Multi-Objective Equivalent Random Search

Evan J. Hughes

Department of Aerospace, Power and Sensors,
Cranfield University, Shrivenham, Swindon,
Wiltshire, England. SN6 8LA
e.j.hughes@cranfield.ac.uk

Abstract. This paper introduces a new metric vector for assessing the performance of different multi-objective algorithms, relative to the range of performance expected from a random search. The metric requires an ensemble of repeated trials to be performed, reducing the chance of overly favourable results. The random search baseline for the function-under-test may be either analytic, or created from a Monte-Carlo process: thus the metric is repeatable and accurate. The metric allows both the median and worst performance of different algorithms to be compared directly, and scales well with high-dimensional many-objective problems. The metric *quantifies* and is sensitive to the *distance* of the solutions to the Pareto set, the *distribution* of points across the set, and the *repeatability* of the trials. Both the Monte-Carlo and closed form analysis methods will provide accurate analytic confidence intervals on the observed results.

1 Introduction

This paper details a new metric, *Multi-Objective Equivalent Random Search (MOERS)*, that quantifies the performance of an algorithm on an objective function relative to the expected performance of a random search. The metric returns the size of the random search that would be required to achieve results of the same quality. The metric is calculated through a non-parametric statistical analysis of the best solutions found by the optimiser over an ensemble of trials, reducing the chances of occasional overly favourable results biasing the comparison.

The metric focusses on both the median and worst performance of the optimiser under test. Often when we run an optimiser many times, we only remember the good results. Some optimisation routines are capable of providing spectacular results with moderate frequency, but typical results are poor. Other optimisers generate satisfactory results every time, but rarely produce spectacular solutions. If optimisation is being used in a design phase where many repeated runs are possible, then the first algorithm that can produce occasional spectacular results may be preferred. However, in a situation where the optimisation is time-critical, the reliable algorithm is a better choice.

Much research has been performed on different metrics for assessing optimiser performance [7, 6, 5]. The true performance of a multi-objective optimiser cannot be summarised with a single number, or with only a single trial run. For a given test function and optimiser, the MOERS metric provides 10 key performance measures, 5 for the median performance and 5 for the worst-case performance. Each of the 5 is comprised of 2 outer confidence intervals (indicate the distribution of points along the Pareto set), 2 inner confidence intervals (relate to the repeatability of the results) and the median behaviour over the ensemble of trials (indicates the distance to the Pareto set).

In order to remove the effects of constraints, multi-modalities, discontinuous Pareto sets etc., the optimiser results (i.e. the Pareto points) are normalised using the Cumulative probability Density Function (CDF) of the objective surface.[1] The CDF may be calculated analytically from the objective functions and constraints, or generated using a large Monte-Carlo random search (for problems without simple analytic solutions). The metric may be calculated very quickly if an analytic solution to the CDF is used.

Section 2 summarises the Equivalent Random Search metric for single objective problems, and section 3 expands the theory to encompass multi-objective problems. Section 4 demonstrates the metric on a bi-objective function where the analytic CDF has been calcuated, and also demonstrates that a true random search returns the correct metric results. Finally section 6 concludes.

2 Equivalent Random Search Metric

2.1 Introduction

The Equivalent Random Search (ERS) metric assesses the performance of an algorithm, relative to the expected performance of a random search on the same objective function [4]. The metric gives a direct indication of how many points would be needed in a random search to achieve the same solution quality. The metric can then be compared to the actual number of objective evaluations used by the optimiser in order to assess the effectiveness of the optimisation algorithm. For example, if 1000 objective calculations were used by the optimisation process, but the ERS metric reported that 10,000 points would be needed by a random search to achieve equivalent results, then the algorithm is performing well.

2.2 Objective Normalisation

To get a reliable assessment, the optimisation process is repeated M times and the best optima results, $Y_i : i = 1 \ldots M$, gathered. Each of the M results are transformed via the cumulative probability density function of the objective surface, $D(Y)$, to generate the probability of obtaining an objective value *better* than the best value observed in each of the M runs. This normalisation process allows even very rough, multi-modal and deceptive functions to be used for evaluating the optimisers.

The function $D(Y)$ can be generated for any objective functions by performing a large uniform-random sample of the objective surfaces. Further details on Monte-Carlo CDF generation are given in [4]. It is also possible (but not always trivial) to obtain an analytic solution for the CDF if the equations for the objective function are known. This allows the ERS metric to be calculated quickly, but more importantly, will allow functions that have a very low density of points at the Pareto surface to be analysed without resorting to massive Monte-Carlo searches.

[1] Each objective function provides a unique problem to the optimiser, but a set of optimisers may be compared directly if the results are generated for a common multi-objective function.

2.3 Metric Calculation

Once we have calculated the probability of a solution better than Y existing, $D(Y)$, we can compare the result directly to a random search. For the random search, if we generated a single random point, there would be a probability $D(Y)$ that the point would be better than Y, and a probability $1 - D(Y)$ that the point will be worse than Y. If we generate N independent random points, then we will not find a better solution than Y with a probability of $(1 - D(Y))^N$. Therefore the probability of finding *at least one* solution better than Y with an N point random search is given by:

$$D'(Y) = 1 - (1 - D(Y))^N \tag{1}$$

For example, if $D(Y) = 1/1000$ and we generated $N = 100$ random points, the probability of at least one of the N solutions being better than Y is $D'(Y)$ =9.5%.

Importantly, the new cumulative density function, $D'(Y)$ in equation 1, describes the probability that a random search of N points would find an optimum value better than Y. If we repeated an N-point random search M times, $D'(Y)$ would describe the distribution of the M results. Thus the median value of Y from our M searches would be an approximation of the value of Y necessary to make $D'(Y) = 0.5$. We can exploit this property of $D'(Y)$ to create a metric that uses a simple random search as its reference. As the reference can be described analytically, we can use the metric to **quantify** the performance of any optimisation algorithm on any evaluation function.

The ERS metric is calculated by performing M independent runs of our optimisation algorithm under test, and then exploiting (1) to calculate a value for N, given the observed values for Y from our optimiser. By setting $D'(Y_{median}) = 0.5$, where Y_{median} is chosen to be the median result from our M trials of the optimiser, we can calculate the value for N to give us an equivalent size of random search that we would have to perform to achieve the same median result.

Therefore we can re-arrange (1) (and taking logarithms) to give:

$$N_{median} = \frac{\log(0.5)}{\log(1 - D(Y_{median}))} \tag{2}$$

Ultimately, the calculated value for N_{median} is only an estimate and is subject to sampling error (median is calculated by ranking the M values for Y and finding the central value). If we consider that the probability of the true value of N_{median} being less than the estimate is 0.5, and the probability of the true value being greater is also 0.5, we can describe the error in the estimate using a binomial distribution of the rank locations with the two probabilities being $p = 0.5$ and $q = 1 - 0.5$. A binomial distribution can be approximated by a normal distribution when $Mp \geq 5$ and $Mq \geq 5$. Thus a minimum value of $M = 10$ trials will suffice. The variance is given by $Mpq = M/4$ and therefore the standard deviation by $\sigma = \sqrt{M}/2$. The 95% confidence limits of a normal distribution are given by $\pm 1.96\sigma$. Therefore the the upper and lower bounds to give 95% confidence intervals on the estimate of the median correspond to the values of Y from the ranked data in indexes $(M + 1)/2 \pm 1.96\sqrt{M}/2$.

We can also process other statistics such as the best and worst values of Y and associate them to the best and worst values expected from a random search. In practice, the *best* value found is subject to very wide confidence bounds unless a very large M is used (for example, to find the 99$^{\text{th}}$ percentile, $p = 0.01$, $q = 0.99$, $\therefore M > 500$). However although the estimate of the *worst* value of Y should also require a large number of samples, in practice it is far better behaved, and also a far more useful indicator of algorithm performance.

For the random search, the probability given by $D'(Y)^M$ is the probability that M searches will all return values better than Y. Therefore the cumulative probability distribution in (3) is the distribution of probabilities that at least one worse value than Y will be found in M trials. The distribution $D''(Y)$ is therefore the distribution of the worst optimisation results from M trials.

$$D''(Y) = 1 - D'(Y)^M \tag{3}$$

Equation 4 shows equations (3) and (1) re-arranged to obtain a median estimate and the 95% confidence limits of the worst optimisation value.

$$\begin{aligned} N_{worst_{upper}} &= \frac{\log(1 - \sqrt[M]{0.025})}{\log(1 - D(Y_{worst}))} \\ N_{worst_{median}} &= \frac{\log(1 - \sqrt[M]{0.5})}{\log(1 - D(Y_{worst}))} \\ N_{worst_{lower}} &= \frac{\log(1 - \sqrt[M]{0.975})}{\log(1 - D(Y_{worst}))} \end{aligned} \tag{4}$$

The metric N_{median} in (2) is the size of the random search optimisation that must be performed, that when repeated M times, will obtain a median optima Y_{median}. This metric is an indicator of typical algorithm performance (distance to the true global objective value) and a value of N_{median} larger than the actual number of function evaluations used indicates an optimisation algorithm well suited to the test function.

The metric N_{worst} in (4) is the size of the random search optimisation that must be performed, that when repeated M times, will obtain a worst optima of Y_{worst}. If this metric is larger than N_{median}, then the spread of the inferior solutions from the optimisation process (i.e. variance of inferior solutions) is smaller than the spread that would be obtained by a random search process. This is a desirable feature of optimisation algorithms as it suggests that if only a single run of the optimiser can be performed, there is confidence that a good solution will be identified. If N_{worst} is smaller than N_{median}, the optimiser is prone to premature convergence on poor solutions (highly undesirable).

Any situations that give N_{median} lower than the actual number of evaluations used indicate that a random search would have most likely provided better results than from the optimiser.

3 Multi-Objective Equivalent Random Search

The extension to multi (and many) objective problems is straightforward. To assess the quality of M non-dominated surfaces generated from the optimisation algorithm under test, each objective vector in each non-dominated set can be combined using an aggregation function to allow a set of single-objective metrics to be calculated. For assessing

Pareto sets, the weighted min-max aggregation function in equation 5 is suitable, but alternative metrics may be used if desired (e.g. for assessing objective surfaces).

The weighted min-max score of k objectives is calculated using equation 5,where w_i is the weight of the i$^{\text{th}}$ objective, O_i. Weighted min-max is able to generate points on both convex, concave and discontinuous Pareto sets.

$$Y = \max_{i=1}^{k}(w_i O_i) \tag{5}$$

As the weight vector is changed, the aggregated objective surface is modified and the cumulative probability distribution is modified. If a large random search has been performed of the objective space, then the results can be transformed by the aggregation function and sorted to form a CDF. Although the vectors will not be truly independent, one large random sampling of the objective space may be re-cycled for calculating the CDFs for any weight vector set (equation 7 shows an example analytic CDF).

The multi-objective assessment is performed by first generating a set of H weight vectors (typically $H = 100$ or more), giving $W_j : j = 1 \ldots H$, that span the true Pareto surface (or the non-dominated surface of the large random sampling). Each of the vectors W_j corresponds to a full set of weights, $W_j = [w_{1j}\ w_{2j}\ \ldots w_{kj}]$.

For each of the $j \in H$ weight vectors, all the $n \in M$ non-dominated surfaces are scanned in turn, aggregating using W_j. The best performing aggregated point from each of the M non-dominated surfaces is gathered, Y_{nj}, resulting in M best points for each of the H test weight vectors.

Thus each of the sets of M non-dominated surfaces can be assessed using the single objective theory. If we take the first weight vector W_1 for example, we can take the worst and median of the $Y_{n1} : n \in M$ aggregated values, and process using the CDF corresponding to the vector W_1 with equations 2 and 4 and therefore obtain the performance in the direction of the weight vector W_1. The process can be repeated for all of the H weight vectors, yielding vectors of results for $\boldsymbol{N}_{median}$, $\boldsymbol{N}_{worst}$ and their associated confidence intervals. Sorting the vectors $\boldsymbol{N}_{median}$ etc. will create CDFs of the performance across the entire Pareto surface.

The median of the $\boldsymbol{N}_{median}$ vector can be used as a good indicator of general performance, but for the most accurate representation, the overall 95% limits on the $\boldsymbol{N}_{median}$ vector should also be reported (the ‘outer’ confidence interval), along with the analytic 95% confidence limits on the median of $\boldsymbol{N}_{median}$ (the ‘inner’ confidence interval). As the Pareto surface is being analysed using an ensemble of aggregations, the ‘single objective’ problem that is being analysed is being changed as we scan the Pareto set with the weight vectors. Thus it is likely that the optimiser under test may perform differently on different regions of the Pareto set (typically the edges of the Pareto surface are different to the central region) and a spread of ERS values that is wider than the analytic error will be observed. The spread (especially the lower limit) can be very informative about the reliability of the optimiser at identifying solutions. The ‘outer’ confidence limits are calculated from this spread and relate closely to the distribution of solutions across the Pareto set, as any gaps in coverage will result in a low ERS value for the lower ‘outer’ confidence limit. The multi-objective ESR metrics are be denoted by the quintet of results: $[N^{-}_{median_{outer}}\ \ N^{-}_{median_{inner}}\ \ N_{median}\ \ N^{+}_{median_{inner}}\ \ N^{+}_{median_{outer}}]$, abbreviated to $[N^{--}_{M}\ \ N^{-}_{M}\ \ N_{M}\ N^{+}_{M}\ \ N^{++}_{M}]$.

A similar set of results can be provided for the worst case performance too: $[N_W^{--}\ N_W^{-}\ N_W\ N_W^{+}\ N_W^{++}]$. The outcome is 10 numbers that summarise the equivalent random search sizes that would be required to mimic the performance distribution of the optimiser under test.

4 Example Analytic Density Function

The equation for the cumulative density function under the weighted min-max aggregation function has been derived for a simple multi-objective test function. Equation 6 details the function, and the objective space is depicted pictorially in Fig. 1.

$$\begin{aligned} O_1 &= \sqrt[v]{x} \\ O_2 &= \sqrt[v]{y} \\ 1 \leq O_1 + &O_2 \qquad 0 \leq x, y \leq 1 \end{aligned} \tag{6}$$

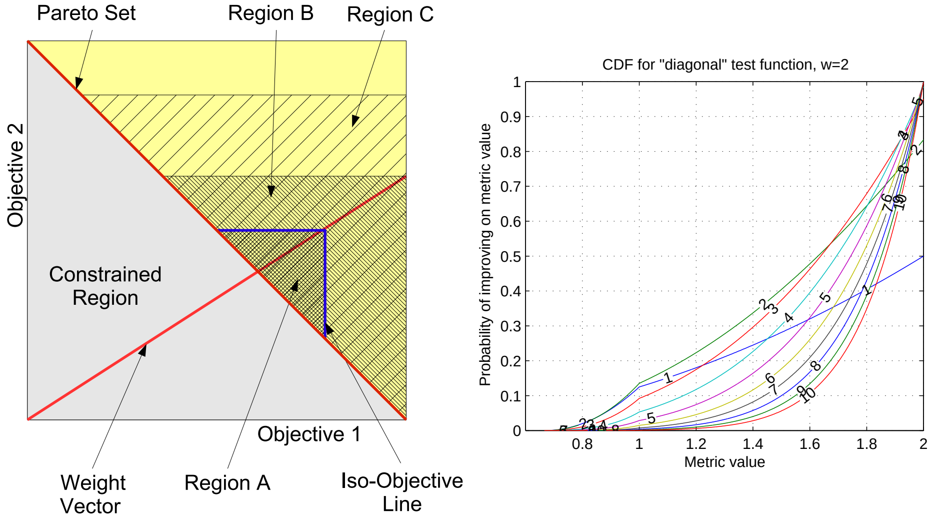

Fig. 1. Objective space and Cumulative Probability Density function for the "diagonal" function.

In Fig. 1, the Pareto surface is simply a diagonal line across the objective space which is defined by the constraint boundary of the feasible region $O_1 + O_2 \geq 1$. To calculate the cumulative density function $D(Y)$, the two objectives are combined using equation 5 to form the metric Y. If the points in objective space that result in a constant metric value of Y are plotted, an *iso-objective* contour results and a typical example is shown in blue on Fig. 1. If the distribution of points in the objective space is uniform (when $v = 1$), then $D(Y)$ is simply the ratio of the area of the superior feasible region bounded by the iso-objective line defined by Y (region A in the figure), to the total area of the objective space. If the distribution of the objectives is not uniform, then $D(Y)$ is the area integral of the probability density of the objectives bounded by the constraints and the iso-objective contour defined by Y.

In the example shown in Fig. 1, $D(Y)$ will grow as a square-law relationship until the iso-objective line reaches the boundary of region B, then will progress on a compound law thereafter (e.g. as seen in region C). The resulting equation for $D(Y)$ is given in equation 7, where $w = \max(w_1, w_2)$, $\min(w_1, w_2) = 1$, w_1 and w_2 are the weights applied to objective 1 and 2 respectively and v is a shape parameter that describes the density of the Pareto set. An integer shape parameter in the range $[2, 10]$ provides a useful range of difficulty for the optimisation process.

$$D(Y) = \begin{cases} \frac{Y^{2v}}{w^v} - Y^v(1-Y)^v - v \sum\limits_{r=0}^{v} \binom{v}{r} \frac{(-1)^r}{v+r} \left[\left(\frac{Y}{w}\right)^{v+r} - (1-Y)^{v+r} \right] & Y < 1, \\ \frac{Y^v}{w^v} - v \sum\limits_{r=0}^{v} \binom{v}{r} \frac{(-1)^r}{v+r} \left(\frac{Y}{w}\right)^{v+r} & Y \geq 1. \end{cases} \tag{7}$$

Figure 1 shows the cumulative density function for equation 6 and $w = 2$, for v over a range [1,10]. The 'knee' in the CDF when $Y = 1$ is visible clearly. At low values for v, there are a large number of constrained solutions and the density of the solutions at the Pareto surface is high. With a high density of solutions, the random search performs well and the optimisers have difficulty improving on the random solutions. At high values of v, there are very few constrained solutions, but the Pareto set density is low and the random search is not so good. There is more scope for improvements by the optimisation algorithms. A good general-purpose optimiser will perform satisfactorily across a wide range of Pareto set densities.

5 Performance Trials of Optimisers

To illustrate the metric, two alternative optimisation strategies have been tested against the objective function in equation 6 with a shape parameter of $v = 5$ to provide a Pareto set with a reasonably low density. As a baseline, random search has been used to confirm that the MOERS metric is truly relative to the analytic random search process. The second optimiser is NSGA-II using software downloaded from the algorithm authors website [1]. In order to allow independent verification, the Matlab software for the MOERS metric used to generate the results in this section is available at [3].

Figure 2 shows the different metrics as assessed from NSGA-II [2] on the objective function in equation 6 (with a shape parameter of $v = 5$) using 5000 actual objective calculations, $H = 200$ weight vectors and $M = 100$ repeated trials. On the figure, the horizontal solid line indicates $log_{10}(5000) = 3.7$, the upper varying solid line is the median equivalent random search size ($\boldsymbol{N}_M$) and the lower varying solid line is the worst-case equivalent random search size ($\boldsymbol{N}_W$). The dashed lines indicate the upper and lower bound of the analytic (inner) 95% confidence intervals ($\boldsymbol{N}_M^+$, $\boldsymbol{N}_M^-$ and $\boldsymbol{N}_W^+$, $\boldsymbol{N}_W^-$). The horizontal dashed lines indicate the locations of N_M^{++} and N_M^{--} and N_W^{++} and N_W^{--} (outer confidence limits).

The corresponding equivalent random search metrics are shown in table 1. Table 2 shows the *Log Search Ratio* (LSR) which is the logarithm of the ratio of the ERS metric

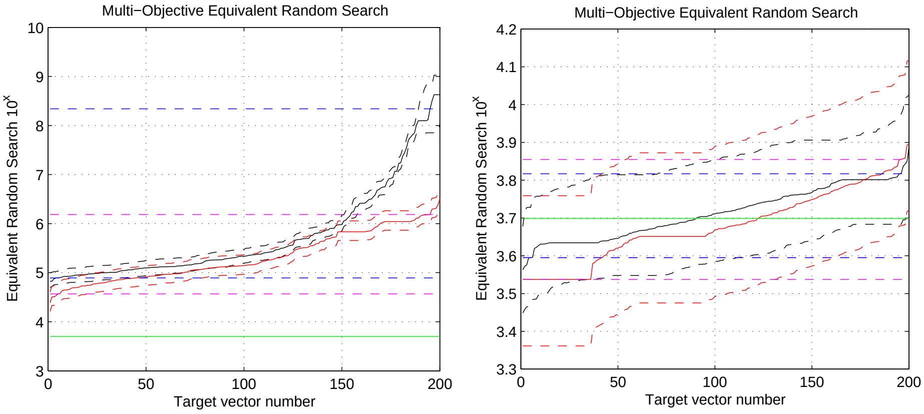

Fig. 2. Multi objective equivalent random search metric for NSGA-II (left) and random search (right) on test function equation 6

over the actual number of evaluations used. A LSR of zero would indicate that the optimiser is equivalent to a random search.

Table 1. Equivalent random search for NSGA-II with equation 6 and 5000 evaluations.

Metric	N^{--}	N^{-}	N	N^{+}	N^{++}
N_{worst}	36914	93682	**140460**	233808	1534218
N_{median}	78128	157795	**217641**	306711	217931129
$\log_{10}(N_{worst})$	4.57	4.97	**5.15**	5.37	6.19
$\log_{10}(N_{median})$	4.89	5.20	**5.34**	5.49	8.34

Table 2. Log Search Ratio for NSGA-II with equation 6 and 5000 evaluations.

Metric	LSR^{--}	LSR^{-}	LSR	LSR^{+}	LSR^{++}
LSR_{worst}	0.87	1.27	**1.45**	1.67	2.49
LSR_{median}	1.19	1.50	**1.64**	1.79	4.64

It is clear from Fig. 2 that there is a definite improvement over the analytic random search with the optimisation algorithm as all metrics are above the horizontal solid line (and therefore all LSR values are positive). The shallow slope of $\boldsymbol{N}_M$ and $\boldsymbol{N}_W$ on the left of the graph indicates that there are no large gaps in the Pareto set in any of the 100 trials. The $\boldsymbol{N}_W$ and $\boldsymbol{N}_M$ curves are near coincident for the worst 150 weight vectors, indicating that the CDF of the spread of the optima is the same shape as would be expected from a random search (a good feature). The results for $\boldsymbol{N}_W$ in the right-hand 50 weight vectors however are lower than the $\boldsymbol{N}_M$ results and suggests that the worst case results are spread much further than the $\boldsymbol{N}_M$ equivalent random search would provide. This indicates that the behaviour is becoming erratic over a few small regions of the Pareto surface and the algorithm is converging to local solutions. The good median performance shown on the right-hand-side is due to the random search not being good at finding the extremes of the Pareto set for the test problem. NSGA-II however, obtains a good spread of results right across the Pareto set. Hence the best of the H median ERS vectors are at the edges of the Pareto set, and the worst at the centre. An ideal algorithm would have $\boldsymbol{N}_W$ consistently higher than $\boldsymbol{N}_M$ demonstrating a very robust optimiser whose bad results are still very good. Overall the assessment is that NSGA-II is performing well, a random search would need approximately 43 times as many points ($10^{1.64}$) to achieve equivalent optima.

Table 3. Log Search Ratio for random search with equation 6 and 5000 evaluations.

Metric	LSR^{--}	LSR^{-}	LSR	LSR^{+}	LSR^{++}
LSR_{worst}	-0.16	-0.21	**-0.03**	0.19	0.16
LSR_{median}	-0.10	-0.11	**0.01**	0.15	0.12

Figure 2 and table 3 shows the results of performing a 5000 point random search on equation 6. The search was repeated $M = 100$ times and is assessed over $H = 200$ weight vectors. The horizontal solid line shows the actual number of points used and it is clear that the metrics all lie within the anticipated 95% confidence intervals predicted from the median values of the ensemble of weight vectors. The random search test is very useful as it confirms the correctness of the analytic equations.

The two optimisation processes have been tested further with 10 different maximum number of evaluations in the range [800, 500000]. At each configuration $M = 100$ trials were performed in order to allow the MOERS metrics to be calculated with useful confidence intervals. The MOERS results were converted to the Log Search Ratio so that a direct comparison with the performance against the analytic random search can be made as the algorithm computational allowance is increased.

Figure 3a shows a graph of $[LSR_M^{--}\ \ LSR_M^{-}\ \ LSR_M\ LSR_M^{+}\ \ LSR_M^{++}]$ for a range of different search sizes with NSGA-II. It is clear that the performance relative to the analytic random search improves as the number of evaluations allowed increases, but eventually, the performance ‘saturates’. Different optimisers saturate at different levels and the saturation is an indication of intrinsic optimisation capacity on the function under test.

Figure 3b shows a graph of $[LSR_M^{--}\ \ LSR_M^{-}\ \ LSR_M\ LSR_M^{+}\ \ LSR_M^{++}]$ for a range of different search sizes and the random search algorithm. It is clear that the Log Search Ratio is a good approximation to zero, i.e. the actual number of points used matches the analytic prediction that was based on the $M = 100$ non-dominated sets that were analysed. It is also clear that LSR_M^{--} is very similar to LSR_M^{-}, and LSR_M^{+} is very similar to LSR_M^{++}, demonstrating that the analytic confidence intervals are accurate.

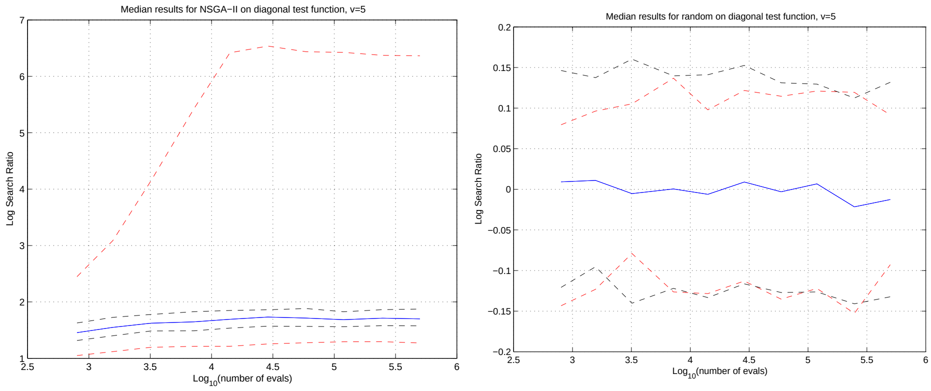

Fig. 3. Plot of Log search ratio of NSGA-II and random search for different search sizes on equation 6

6 Conclusions

This paper has introduced a new metric for assessing the performance of multi-objective optimisation algorithms. The metric uses an analytic random search as the reference, allowing performance to be *quantified*. The metric requires multiple independent runs of the optimiser and assesses both median and worst performance, and provides analytic confidence intervals on the results. The metric is both repeatable and accurate.

References

1. K. Deb. NSGA-II code in C. http://www.iitk.ac.in/kangal/codes/nsga2/nsga2-v1.1.tar.
2. K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature – PPSN VI*, pages 849–858, Berlin, 2000. Springer.
3. E. J. Hughes. MOERS example software. http://code.evanhughes.org.
4. E. J. Hughes. Assessing robustness of optimisation performance for problems with expensive evaluation functions. In *World Congress on Computational Intelligence*, Vancouver, Canada, July 2006. IEEE. to appear.
5. J. Knowles and D. Corne. On metrics for comparing non-dominated sets. In *Congress on Evolutionary Computation (CEC 2002)*, volume 1, pages 711–716, Hawaii, 2002.
6. T. Okabe, Y. Jin, and B. Sendhoff. A critical survey of performance indices for multi-objective optimization. In *Congress on Evolutionary Computation (CEC'2003)*, volume 2, pages 878–885, Canberra, Australia, Dec. 2003.
7. E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. In *IEEE Transactions on Evolutionary Computation*, volume 7, pages 117–132, April 2003.