuals in the best ranked front averaged over 50 runs over a period of 300 Generations (30,000 problem evaluations).

(NSDE-DC) had marginally better performance over the baseline NSDE. This suggests that directional spread information plays a larger role in improving the speed of convergence towards the Pareto-optimal set than directional convergence information, while also maintaining a high quality of diverse solutions which cover the Pareto-optimal set.

Although the algorithms that do not emphasize convergence or spread generate offspring which contribute to convergence and spread, they do so less efficiently than those incorporating the directional information.

Figure 5 shows the average number of individuals in the best ranked front over the generations. This figure shows that the best DE variants also maintain the largest number of individuals in the best ranked front. The plot presented in Figure 5 for problem R1 was consistently observed for the other problems as well, with the variants incorporating directional spread maintaining the largest number of individuals in the best ranked front.
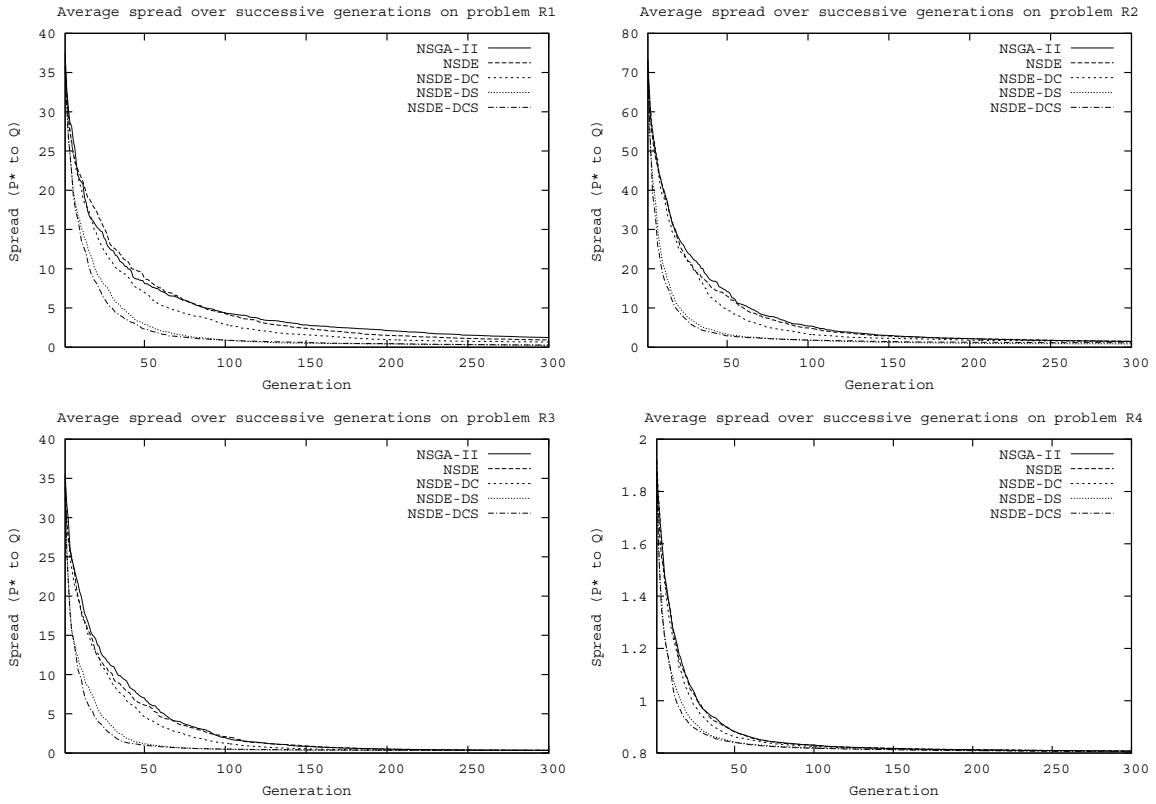
Figure 6: Average spread over 50 runs over a period of 300 Generations (30,000 problem evaluations).
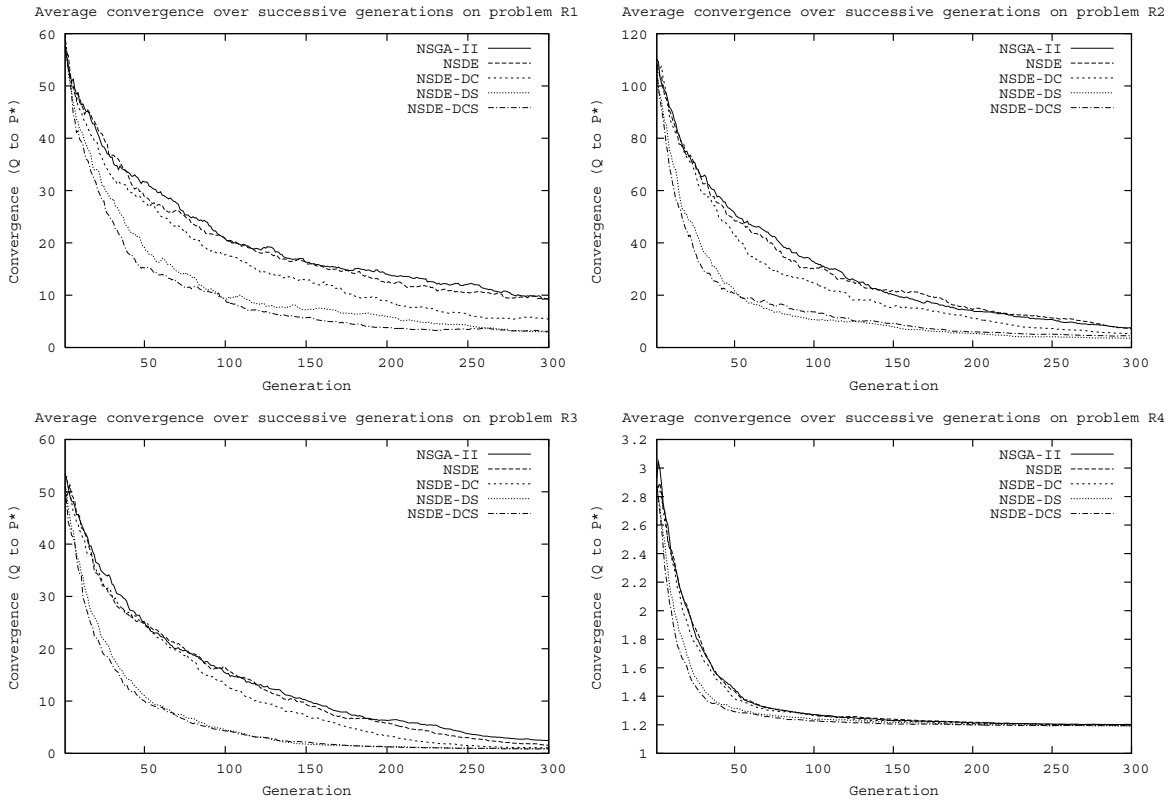

Figure 7: Average convergence over 50 runs over a period of 300 Generations (30,000 problem evaluations).

It may seem non-intuitive that directional information associated with spread can also improve convergence, but there is actually a straight-forward reason for this. If more individuals are maintained along a non-dominated front, there are more regions of the space where new, dominant solutions can be discovered [20].

There are a number of areas which are being considered for further study, such as directing the search towards less crowded regions, and other approaches to incorporating directional information. These preliminary results present compelling evidence for the utility of a directional information approach for multi-objective optimization, which can be simply implemented using DE.

7. REFERENCES

[1] H. A. Abbass. A memetic pareto evolutionary approach to artificial neural networks. In *Proceedings of the Australian Joint Conference on Artificial Intelligence, Adelaide, Australia, Lecture Notes in Artificial Intelligence*, volume 2256, pages 1–12. Springer-Verlag, 2001.

[2] H. A. Abbass. The self-adaptive pareto differential evolution algorithm. In *Proceedings of the 2002 Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 831–836, 2002.

[3] H. A. Abbass and R. Sarker. The pareto differential evolution algorithm. *International Journal on Artificial Intelligence Tools*, 11(4):531–552, 2002.

[4] H. A. Abbass, R. Sarker, and C. Newton. Pde: A pareto-frontier differential evolution approach for multi-objective optimization problems. In *Proceedings of the 2001 Congress on Evolutionary Computation (CEC'2001)*, volume 2, pages 971–978, 2001.

[5] M. Brown and R. E. Smith. Effective use of directional information in multi-objective computation. In *Proceedings of the 2003 Genetic and Evolutionary Computation Conference (GECCO-03)*, pages 778–789, 1999.

[6] C. S. Chang and D. Y. Xu. Differential evolution of fuzzy automatic train operation for mass rapid transit system. *IEEE Proceedings of Electric Power Applications*, 147(3):206–212, 2000.

[7] K. Deb. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation*, 7(3):205–230, 1999.

[8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Trans. Evol. Comput.*, 6(2):182–197, 2002.

[9] J. Ilonen, J.-K. Kamarainen, and J. Lampinen. Differential evolution training algorithm for feed-forward neural networks. *Neural Processing Letters*, 7(1):93–105, 2003.

[10] A. Iorio and X. Li. Solving rotated multi-objective optimization problems using differential evolution. In *AI 2004: Advances in Artificial Intelligence: 17th Australian Joint Conference on Artificial Intelligence, Cairns, Australia*, page 861, Heidelberg, 2004.

[11] A. Iorio and X. Li. Rotated test problems for assessing the performance of multi-objective optimization algorithms. In *Proceedings of the 2006 Genetic and Evolutionary Computation Conference (GECCO-06)*, 2006.

[12] X. Li, J. Branke, and M. Kirley. Performance metrics and particle swarm methods for dynamic. Technical Report TR-05-6, RMIT University, Melbourne Australia, 2005.

[13] N. K. Madavan. Multiobjective optimization using a pareto differential evolution approach. In *Proceedings of the 2002 Congress on Evolutionary Computation (CEC'2002)*, volume 2, pages 1145–1150. IEEE Press, 2002.

[14] K. Price. Differential evolution: a fast and simple numerical optimizer. In *Biennial Conference of the North American Fuzzy Information Processing Society*, volume 3339, pages 524–527, New York, 1996.

[15] K. Price. *Differential Evolution*, chapter 2, pages 79–108. McGraw-Hill, London UK, 1999.

[16] T. Robic. Performance of demo on new test problems: A comparison study. In *Proceedings of the Fourteenth International Eletrotechnical and Computer Science Conference ERK 2005*, pages 121–124, 2005.

[17] T. Rogalsky, R. W. Derksen, and S. Kocabiyik. Differential evolution in aerodynamic optimization. In *Proceedings of the 46th Annual Conference of the Canadian Aeronautics and Space Institute*, pages 29–36, 1999.

[18] R. Salomon. Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions: A survey of some theoretical and practical aspects of genetic algorithms. *Bio. Systems*, 39(3):263–278, 1996.

[19] R. Storn. Differential evolution design of an iir-filter. In *Proceedings of IEEE International Conference on Evolutionary Computation ICEC'96*, pages 268–273, New York, 1996.

[20] A. Toffolo and E. Benini. Genetic diversity as an objective in multi-objective evolutionary algorithms. *Evolutionary Computation*, 11(2):151–167, 2003.

[21] D. V. Veldhuizen. *Multiobjective Evolutionary Algorithms:Classifications, Analyses, and New Innovations*. PhD thesis, Airforce Institute of Technology, 1999.

[22] F. Xue. Multi-objective differential evolution and its application to enterprise planning. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation (ICRA'03)*, volume 3, pages 3535–3541. IEEE Press, 2003.

[23] F. Xue, A. C. Sanderson, and R. J. Graves. Pareto-based multi-objective differential evolution. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, volume 2, pages 862–869. IEEE Press, 2003.

[24] E. Zitzler, L. Thiele, M. M. Laumanns, C. M. Fonseca, and V. G. Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Trans. Evol. Comput.*, 2(2):117–132, 2003.