Evolutionary Multi-Objective Ranking with Uncertainty and Noise

Evan J. Hughes

Department of Aerospace, Power, and Sensors,
Cranfield University, Royal Military College of Science,
Shrivenham, Swindon, SN6 8LA, UK,
ejhughes@iee.org,
http://www.rmcs.cranfield.ac.uk/departments/daps/

Abstract. Real engineering optimisation problems are often subject to parameters whose values are uncertain or have noisy objective functions. Techniques such as adding small amounts of noise in order to identify robust solutions are also used. The process used in evolutionary algorithms to decide which solutions are better than others do not account for these uncertainties and rely on the inherent robustness of the evolutionary approach in order to find solutions.

In this paper, the ranking process needed to provide probabilities of selection is re-formulated to begin to account for the uncertainties and noise present in the system being optimised. Both single and multi-objective systems are considered for rank-based evolutionary algorithms.

The technique is shown to be effective in reducing the disturbances to the evolutionary algorithm caused by noise in the objective function, and provides a simple mathematical basis for describing the ranking and selection process of multi-objective and uncertain data.

1 Introduction

The use of evolutionary algorithms (EA's) in engineering is now well established and widespread. As the use of the algorithms migrates deeper into industry, and with more processing power available, the scale and characteristics of the problems being solved are changing. The objective functions are becoming more complex, nonlinear and often uncertain. Many model coefficients are derived by experiment and are therefore subject to experimental errors. In real systems, the true coefficients will not be the same as measured and are often time dependent or correlated with platform motion etc.

These errors in the modelling are unavoidable and inevitably propagate into the outputs of the objective functions, the results of which are used to classify the quality of the individual solutions to the problem. All optimisation algorithms attempt to find the problem solution that gives the most favourable output from the objective functions. With complex systems, evolutionary algorithms are a useful tool in that they can tolerate highly nonlinear and noisy system models and objective functions and still provide reasonable suggested solutions [1].

This robustness to errors has also been exploited by artificially adding noise to the objectives in an attempt to identify solutions that are robust to noise and uncertainty in the real system [2–4]. Noise is also often present when trying to optimise hardware systems such as in robotics. Noise or uncertainty in the objectives tend to slow evolution and reduce solution quality.

Attempts to reduce noise by repeating objective calculations and then averaging or combining results have been tried [5], but often with many realistic problems, the time to re-evaluate is prohibitive and often the number of samples used to average must be very small and therefore subject to considerable error. Most evolutionary algorithms to date have accepted these problems as the robustness of the algorithms allows small errors to be tolerated.

Therefore we may form two categories of problem:

1. **Noisy:** Two successive evaluations of the same chromosome information return two different sets of objectives.
2. **Uncertain:** Two successive evaluations of the same chromosome return the same objective values, but when comparing two different chromosomes, errors in the modelling and model coefficients may cause the objective values returned to classify the wrong solution as being superior.

This paper takes a fresh look at the problems of uncertain and noisy systems, both with single and multiple objectives, in order to provide a selection process that is aware of the uncertainties and noise. The techniques discussed form a small step towards creating algorithms that can address the problems associated with the different categories of noisy or uncertain problems.

2 Problem Definition

As most engineering problems have multiple objectives that must be satisfied, the work concentrates on multi-objective evolutionary algorithms (MOEA). Carlos Coello Coello maintains an excellent database of publications relating to multi-objective optimisation [6]. Many of the publications tackling engineering problems (e.g. [7]) use techniques such as MOGA [8] and NSGA [9]. These methods use ranking techniques to address the problems of non-domination, then use sharing to spread the solutions across the objective surface. The use of ranking is widespread in EA's to prevent good solutions taking over the population in the early generations of the algorithm. Van Veldhuizen and Lamont [10] have studied the benefits / disadvantages of a number of techniques, including MOGA and NSGA, and begun to define techniques for assessing MOEA performance. These have been developed in the context of noise-free and certain problems and similar work is needed to address noisy and uncertain problems but is beyond the scope of this paper.

In all evolutionary algorithms, the key medium to evolution is being able to take two potential solutions to a problem, test them in the problem domain against some performance criteria, then given some values relating to the performance of each, decide which solution is better than the other. With noisy or uncertain problems, we find that given the results of the performance criteria, unless they are very different, we cannot say for certain which solution is better. Thus we must now refer to the probability that one solution is better than the other. This paper aims to review the process needed in order rank a set of objective results, given that we can no longer make a crisp decision about solution superiority.

3 Comparing Two Uncertain Fitness Measurements

3.1 Introduction

In a noise free situation, if we have two fitness values, A and B, and are trying to minimise, the lower value is always superior. However, if we know the fitness values are subject to noise, even if the measured fitness A is less than the measured fitness B, the expected value of the distribution from which A is drawn may be greater than the expected value of the distribution from which B is drawn. Therefore we would make the wrong decision. In the presence of noise, if we choose the simple case of *take the best measured objective*, we need to quantify the probability that we have made the wrong decision.

3.2 Analysis of Distributions With Unknown Expected Values

If we have a pair of samples from distributions with known characteristics and spread, but unknown expected values, we need to be able to calculate the probability that although sample A is less than sample B say, the expected value of distribution B is less than the expected value of distribution A. This will give us a probability of making the wrong decision. Figure 1 shows a scenario with two Gaussian distributions.

Here A and B are the measurements that were observed, and μ is an arbitrary point. The observed value A was less than B and is therefore superior. If the expected value of A was actually at point μ, the expected value of B would have to be in any position to the left of μ for us to make the wrong decision. We can calculate the probability of the distributions being in this location as the probability of value A occurring, given μ_a, multiplied by the probability that μ_b is less than μ_a, shown as the shaded region on Fig. 1. This may be described mathematically as

$$P(\text{wrong decision}) = \int_{-\infty}^{\infty} \text{pdf}_A(A-\mu) \cdot \text{cdf}_B((X-\mu) > (B-\mu))d\mu \ . \quad (1)$$

With Gaussian distributions, we may write this as

$$P(\text{wrong}) = \int_{-\infty}^{\infty} \left(\frac{1}{\sigma_a\sqrt{2\pi}} e^{-\frac{(A-\mu)^2}{2\sigma_a^2}} \int_{(B-\mu)}^{\infty} \frac{1}{\sigma_b\sqrt{2\pi}} e^{-\frac{y^2}{2\sigma_b^2}} dy \right) dx \ . \quad (2)$$

Equation 3 has the limits on the inner integration adjusted, as the Gaussian distribution is symmetrical $\text{pdf}(a) = \text{pdf}(-a)$ and $\text{cdf}(a, \infty) = \text{cdf}(-\infty, -a)$. The

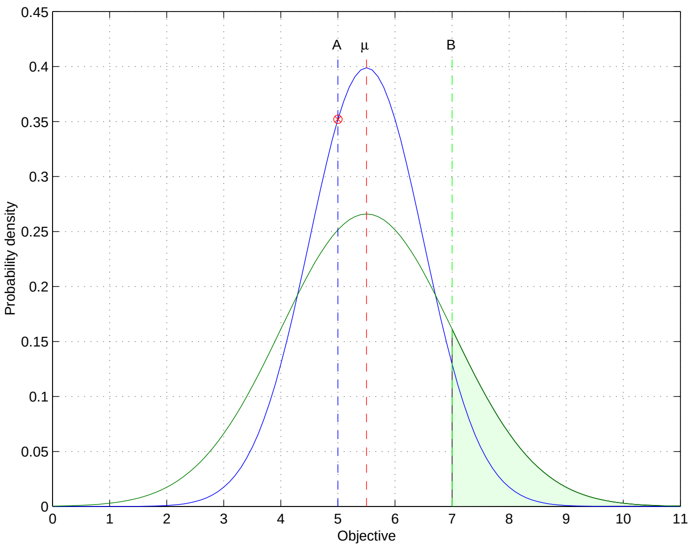

Fig. 1. Choice between two noisy values

axis is shifted to make sample point $B = 0$ then distribution B is normalised, modifying distribution A accordingly. Equation 4 has been simplified with the replacements $m = \frac{(A-B)}{\sigma_b}$ and $s = \sigma_a/\sigma_b$.

$$P(\text{wrong}) = \int_{-\infty}^{\infty} \left(\frac{\sigma_b}{\sigma_a\sqrt{2\pi}} \, e^{-\frac{(\mu - \frac{A-B}{\sigma_b})^2 \sigma_b^2}{2\sigma_a^2}} \int_{-\infty}^{\mu} \frac{1}{\sqrt{2\pi}} \, e^{-\frac{y^2}{2}} dy \right) d\mu \tag{3}$$

$$P(\text{wrong}) = \int_{-\infty}^{\infty} \left(\frac{1}{s\sqrt{2\pi}} \, e^{-\frac{(\mu - m)^2}{2s^2}} \int_{-\infty}^{\mu} \frac{1}{\sqrt{2\pi}} \, e^{-\frac{y^2}{2}} dy \right) d\mu \tag{4}$$

We may now use the error function

$$\text{erf}\,(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{t^2} dt \tag{5}$$

to give

$$\begin{aligned} P(\text{wrong}) &= \int_{-\infty}^{\infty} \left(\frac{1}{s\sqrt{2\pi}} \, e^{-\frac{(x-m)^2}{2s^2}} \, \frac{1 + \text{erf}\,(\frac{x}{\sqrt{2}})}{2} \right) dx \\ &= \frac{1}{2} + \frac{1}{2s\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-\frac{(x-m)^2}{2s^2}} \, \text{erf}\,(\frac{x}{\sqrt{2}}) \, dx \ \ . \end{aligned} \tag{6}$$

Unfortunately (6) is difficult to integrate directly. An alternative approach is to recognise that the difference between two Gaussian distributions is also

Gaussian but with an expected value that is the difference between the expected values of the two distributions and a variance which is a sum of the two variances (Cramer's Theorem), i.e.,

$$N(\mu_a, \sigma_a^2) - N(\mu_b, \sigma_b^2) = N(\mu_a - \mu_b, \sigma_a^2 + \sigma_b^2) \tag{7}$$

If A dominates B in a minimisation sense, then the area under the resulting curve from zero to infinity will give the probability that the decision that A dominates B is wrong. If we normalise B to give

$$\begin{aligned} N(\frac{\mu_a - \mu_b}{\sigma_b}, \frac{\sigma_a^2}{\sigma_b^2}) - N(0,1) &= N(\frac{\mu_a - \mu_b}{\sigma_b}, \frac{\sigma_a^2}{\sigma_b^2} + 1) && (8) \\ &= N(m, s^2 + 1) && (9) \end{aligned}$$

then the probability of being wrong is

$$\begin{aligned} P(\text{wrong}) &= \frac{1}{\sqrt{2\pi\,(s^2+1)}} \int_0^\infty e^{-\frac{(x-m)^2}{2(s^2+1)}} dx && (10) \\ &= \frac{1}{2} + \frac{1}{2} \operatorname{erf}\left(\frac{m}{\sqrt{(2+2s^2)}} \right) \quad . && (11) \end{aligned}$$

Equation 11 can be shown to be equal to (6) by numerical integration.

3.3 Numerical Approximation of Probability

If we cannot afford to do multiple evaluations for each chromosome (often the case), we can choose a random chromosome before running the EA and perform multiple evaluations to estimate the noise standard deviation (and possibly noise distribution). This estimate may be used subsequently for all the comparisons of individual samples using the equations in section 3.2. If the noise statistics are known to be nonlinear, it may be advantageous to either re-estimate the statistics every few generations from an average chromosome, or even from the current population. When the same standard deviation is used for comparing two objective values, $\sigma_a = \sigma_b$ therefore $s = 1$. Thus the probability is only determined by the value of m.

As the case of $s = 1$ is likely to be the most commonly used, we can tailor the equations specifically. The equations are calculated as the probability of being wrong in minimisation, this is the same as the probability of acceptance in maximisation. Thus the probability of sample A dominating sample B in maximisation ($P(A > B)$) is

$$\begin{aligned} P(A > B) &= \frac{1}{2} + \frac{1}{2} \operatorname{erf}(\frac{m}{2}) \\ &= \frac{1}{2} + \frac{1}{2} \operatorname{erf}(\frac{A - B}{2\sigma}) \ , && (12) \end{aligned}$$

therefore if $A = 0$, $B = 5$ and $\sigma = 1$, $P(A > B) = 0$ as expected.

Unfortunately, the error function erf (x) is not easy to calculate quickly. It can be approximated using Chebyshev fitting [11, Section 6.2] but even this is not very quick. Recognising that (11) is sigmoidal in shape, other standard sigmoidal curves have been fitted to give a good approximation to the curve, but allow the probability to be calculated quickly. Figure 2 shows the curve approximation and (13) & (14) show the equations. The results of the two different approximations are so similar to each other, they appear as a single line on the graph.

$$P(A > B) \approx \frac{1}{2}(1 + \tanh \frac{m}{0.8\sqrt{2 + 2s^2}}) \tag{13}$$

$$P(A > B) \approx \frac{1}{1 + e^{-\frac{2.5\,m}{\sqrt{2+2s^2}}}} \tag{14}$$

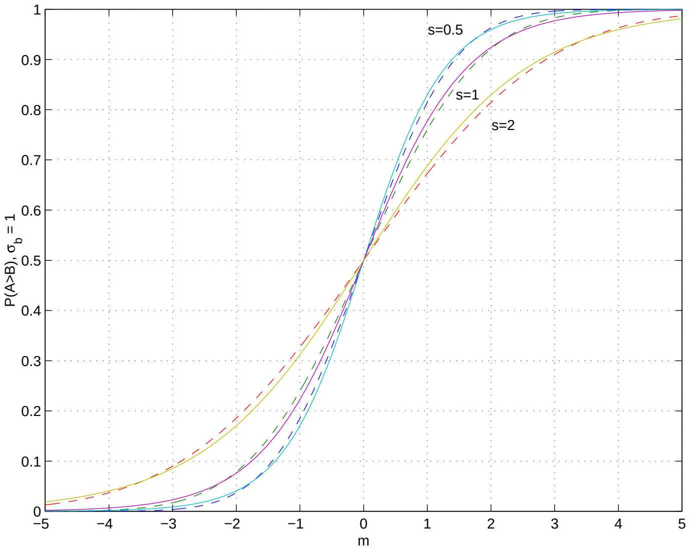

Fig. 2. Approximation of $P(A > B)$ against m, dashed curve is (11)

The small errors in the approximation can be tolerated as a tradeoff for the speed gain. Further improvements in calculation speed may be obtained for certain problems by utilising the decomposition of $\tanh(A + B)$. This is detailed in [12].

3.4 Multi-Objective Fitness Functions

With multiple objectives, we no longer have only two possible outcomes from comparing two objectives A and B. We now have the possibility of the two objectives being non-dominated. We therefore can have $P(A < B)$, $P(A > B)$, and $P(A \equiv B)$ (A is equivalent to B in Pareto optimal sense). Figure 3 shows the effect graphically, with the point A in the centre of the figure ($[0.5, 0.5]$) representing one sample of the fitness. The shaded regions correspond to regions in which there is information to drive the evolutionary process, i.e. A is either dominates or is dominated by any point x that lies in the shaded region. In the remaining regions, any point x will be non-dominated when compared to point A and we have no way of deciding between the points. For the ranking process, the points are equivalent, and just as good as each other.

If we have two, k objective, independent fitness measurements with corresponding objective values $A_1 \ldots A_k$, and $B_1 \ldots B_k$, the probabilities $P(A < B)$, $P(A > B)$, and $P(A \equiv B)$ are simply

$$P(A < B) = \prod_{j=1}^{k} P(A_j < B_j) \tag{15}$$

$$P(A > B) = \prod_{j=1}^{k} P(A_j > B_j) \tag{16}$$

$$= \prod_{j=1}^{k} (1 - P(A_j < B_j)) \tag{17}$$

$$P(A \equiv B) = 1 - P(A < B) - P(A > B) \ . \tag{18}$$

4 Probabilistic Ranking and Selection

4.1 Introduction

Ranking is often employed to prevent a superior solution dominating the early populations in the evolutionary process. The conventional ranking process, however, does not take the uncertainty in the measured fitness values into account. The following sections provide a fresh view of the ranking process and develop theory for multi-objective ranking of uncertain fitness measurements.

4.2 Single Objective Ranking

Figure 4 shows seven fitness values to be ranked. If we are minimising, the best fitness value is the lowest. In the case shown, value A will get rank 0, and value G will be rank 6. Values C and D are equal and therefore should be assigned the same rank. We can assign rank values as shown in Table 1

If we did not have a tie between C & D, we could use linear selection (19) to calculate probabilities of selection, based on the ranked fitness, where n is the number of fitness values and R_i is the rank of individual i. The sum of the rank values on the denominator will sum to $n(n-1)/2 = 21$ which is the sum of the

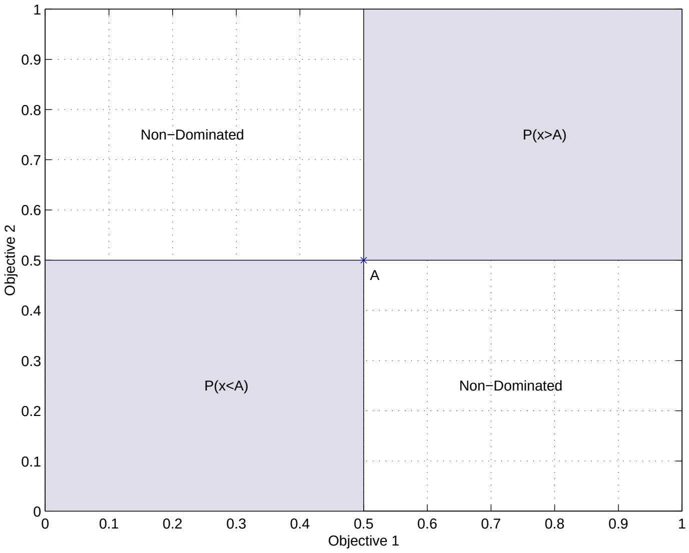

Fig. 3. Noise free non-domination map (maximisation)

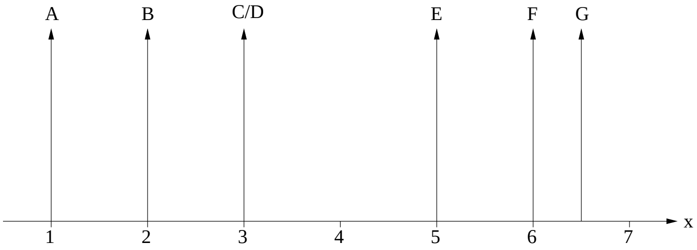

Fig. 4. Fitness values to be ranked

Table 1. Ranks of example fitness values

Value	A	B	C	D	E	F	G
Rank	0	1	2	2	4	5	6

arithmetic series zero to six, therefore the best individual will get a probability of selection of $2/n$ and the worst a probability of zero.

$$P(\text{select}_i) = \frac{(n-1) - R_i}{\sum_{j=1}^{n} R_j} = \frac{2((n-1) - R_i)}{n(n-1)} \tag{19}$$

If we use the rank values in Table 1 with both the tied fitness values being given the best 'untied' rank, we find that the sum of the ranks is no longer consistent, and in this case, $\sum_{j=1}^{n} R_j = 20$. Alternatively, as C & D are tied, it may be better to penalise them both a little and therefore take an average of the rank positions they could have shared, i.e., give them both a rank of 2.5. This would return the overall sum to be 21 and would be consistent, no matter how many fitness values share a rank. This is the method most used for ranking a vector of data.

We can view the ranking process as counting the number of fitnesses that dominate the fitness of interest [8]. If a fitness equal to the current one is encountered, then it is half dominating, and half dominated by the current fitness. Therefore we can create the rank position numbers by this simple counting process. For example, E is dominated by *A, B, C & D* and therefore has a rank of 4. Value C is dominated by *A & B* but is tied with D and so gets a rank of 2.5.

Alternatively, we could consider the dominating / not dominating decision as being the ***probability*** that each fitness value dominates the value of interest. For example, if we consider fitness C, the probability that A dominates C is one. The probability that G dominates C is zero. The probability that D dominates C, from (6) with $m = 0$, is $P = 0.5$. Thus we can represent the rank position as the sum of probabilities of domination as shown in (20), where $P(F_j > F_i)$ is the probability that fitness value j dominates fitness value i.

$$R_i = \sum_{j=1}^{n} P(F_j > F_i) \Big|_{i \neq j} \tag{20}$$

In (20), we have to be sure not to compare fitness F_i with itself. If we did, we would get an extra probability of 0.5 added to the sum. We can therefore include F_i in the sum, but subtract the effect of comparing the fitness with itself. This is shown in (21).

$$R_i = \sum_{j=1}^{n} P(F_j > F_i) - 0.5 \tag{21}$$

As (21) is based on probability, if the fitness values are uncertain, we can use (6) or the approximations (13) or (14) to calculate the probability of domination. For example, if fitness values A to G have a standard deviation of $\sigma = 1$, the rank positions (using (13)) compared to the no noise case are shown in Table 2.

With $\sigma = 0$, we have conventional ranking and the probabilities will range from $2/n$ to zero. If $\sigma = \infty$, all of the fitness values will be assigned the same rank, and will have a probability of selection of $1/n$. Thus the standard deviation of the uncertainty has a similar effect to selective pressure in conventional selection processes [13].

Table 2. Ranks with uncertainty of $\sigma = 0$ and $\sigma = 1$

Value	Rank ($\sigma = 0$)	Rank ($\sigma = 1$)
A	0	0.38
B	1	1.27
C	2.5	2.31
D	2.5	2.31
E	4	4.17
F	5	5.07
G	6	5.49

4.3 Multi-Objective Ranking

With multiple objectives, we now have three possible outcomes from comparing the two fitness values: A dominates B, A is dominated by B, and A and B are non-dominated. If we apply the single objective ranking equation, we find that the total of the rank positions is no longer $n(n-1)/2$ as we now have to account for the non-domination. If we have no noise, for two fitness values where A dominates B, $P(A > B) = 1$, $P(A < B) = 0$, and $P(A \equiv B) = 0$ Therefore when we sum the probabilities of domination, the contribution from this pair will be 1. If the fitness values are non-dominated, the corresponding probabilities are $P(A > B) = 0$, $P(A < B) = 0$, and $P(A \equiv B) = 1$. We have now lost the value 1 from the probability of domination calculations, therefore reducing the sum of ranks total. This state will be the same when we compare A to B and also when we compare B to A, therefore if we sum all the probabilities of non-domination, this will give us twice what was lost from the probability of domination calculations.

If we consider the ranking case for a single dimension, if A and B are identical, we cannot choose between them and so add in 0.5 to the sum. With non-domination, we also have the situation where we cannot choose between objectives and should therefore add 0.5 to the sum as required. In the case of uncertain measurements, we can multiply the value of 0.5 by the probability of non-domination, and still subtract off 0.5 to allow for comparing the individual with itself, thereby maintaining the sum of the rank positions as $n(n-1)/2$. Thus we can add the non-domination term into (21). The rank calculation for multi-objective ranking is shown in (22), where n is the number of measurements being ranked.

$$R_i = \sum_{j=1}^{n} P(F_j > F_i) + \frac{1}{2} \sum_{j=1}^{n} P(F_j \equiv F_i) - 0.5 \tag{22}$$

This *probabilistic ranking* equation allows chromosomes to be selected based on uncertain multi-objective fitness measurements. For the objectives shown in Fig. 5, we can calculate the rankings in order to minimise the fitness values. Table 3 shows ranks (R) for no noise, and 1 standard deviation noise.

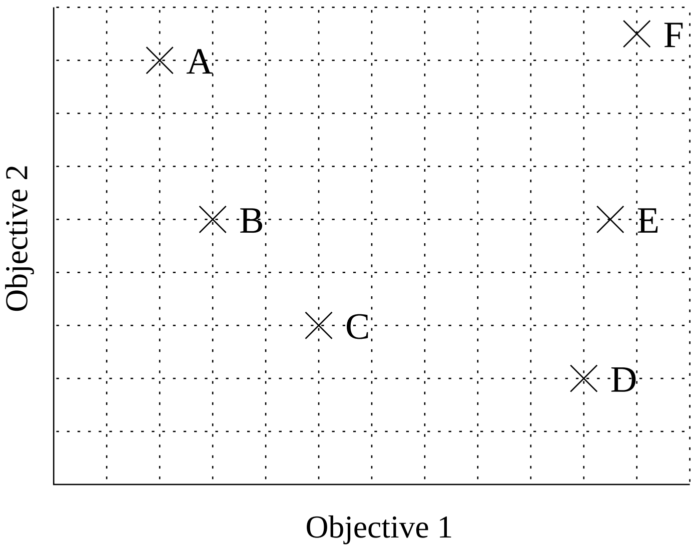

Fig. 5. Multiple fitness values to be ranked

In the example, we see that A is non-dominated with *B, C, D, & E* and therefore gets a rank of 2. Fitness B is non-dominated with *A, C, & D* but shares an objective value with E, thus being half dominating and half non-dominated with E, the rank of B is 1.5 from the three non-dominated points and 0.25 from E, giving a total of 1.75. We also see that each of the columns of Table 3 sums to 15 ($= n(n-1)/2$) as expected. The ranking process is $O(n^2)$, as are many of the other ranking methods [8, 9].

Table 3. Ranks with uncertainty of $\sigma = 0$ and $\sigma = 1$

Value	R ($\sigma = 0$)	R ($\sigma = 1$)
A	2	2.27
B	1.75	1.65
C	1.5	1.42
D	1.5	1.92
E	3.25	3.22
F	5.0	4.53

In the general noisy or uncertain scenario, we see that the proximity of other fitness values, even if only close on one objective, can influence how the rank is assigned. Measurements such as C which are relatively well spaced out on all objectives are ranked more highly than other fitness values that are uncertain.

With no noise, the basic ranking by just counting how many points dominate each fitness measurement described by Fonseca and Flemming [8] is very similar, but does not allow for the non-dominated cases. The sum of the rank values will not be consistent if non-dominated solutions are present, causing a bias towards non-dominated solutions over other solutions. The ranking used by Srinivnas and Deb [9] is based on 'layers' of non-dominated solutions and has no consistency with regards to how many layers, or ranks, are produced, therefore making calculating selection probabilities awkward.

It is interesting to note that if we require an objective to be maximised, setting σ negative will cause the probabilities to be calculated for maximisation, setting σ negative has the same effect as negating the fitness values (the conventional way of converting from minimisation to maximisation). Therefore both minimisation and maximisation objectives may be handled easily by just setting the sign of the corresponding value of σ appropriately.

Limits on objectives, constraints on the chromosomes, and sharing can all be implemented easily within this ranking framework, allowing interactive decision making with uncertain or noisy systems viable. The equations for limits, constraints, and sharing are derived and discussed in [12].

5 Experiment Results

5.1 Introduction

Noise and uncertainty can be split into two broad categories relating to noise that occurs within the process (Type A) and measurement noise (Type B):

1. **Type A Noise:** Noise is applied to the chromosome *before* the objective function is calculated, i.e. $\mathcal{O} = F(\chi + N)$.
2. **Type B Noise:** Noise is applied to the objective function *after* calculation, i.e. $\mathcal{O} = F(\chi) + N$.

Both types of noise are of interest and often the observed noise will be a combination of type A and B.

Trials have been performed to assess how the noise effects the assigned rank position within a population of chromosomes. For the following results, 100 two-parameter chromosomes were generated uniformly distributed in the range [0,1]. A scaled version of the objective function MOP3, defined by Van Veldhuizen and Lamont [10] and given in (23), was used to provide input data to the ranking processes, with either type A or B noise applied as appropriate. The data were ranked and the assigned rank postion for each chromosome recorded. The process was repeated 1000 times with different values chosen for the applied noise each time. For each chromosome, the standard deviation of the rank position was calculated. The mean standard deviation of the 100 chromosome rank positions was then generated and plotted.

$$x = 6\,\chi(1) - 3$$

$$
\begin{aligned}
y &= 6\,\chi(2) - 3 \\
m_1 &= \frac{(x^2+y^2)}{2} + sin(x^2+y^2) \\
m_2 &= \frac{(3x-2y+4)^2}{8} + \frac{(x-y+1)^2}{27} + 15 \\
m_3 &= \frac{1}{x^2+y^2+1} - 1.1e^{-x^2-y^2} \\
\mathcal{O}_1 &= \frac{m_1}{8.249} \\
\mathcal{O}_2 &= \frac{m_2-15}{46.940} \\
\mathcal{O}_3 &= \frac{m_3+0.1}{0.296}
\end{aligned}
\tag{23}
$$

In (23), $\chi(1)$ and $\chi(2)$ are the two parameters of the input chromosome in the range [0,1]. The parameters x and y are scaled to lie within [-3,3] as defined by Van Veldhuizen and Lamont. The three objective functions are then calculated and scaled to give each of the objectives in the range [0,1]. Noise was then applied either to the input chromosome χ for type A noise, or to the output objectives $\mathcal{O}$ for type B noise. The applied noise was Gaussian with a standard deviation of σ.

The ranking algorithms from NSGA and MOGA were generated for comparison with the new multi-objective probabilistic selection evolutionary algorithm (MOPSEA) ranking process developed in this paper. With a different set of 100 initial chromosomes, a slightly different set of graphs will result. The differences have been found to be small however.

5.2 Results

From figures 6 & 7 it is clear that both MOGA and MOPSEA outperform the NSGA ranking process in the presence of noise for this objective function. As the uncertainty parameter σ_n is increased, it is clear that MOPSEA can out perform both alternative algorithms. The specific performance of each algorithm is dependent on the objective function though. Other objective functions are covered in [12].

6 Conclusions

The results have shown that the modified ranking process can reduce the disturbances in the rank positions caused by noisy objectives. Unlike conventional ranking processes, the rank values and therefore the corresponding selection probabilities take some account of the noise and uncertainty in the system. The theory developed in this paper forms an important first step towards addressing directly noise and uncertainty in multi-objective problems. The simplicity of the ranking and selection equations may also provide a route to further theoretical research into the operation and performance of evolutionary algorithms.

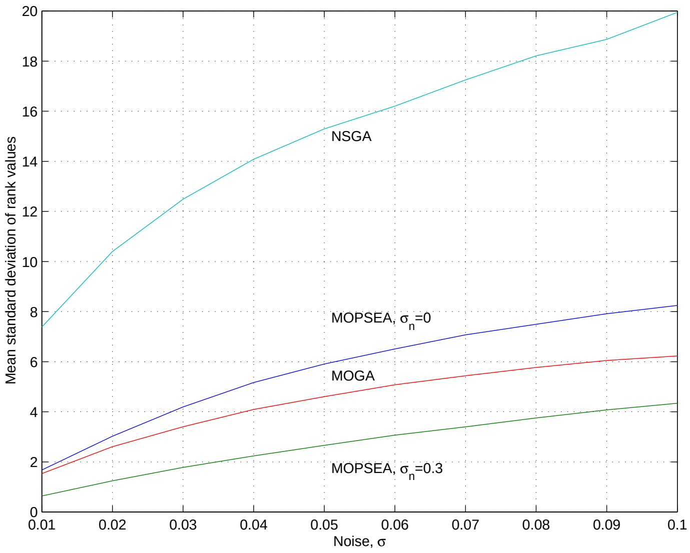

Fig. 6. Applied noise with respect to mean standard deviation of rank position for MOP3, type A noise. Performance of MOGA and NSGA ranking compared to MOPSEA with $\sigma_n = 0$ & $\sigma_n = 0.3$

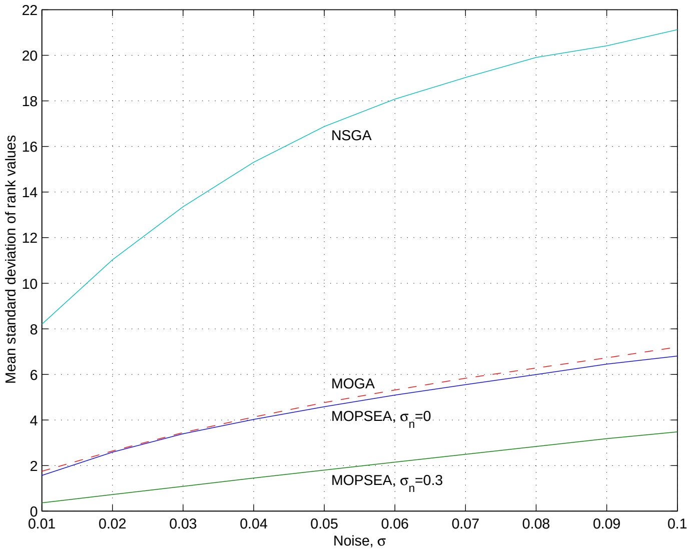

Fig. 7. Applied noise with respect to mean standard deviation of rank position for MOP3, type B noise. Performance of MOGA and NSGA ranking compared to MOPSEA with $\sigma_n = 0$ & $\sigma_n = 0.3$

7 Acknowledgements

The author would like to acknowledge the use of the Department of Aerospace, Power, and Sensors DEC Alpha Beowulf cluster for the production of the results.

References

1. T. W. Then and Edwin K. Chong. Genetic algorithms in noisy environment. In *IEEE International Symposium on Intelligent Control*, pages 225–230, Columbus, Ohio, 16-18 August 1994.
2. Adrian Thompson. Evolutionary techniques for fault tolerance. In *Control '96, UKACC International Conference on*, volume 1, pages 693–698. IEE, 2-5 September 1996. Conf. Publ. No. 427.
3. T. Bäck and U. Hammel. Evolution strategies applied to perturbed objective functions. In *IEEE World Congress on Computational Intelligence*, volume 1, pages 40–45. IEEE, 1994.
4. Shigeyoshi Tsutsui and Ashish Ghosh. Genetic algorithms with a robust searching scheme. *IEEE Transactions on Evolutionary Computation*, 1(3):201–8, September 1997.
5. Kumar Chellapilla and David B. Fogel. Anaconda defeats hoyle 6-0: A case study competing an evolved checkers program against commercially available software. In *Congress on Evolution Computation - CEC2000*, volume 1, pages 857–863, San Diego, CA, 16-19 July 2000. IEEE.
6. Carlos A. Coello Coello. List of references on evolutionary multiobjective optimization. http://www.lania.mx/~ccoello/EMOO/EMOObib.html. Last accessed 3 July 2000.
7. Anna L. Blumel, Evan J. Hughes, and Brian A. White. Fuzzy autopilot design using a multiobjective evolutionary algorithm. In *Congress on Evolution Computation - CEC2000*, volume 1, pages 54–61, San Diego, CA, 16-19 July 2000. IEEE.
8. Carlos M. Fonseca and Peter J. Flemming. Multiobjective genetic algorithms made easy: Selection, sharing and mating restriction. In *GALESIA 95*, pages 45–52, 12-14 September 1995. IEE Conf. Pub. No. 414.
9. N. Srinivas and Kalyanmoy Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1995.
10. David A. Van Veldhuizen and Gary B. Lamont. Multiobjective evolutionary algorithm research: A history and analysis. Technical Report TR-98-03, Air Force Institute of Technology, 1 Dec 1998.
11. William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *NUMERICAL RECIPES in C. The Art Of Scientific Computing.* Cambridge University Press, second edition, 1993.
12. Evan J. Hughes. Multi-objective probabilistic selection evolutionary algorithm. Technical Report DAPS/EJH/56/2000, Dept. Aerospace, Power, & Sensors, Cranfield University, RMCS, Shrivenham, UK, SN6 8LA, September 2000.
13. J. E. Baker. Adaptive selection methods for genetic algorithms. In *Proc. 1st Int. conf. on* Genetic Algorithms, pages 101–111, 1985.