Fitness Inheritance For Noisy Evolutionary Multi-Objective Optimization

Lam T. Bui
School of ITEE, University of New South Wales@ Australian Defence Force Academy
NorthCott Drive, Canberra ACT, Australia, 2600
l.bui@adfa.edu.au

Hussein A. Abbass
School of ITEE, University of New South Wales@ Australian Defence Force Academy
NorthCott Drive, Canberra ACT, Australia, 2600
h.abbass@adfa.edu.au

Daryl Essam
School of ITEE, University of New South Wales@ Australian Defence Force Academy
NorthCott Drive, Canberra ACT, Australia, 2600
d.essam@adfa.edu.au

ABSTRACT

This paper compares the performance of anti-noise methods, particularly probabilistic and re-sampling methods, using NSGA2. It then proposes a computationally less expensive approach to counteracting noise using re-sampling and fitness inheritance. Six problems with different difficulties are used to test the methods. The results indicate that the probabilistic approach has better convergence to the Pareto optimal front, but it looses diversity quickly. However, methods based on re-sampling are more robust against noise but they are computationally very expensive to use. The proposed fitness inheritance approach is very competitive to re-sampling methods with much lower computational cost.

Categories and Subject Descriptors: B.X.X [Evolutionary Multiobjective Optimization]:

General Terms: Algorithms, Performance.

Keywords: Evolutionary multiobjective optimization, noise, probabilistic model, fitness inheritance.

1. INTRODUCTION

Evolutionary algorithms (EAs), particulary genetic algorithms (GA), are known to be robust in the presence of noise [1, 5]. Population based methods are generally known to be robust in the single objective case against noise since the average performance of the population acts as a filter for noise. However, in the case of evolutionary multi-objective optimization algorithms (EMOs), the aim is to obtain a Pareto set of non-dominated solutions, which makes it harder to filter the noise. So far, comparisons of performance in EMO have been undertaken in the presence of many types of problem difficulties, such as: convexity, non-convexity, or discontinuity. However, not much work has been done in the area of noisy landscapes. In real life black–box optimization problems, the existence of noise during evaluation is inevitable. Sources of noise can vary from noise in the sensors, actuators, or because of the stochasticity pertaining in some problems such as multi–agent simulations.

The research presented in this paper aims to compare a number of approaches to overcome noise during objective evaluation. In particular, we compare two re–sampling techniques and a probabilistic approach proposed by Hughes (2001). We then propose a fitness inheritance technique to reduce the calculation time. NSGA2 is used as a standard EMO algorithm.

The paper is divided into six sections. A review of the EMO literature, noise, and performance metrics is given in section 2. A description of the methods is given in the third section. Section four presents the specifications of the experiments. The results of the experiments are analyzed and discussed in the fifth section then conclusions are drawn in the last section.

2. BACKGROUND

2.1 EMOs

Similar to other optimization algorithms, EMOs are used to find at least one feasible solution for a particular problem [3]. In contrast to single objective optimization, they are associated with conflicting multi–objective functions, defining a multi–dimensional fitness landscape. With EMOs, multiple solutions are usually expected after any iteration. As a result, this is expected to ideally lead to a population of efficient solutions when the termination condition is satisfied. It thus offers decision makers more options from which to choose the best solution according to some preference information.

EMOs have to overcome two major problems [10]. The first problem is how to get as close as possible to the *Pareto optimal front* (POF). Each solution of the POF is a Pareto solution, where no other feasible solution in the search space is better than the former when evaluated on all objective functions. The second problem is how to keep diversity among solutions in the obtained set. These two problems become common criteria for most current comparison measures.

To date, many EMOs have been developed. Generally speaking, they can be classified into two broad categories: non–elitism and elitism. With the elitism approach, EMOs employ an external set to store the best solutions in each generation. This set will then be a part of the next generation. With this method, the best individuals in each

GECCO'05, June 25–29, 2005, Washington, DC, USA.

generation are always preserved, and this helps the algorithm to get as close as possible to the POF. NSGA2 [3] and SPEA2 [10] are examples of this approach. In contrast, the non elitism approach has no concept of elitism when it selects individuals for reproduction [10]. Examples of this approach include VEGA [8] and NSGA [3].

2.2 Noise

When EMOs are applied to real life problems, noise in the evaluation cannot be avoided. When noise exists, it makes the evolving process slow and affects the solution's quality. Generally, noise comes from many different sources, such as: data inputting or sampling [7]. However, this paper focuses on an important form of noise: noise in the objective function evaluation. The way noise influences the fitness value is varied. We use additive noise. This noise can be seen as additional values randomly added to or subtracted from the real fitness value. Since the noisy fitness value is used for selection, it can mislead the algorithm to inferior results; bad solutions might be kept for the next generation, and the good ones might be excluded [1]. Formally, a noisy fitness function takes the following form:

$$F_{noise} = F + noise \tag{1}$$

where F is the noise–free fitness function and *noise* is a source of noise [7]. The source of noise comes in any form such as normal distribution or uniform distribution. In general, the normal distribution is often used to simulate noise [1].

In the context of EMOs, there are a few techniques to deal with noise that have been introduced to date. Re-sampling or re-evaluation of objective values is thoroughly investigated in Miller's PhD thesis [7]. It is a simple, but costly, method because it requires re–evaluating a solution a number of times.

More recently, Hughes's work introduced a probabilistic ranking process [5]. The assumption of the method is that a probabilistic ranking that takes into account the standard deviation in the evaluation of each solution can be used to correct for and avoid any inaccurate judgement because of noise. The probabilistic rank of an individual is the sum of all probabilities of those dominates it. Hughes's experiments assumed that the noise in all individuals come from the same normal distribution, while claiming that the theoretical framework will work regardless of the previous assumption. In addition, he did not estimate the variance for every individual since all individuals are assumed to have noise from the same distribution. For this, Hughes's approach is expected to have an advantage in terms of computational cost. However, in reality we do not know if noise is coming from different normal distributions or from the same distribution. As a result, re–sampling is necessary for estimating the variance of each individual independently. The purpose of this paper is to compare Hughes approach and re–sampling techniques, then propose a method that combines the advantages of both methods.

2.3 Performance metrics

Performance metrics are usually used to compare algorithms in order to form an understanding of which one is better and in what aspects. However, it is hard to define a concise definition of algorithmic performance. In general, when doing comparisons, a number of criteria are employed [10]. We will look at two of these criteria. The first measure is the generation distance, *GD*, which is the average distance from the set of solutions found by evolution to the POF [9]:

$$GD = \frac{\sqrt{\sum_{i=1}^{N} d_i^2}}{N} \tag{2}$$

where d_i is the Euclidean distance (in objective space) from solution i to the nearest solution in the POF. If there is a large fluctuation in the distance values, it is also necessary to calculate the variance of the metric. Finally, the objective values should be normalized before calculating the distance.

The second metric presented in this section uses the statistical comparison method. It was first introduced by Fonesca and Fleming [4]. For EMO experiments, which generate a large set of solutions, this metric is often the most suitable, as their data can easily be assessed by statistical methods. Knowles and Corne [6] modified this metric and instead of drawing parallel lines, all lines originate from the origin. The basic idea is as follows: suppose that two algorithms (*A1, A2*) result in two non-dominated sets: *P1* and *P2* (as in Figure 1). The lines that join the solutions in P1 and P2 are called attainment surfaces. The comparison is carried out in the objective space. In order to do the comparison, a number of lines are drawn from the origin (assuming minimization problem), such that they intersect with the surfaces. The comparison is then individually done for each sampling line to determine which one outperforms the other. Each intersection line will then yield to a number of intersection points. In this case, statistical tests are necessary to determine the percentage an algorithm outperforming the other in each section. For both of these methods, the final result is two numbers that show the percentage of the space where each algorithm outperforms the other.

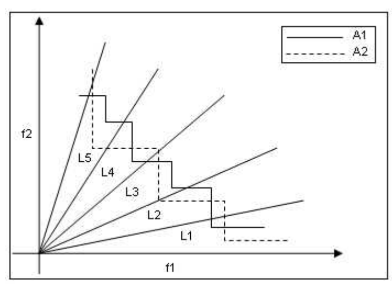

Figure 1: Sampling the non-dominated sets using lines of intersection

3. METHODS

3.1 Original NSGA2

NSGA2 is an elitism algorithm [3]. The main feature of NSGA2 lies in elitism-preservation operation. Firstly, the archive size is set equal to the initial population size. The current archive is then determined based on the combination of the current population and the previous archive. To do this, NSGA2 uses dominance ranking to classify the population into a number of layers, such that the first layer is the best layer in the population. The archive is created based on the order of ranking layers: the best rank being selected first. If the number of individuals in the archive is smaller than the population size, the next layer will be taken into account and so forth. If adding a layer makes the number of individuals in the archive exceeds the initial population size, a truncation operator is applied to that layer based on crowding distance.

The crowding distance of a solution x is calculated as follows: the population is sorted according to each objective to find adjacent solutions to x; boundary solutions are assigned infinite values; the average of the differences between the adjacent solutions in each objective is calculated; the truncation operator removes the individual with the smallest crowding distance.

$$D(x) = \sum_{m=1}^{M} \frac{F_m^{I_x^m+1} - F_m^{I_x^m-1}}{F_m^{max} - F_m^{min}} \tag{3}$$

in which F is the vector of objective values. I_x^m returns the sorted index of the solution x, according to objective m-th.

An offspring population of the same size as the initial population is then created from the archive by using crowded tournament selection, crossover, and mutation operators. Crowded tournament selection is a traditional tournament selection method, but when two solutions have the same rank, it uses the crowding distance to break the tie.

3.2 Probabilistic-based NSGA2

We use Hughes's probabilistic model [5] and adopt it to NSGA2. In the presence of noise, the objective value of an individual A may be smaller than an individual B while the A's noise–free value is greater. As a result, if a decision is made that A is better than B (assuming minimization), inferior solutions will be selected more often. Therefore, in a noisy environment, a probabilistic decision is more adequate. The probability in which A is better than B is estimated based on an estimate of uncertainty in the values assigned to A and B. This estimate uses the variance of the expected noise in both solutions.

Hughes proposed a probabilistic ranking model. It gives an individual a rank that is the sum of probabilities that each individual in the population dominates the individual of interest. This probability is interpreted as probability of the wrong decision made on two individuals (Equation 4). So, the smaller the rank, the better the individual.

$$P(A > B) \approx \frac{1}{1 + e^{-\frac{2.5m}{\sqrt{2+2s^2}}}} \tag{4}$$

with $m = \frac{\mu_a - \mu_b}{\sigma_b}$ $s = \frac{\sigma_a}{\sigma_b}$ in which μ and σ are the mean and standard deviation of the fitness values of A and B, respectively.

In this paper, we generated a probabilistic version of NSGA2, called $NSGA2 - P$, using Hughes' probabilistic framework in order to deal with noise. NSGA2's ranking and crowding distance are replaced with probabilistic ranking and crowding distance, respectively. The original NSGA2 uses objective values to rank the individuals, while probabilistic–based NSGA2 uses the probability that an individual dominates another. In the multi–objective optimization context, the concept of dominance needs to be generalized. Formally, suppose that there are K objectives, three types of probabilities are calculated as follows

$$P(A > B) = \prod_{k=1}^{K} P(A_k > B_k)$$

$$P(A < B) = \prod_{k=1}^{K} (1 - P(A_k > B_k))$$

$$P(A \equiv B) = 1 - P(A < B) - (A > B) \tag{5}$$

where, $P(A_k > B_k)$ is the probability that A is better than B in objective k and $P(A \equiv B)$ is the probability that A and B are non–dominated.

Now, the rank of an individual i is calculated as follows:

$$R_i = \sum_{j=1}^{N} P(Ind_j > Ind_i) + \frac{1}{2} \sum_{j=1}^{N} P(Ind_j \equiv Ind_i) - 0.5 \tag{6}$$

where $P(Ind_j > Ind_i)$ is the probability that individual j is better than i.

For the sake of simplicity of implementation, the double-valued ranks are converted equivalently to integral ranks. In NSGA2, solutions with the same rank are grouped and sorted into layers. All solutions within a layer are selected and added to the archive. When the number of solutions in a layer exceeds the number of solutions required to fill the archive, a truncation process is undertaken.

We adapt both the selection and truncation processes to the probabilistic method as well. In the selection process, if the rank of a solution A is better than B's, A is selected. If they have the same rank, a probabilistic tournament selection is undertaken as follows: A random number R is generated. If $R < P(A_D > B_D)$, A is selected and vice versa. $P(A_D > B_D)$ is calculated as in Equation 4 in which $m = \frac{D(A) - D(B)}{\sigma_b}$.

During the truncation process, the crowding distance, D, is calculated for each individual. In the probabilistic version, Equation 3 is replaced with Equation 7.

$$R_x = \sum_{y=1}^{N} P(y_D > x_D) - 0.5 \tag{7}$$

N is the population size, and $P(y_D > x_D)$ is the probability that an individual y has a better crowding distance than individual x. An individual with the largest value of R will be truncated first until the maximum archive size is maintained.

3.3 Resampling–based NSGA2

In this paper, an objective value is evaluated a number of times to reduce the effect of noise. Suppose that, the objective value is evaluated N times, the re-sampled value is calculated as follows:

$$F = \frac{\sum_{k=1}^{N} (f_k)}{N} \tag{8}$$

With N times of evaluation, we can estimate the standard deviation as follows:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N} x_i^2 - \frac{({}_{i=1}^{N} x_i)^2}{N}}{N-1}} \tag{9}$$

The performance of the re-sampling technique will be tested on the base of NSGA2 with two different versions: One with fitness value F called $NSGA2-R$ and the other with $\frac{F}{\sigma}$ called $NSGA2-RS$. The former estimates the mean, while the latter takes into account the variance as a measure of stability and confidence in the mean.

3.4 Fitness inheritance in NSGA2

Since all of the previous three algorithms works by re-evaluating each individual a number of times, it is usually a computationally expensive task. Fitness inheritance works by assigning a child a weighted sum of the parents' fitness values [2]. However, one of the main challenges in fitness inheritance is to decide when and when not to inherit. We propose a variation of the re-sampling technique based on fitness inheritance.

In the proposed algorithm, the child inherits the mean fitness of the two parents. The child is then evaluated only once to generate a single objective value for each noisy objective function. If this value falls within a confidence interval based on the inherited fitness, the inherited fitness is accepted and the algorithm continues; otherwise, the child is re-evaluated a number of time and a mean and a standard deviation are estimated. Formally, suppose that two parents A and B are selected to generate a child C, the proposed fitness inheritance process takes place as follows:

1. Define $\overline{\mu} = \frac{\mu_1+\mu_2}{2}$ and $\overline{\sigma} = \frac{\sigma_1+\sigma_2}{2}$
2. Evaluate C's objective value f
3. If $(\overline{\mu} - 3 * \overline{\sigma} \leq f \leq \overline{\mu} + 3 * \overline{\sigma})$

Assign $\overline{\mu}$ as C's objective value and $\overline{\sigma}$ as C's standard deviation.

4. Else

Evaluate C for another nine times to estimate the objective values and the standard deviation.

The re-sampling and probabilistic versions of the fitness inheritance technique are referred to as NSGA2-RH and NSGA2-PH respectively.

4. EXPERIMENTAL SETUP

We have compared the algorithms with all six Zitler's problems [10]. Each case is tested with thirty different random seeds. The POF of these problems takes different forms such as convex, non-convex and discontinuous. All the problems have two objectives, $f1(x)$ and $f2(x)$, that must be minimized. For each problem, $f_2(x) = g(x) * h(f_1(x), g(x))$, while functions f_1, g and h are uniquely defined for each.

We assume that the noise associated with each individual is coming from Normal distributions with the same mean, but with different variances. However, all variances are sampled from a hyper-Normal distribution with fixed parameter values. Our conjecture is that this setup is a challenge for those methods that work only when the source of noise in all individuals is the same. Also, this setup is more suitable for many real life applications such as applications in control or signal processing, where the variance in the noise can change under different experimental setups.

We use a Gaussian distribution $N(0,\sigma)$ with zero mean to simulate noise. σ is sampled from a hyper–Gaussian distribution with 0.12 mean and 0.025 variance. When calculating the objective values for each individual in the population, noise will be added as follows:

$$\mathrm{F}_{real} = \mathrm{F} + N(0,\sigma), \quad \sigma = N(0.12,0.025)$$

For the statistical comparison, we use 500 lines to divide the objective space. We introduce the concept of $NETgain$ instead. Each algorithm is compared against the original NSGA2 ran without noise in the fitness evaluations. At each generation, the $NETgain$ of an algorithm is determined as follows:

$$NETgain = 100 - (X - Y)$$

where X is the percentage that the original NSGA2 ran without noise outperforms the corresponding algorithm ran with noise, while Y is the percentage that the algorithm ran with noise outperforms the original NSGA2 ran without noise. The $NETgain$ shows how good the performance of an algorithm is in comparison to the original NSGA2 in a noiseless environment. This measure would have a maximum value of 100 when $X = Y$; that is, when the algorithm running with noisy fitness is either performing equivalent to the original NSGA2 ran without noise (*i.e.* $X = Y = 0$) or that the two algorithms outperformed each others equally. Ideally, we want this measure to be as close as possible to 100. However, practically, this is not possible unless we are able to filter the noise in the evaluation.

We also use the number of individuals in the non-dominated set and the generation distance as supportive metrics. The population size is set to 100, the number of generations to 1000, and each individual is evaluated 10 different times.

5. RESULTS AND DISCUSSION

To understand the dynamics of each method, we calculated the comparison metrics in each generation. The NET-gains over time for each algorithm including NSGA2-P, NSGA2-R, NSGA2-RS, NSGA2-RH, NSGA2-PH and the original NSGA2 in each problem are generated.

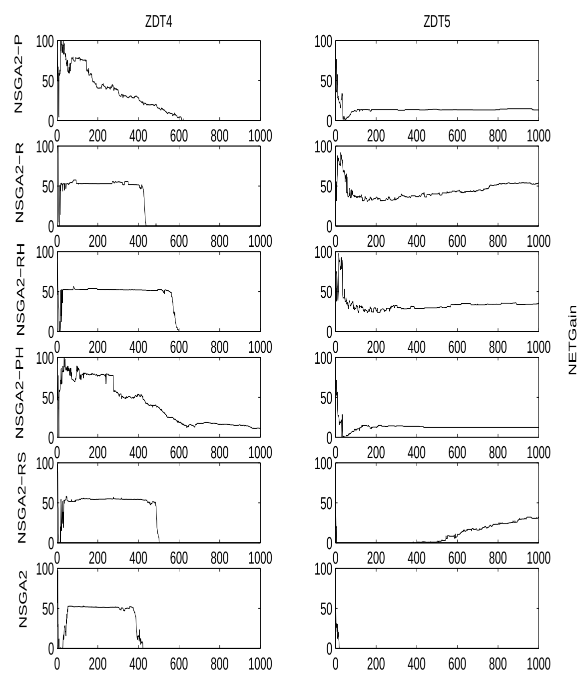

Figure 2: The NET–gains of the five derived algorithms for ZDT4, ZDT5. The x-axis represents the number of generation and the y-axis represents the NET–gain by each algorithm.

The results show that NSGA2-P takes over in the first 100 generations for all problems except in the case of ZDT4 and ZDT5 (Figure 2). In later generations, the NETgain for all algorithms is zero, which means that all algorithms are overtaken by NSGA2 using noise–free evaluations. This is perfectly expected since NSGA2 without noise should be the ceil for the performance. What is important in these figures is that NSGA2-P seems to perform worse because of its initial fast convergence. Obviously, NSGA2-R makes a considerable improvement over the original NSGA2 in the presence of noise.

For ZDT4, algorithms are constrained by multi layers of local optima. In early generations, NSGA2-P is still the winner. However, the resampling-based approaches shows a better capacity of filtering noise over time. For ZDT5, all algorithms converge to a deceptive local optimal front. It is clear that the resampling once more is better.

In all problems, the fitness inheritance approach is consistent with its variant; that is, the fitness inheritance with probabilistic model is as bad as the probabilistic model while the fitness inheritance with resampling is as good as the resampling method (see Fig. 2).

We take a look further at generation distances (See Fig. 3-4). Once more, NSGA2-P is slightly better than the others in early generations. However, NSGA2-R and NSGA2-RS are better than NSGA2-P on ZDT1-ZDT3 and ZDT6, and competitive on ZDT4 and ZDT5 as time progresses. It seems that ZDT4 and ZDT5 with deceptive and multi-modal difficulties cause problems for the resampling-based methods, while the mistakes occurring in NSGA2-P helps it to escape the problem difficulties. This is normal since small noise can help an algorithm to escape local optima. Still, over time the generation distance for NSGA2-P gets slightly smaller.

The improvement in generation distance for NSGA2-P and the deterioration of the statistical measure may seem contradictory. However, on a closer look at the visualization of the Pareto front, it becomes evidenced that the deterioration of the statistical measure for NSGA2-P is because of loss of diversity in its non-dominated set (Figures 5–6).

The findings are clearer when looking further to all snapshots of the Pareto set found by evolution over time. NSGA2-P is inferior in all problems. Its non-dominated sets heavily loose diversity. This lack of diversity makes it hard for NSGA2-P to search for suitable solutions in order to converge to the POF. One possible reason is the possible loss of extreme solutions with NSGA2-P when it builds the archive while resampling-based methods do not to some extent. We also found out that NSGA2-P's exploration capability is substantially limited when we looked at the convergence of the non-dominated set over time. The algorithm was not able to recover from *genetic drift*.

We also calculated the number of non-dominated solutions, but could not visualize it for the space limitation. Except for ZDT5, the number of non-dominated solutions of NSGA2-P is many times smaller than NSGA2-R and NSGA2-RS. For ZDT5, the number of non-dominated set of NSGA2-P is quite high but with very small spread, which implies that NSGA2-P suffers substantially from *genetic-drift*.

To discover the underlying reason for the inferior performance of NSGA2-P, we undertook further experiments in which the algorithm is tested with two alternatives: using NSGA2-R niching plus probabilistic selection, and probabilistic niching plus the NSGA2-R selection. We could not find any improvement in terms of performance.

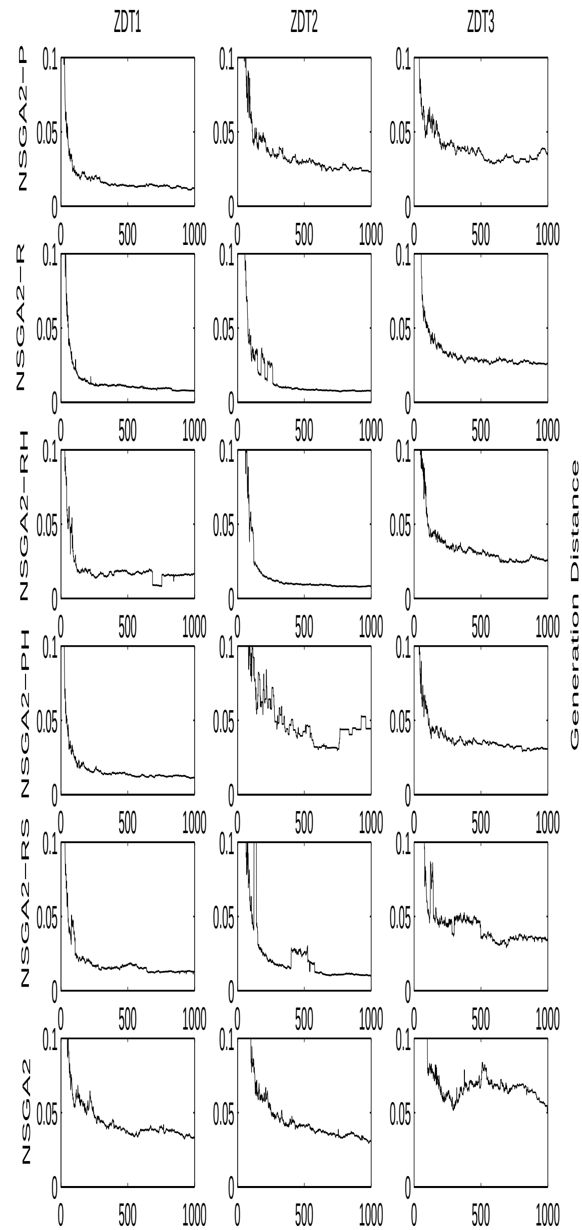

Figure 3: Five derived algorithms with the generation distance values of the non-dominated sets for ZDT1, ZDT2 and ZDT3.

These results came surprising because they imply that the employed niching mechanisms have no effect on NSGA2-P's performance. This explains the loss of diversity. The slight difference is just because of selection techniques. When we verified this, we found out that the niching module was not called. The individuals were taken one by one to the archive until it is full. These results also pose a question about the correctness of integrating probabilistic model with population-based model in which probabilistic model as-

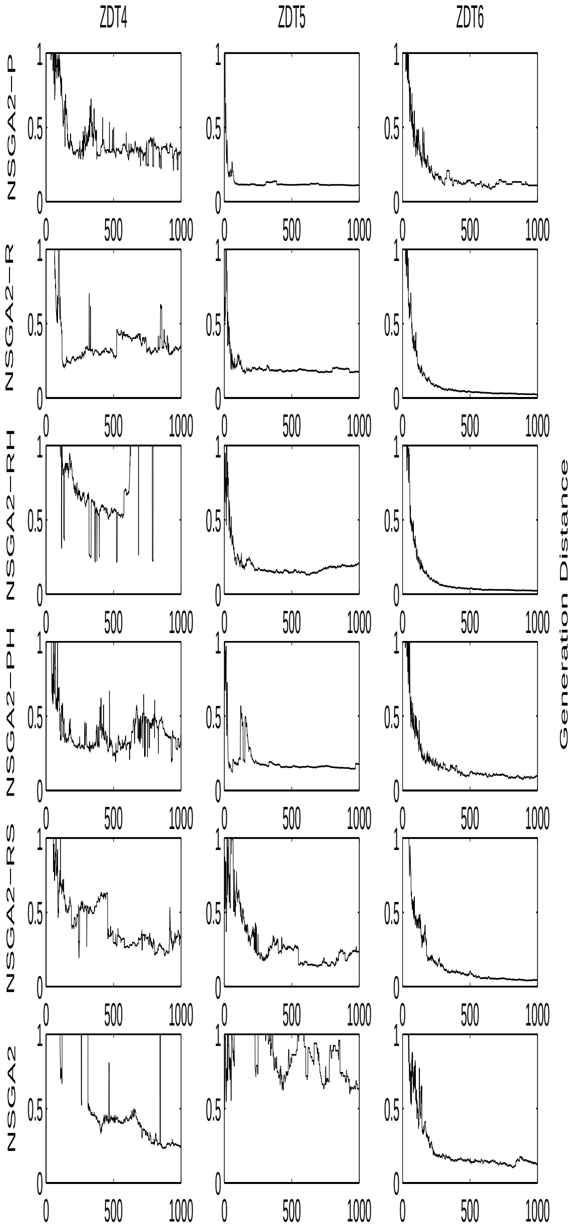

Figure 4: Five derived algorithms with the generation distance values of the non-dominated sets for ZDT4, ZDT5 and ZDT6.

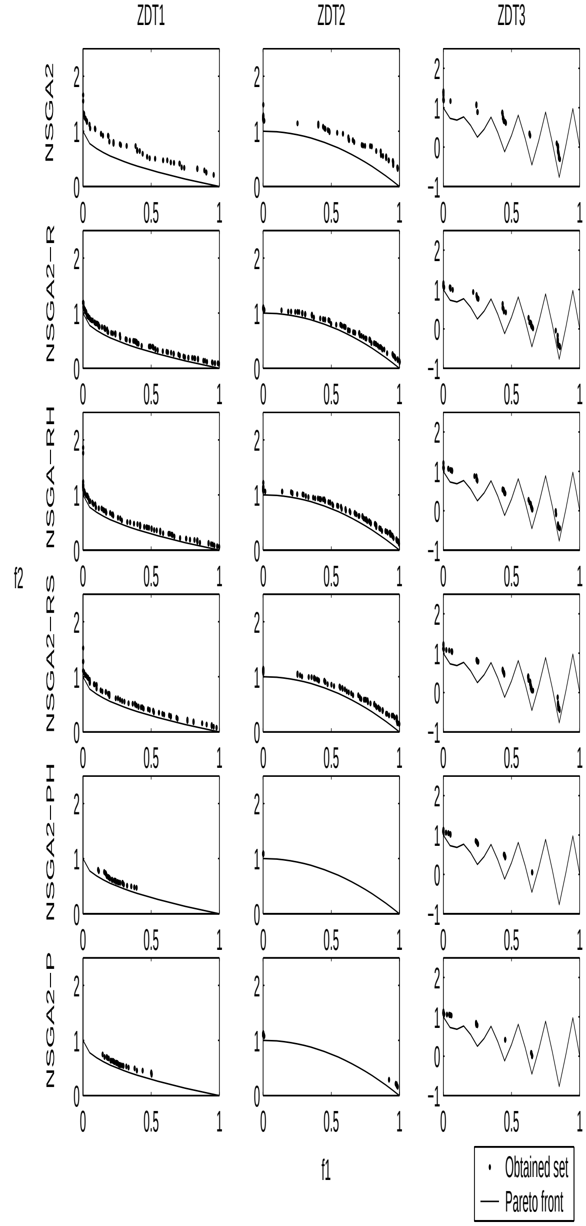

Figure 5: A snapshot of non-dominated sets of the original NSGA2 and five derived algorithms on ZDT1, ZDT2, and ZDT3.

sumes that the solutions represent an independent sample, meanwhile solutions in a population are somewhat correlated as a result of the evolutionary operators.

Lastly, we looked at the amount of savings resultant from using the fitness inheritance approach. Table 1 lists the percentage of computations saved when using fitness inheritance. It is clear that there is a substantial amount of savings without deteriorating the performance.

Table 1: Evaluation times.

Test Function	Savings by $NSGA2-RH$
ZDT1	30%
ZDT2	29%
ZDT3	24%
ZDT4	17%
ZDT5	33%
ZDT6	29%

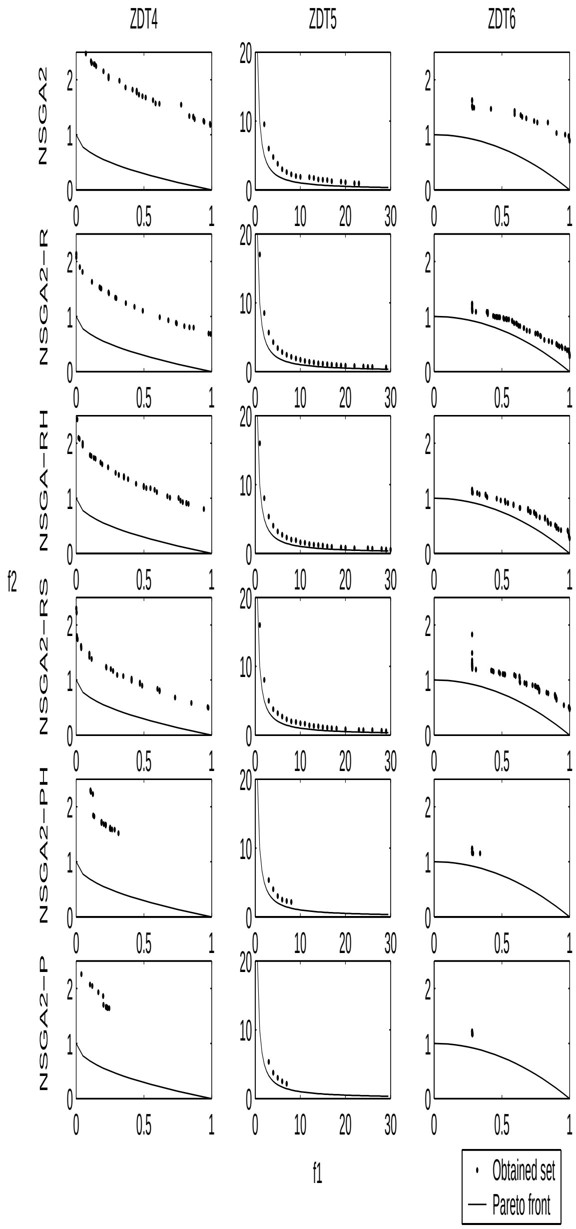

Figure 6: A snapshot of non-dominated sets of The original NSGA2 and five derived algorithms on ZDT4, ZDT5, and ZDT6.

6. CONCLUSION

In this paper, we scrutinized the performance of anti-noise methods on the six ZDT problems. Re-sampling and probabilistic methods are compared in the context of NSGA2. The noisy environment is established by adding a random amount of noise to each individual. The variance of each noise level was generated from a hyper-normal distribution. The results show that the diversity of the probabilistic approach is inferior in comparison to resampling.

In order to reduce the computational cost of these algorithms, we also used fitness inheritance. The performance of different methods was maintained while a substantial amount of reduction in the computational cost was achieved.

For future work, we believe that the probabilistic approach still has merit if we improve its selection procedure.

7. ACKNOWLEDGMENTS

This work is supported by the University of New South Wales grant PS04411 and the Australian Research Council (ARC) Centre on Complex Systems grant number CEO0348249. The authors would like to thank the anonymous reviewers for their useful comments.

8. REFERENCES

[1] D. Bche, P. Stoll, R. Dornberger, and P. Koumoutsakos. Preprint: Multi-objective evolutionary algorithm for the optimization of noisy combustion processes. *IEEE TRansaction on Systems, Man, and cybernetic*, 32(4), 2002.

[2] J. H. Chen, D. Goldberg, K. Satry, and S. H. Ho. Fitness inheritcance in multi-objective optimization. Technical Report 2002017, IlliGAL, University of Illinoise at Urbana-Champaign, 2002.

[3] K. Deb. *Multiobjective Optimization using Evolutionary Algorithms*. John Wiley and Son Ltd, New York, USA, 2001.

[4] C.M. Fonseca and P.J. Fleming. On the performance assessement and comparision of stochastic multiobjective optimizers. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature - PPSN IV, Lecture Notes in Computer Science*, pages 584–593. Springer Verlag, Berlin Germany, 1996.

[5] E. J. Hughes. Evolutionary multi-objective ranking with uncertainty and noise. In Zitzler et al., editor, *Proceedings of the First Conference on Evolutionary Multi-Criterion Optimization*, Zurich, Switzeland, 2001.

[6] J. Knowles and D. Corne. Approximating the nondominated front using the pareto archibed evoltion strategy. *Evolutionary Computation*, 8(2):149–172, 2000.

[7] B.L. Miller. *Noise, Sampling, and Efficient Genetic Algorithms, PhD thesis, Department of ComputerScience*. PhD thesis, Department of ComputerScience, Univeristy of Illinoise at Urbana-Champaign, 1997.

[8] J.D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *Genetic Algorithms and their Applications:Proceedings of the First International Conference on Genettic Algorithms*, pages 93–100, Hillsdale, New Jersey, 1985.

[9] D. A. V. Veldhuizen and G. B. Lamont. Multiobjective evolutionary algorithm test suites. In J. Carroll, H. Haddad, D. Oppenheim, B. Bryant, and G.B. Lamont, editors, *ACM Symposium on Applied Computing*. ACM, 1999.

[10] E. Zitzler, L. Thiele, and K. Deb. Comparision of multiobjective evolutionary algorithms: Emprical results. *Evolutionary Computation*, 8(1):173–195, 2000.