A Tri-Phase Multimodal Evolutionary Search Performance Profile on the 'Hierarchical If and Only If' Problem.

Martin J. Oates

British Telecom, Adastral Park,
Martlesham Heath, Suffolk, UK
martin.oates@bt.com

David.Corne, Roger Loader

Department of Computer Science,
University of Reading, UK
{D.W.Corne, Roger.Loader}@reading.ac.uk

Abstract

Previous work by the authors has explored performance profiles of simple evolutionary algorithms over a range of problems. These studies have revealed a consistent bimodal feature, where the optimal mutation rate occurs at the trough between two peaks in convergence time. However, careful examination revealed anomalies suggesting higher modality features in the performance profile under certain conditions. Here we report on a detailed examination of the performance profile of Watson et al's H-IFF problem, aimed at further exploring and understanding this multimodal performance profile. We find that the series of troughs correspond to a series of mutation rates which each seem tuned towards increasingly better suboptima. Further, by examining the curves of mean fitness, convergence time, and the variance of convergence time, we can identify a three-phase nature to the profile; i.e.: search behaviour cycles through three distinct phases, which repeat in synchronisation with peaks and troughs. These findings seem important in relation to the need for robust and reliable parameter tuning.

1 INTRODUCTION

The needs of real-world applications impel researchers to deliver robust evolutionary algorithms [4,7] which perform within challenging constraints of reliability, time limit, and solution quality. Typically therefore, an evolutionary algorithm deployed in a real world scenario must be tuned for optimal performance in a given time limit. In this context, researchers have tended to focus on finding optimal parameters (e.g.: mutation rate, population size) for which solution quality tends to be reliably best, within the time available, on what are deemed to be suitably realistic test problems depending on the application in hand. Such studies almost invariably find, for example, that a single optimal mutation rate (or small interval of rates) exists; that is, the solution quality/mutation rate curve is fundamentally uni-modal.

In many cases, however, the demands of near real-time applications make it sensible to delve deeper into the performance profile. In particular, it is often true that, at the optimal mutation rate, good solutions are found significantly faster than the assigned time limit. For example, although we may empirically find that a mutation rate of 0.1 leads to best mean fitness at a time limit of 10 minutes, over 50% of runs at that mutation rate might converge to good quality or perhaps optimal solutions in just 1 minute. For primarily this reason, two other performance indicators are of particular importance: `evaluations exploited', and its variance. "Evaluations exploited" is simply the time (measured in number of evaluations) at which a trial run first finds the best solution of that run (hence, it continues from then until the time limit without finding a better solution). This measure, and its variance, obviously provide important information about the performance profile, respectively indicating to what degree it may be useful and advisable to exploit fast convergence.

In previous work which has looked at a real-world problem in the telecommunications field (the ADDMP [12-19]) we have found that the performance profile yielded by a plot of evaluations exploited against mutation rate is *bimodal*. This bi-modality seems robust, appearing over a wide range of problems and EA designs [17-19]. As well as in the ADDMP, we have seen this bimodal performance profile in the Royal Staircase problem [11], Kauffman NK landscapes [5], and the simple Max-Ones problem. The optimal mutation rate, in terms of delivering the best mean fitness, seems to generally occur at the base of the trough between two peaks in evaluations exploited. This mutation rate tends to correspond with the well known 1/L rate [1,2,9], while the bimodal feature itself has recently been predicted in theoretical studies by Van Nimwegen et al [10,11, and personal communications]. These and other findings concerning features of `evaluations exploited'-oriented evolutionary search performance profiles, may have potentially important consequences and applications for parameter tuning and performance guarantees.

In particular, we have recently found evidence that complex problems seem to have a tri-modal or higher modality feature in their evaluations exploited / mutation rate performance profiles. The peaks and troughs seem to loosely correspond to different sub-optima of the problem landscape, and to different phases of search behaviour. In this paper we demonstrate and explore these higher modality features on an interesting recently developed test problem called H-IFF (Hierarchical If and Only If) developed by Watson et al [22,23].

In section 2 we describe the H-IFF problem and preliminary results. Section 3 then describes the experimental set-up and extensive series of experiments reported on in this paper. The results are explored in section 4 and discussed in section 5. As detailed, we find clear evidence of tri-modality in the H-IFF performance profile, and are also able to discern certain discrete, repeating 'phases' of search behaviour corresponding to increasingly smaller intervals of mutation rates. Section 6 summarises our conclusions, whilst Sections 7 and 8 respectively express our acknowledgements and detail references.

2 THE H-IFF PROBLEM

Watson et al's Hierarchical If and only If problem (H-IFF) [22,23] was devised to explore the performance of search strategies employing crossover operators to find and combine 'building blocks' of a decomposable, but potentially contradictory nature. An earlier problem designed on similar lines is the bipolar deceptive function [24]. The fitness of a potential solution to this problem is the sum of weighted, aligned blocks of either contiguous 1's or 0's and can be described by :

$$f(B) = \begin{cases} 1, & \text{if } |B| = 1 \\ |B| + f(B_L) + f(B_R), & \text{if } (|B| > 1) \text{ and } (\forall i\ \{b_i = 0\} \text{ or } \forall i\ \{b_i = 1\}), \\ f(B_L) + f(B_R), & \text{otherwise} \end{cases}$$

where B is a block of bits, $\{b_1, b_2, \ldots b_n\}$, $|B|$ is the size of the block=n, b_i is the *i*th element of B, and B_L and B_R are the left and right halves of B (i.e. $B_L = \{b_1, \ldots b_{n/2}\}$, $B_R = \{b_{n/2+1}, \ldots b_n\}$. n must be an integer power of 2.

This produces a search landscape in which 2 global optima exist, one as a string of all 1s, the other of all 0's. However a single mutation from either of these positions produces a much lower fitness. Secondary optima exist at strings of 32 contiguous 0's followed by 32 contiguous 1's (for a binary string of length 64) and vice versa. Again, further sub-optima occur at 16 contiguous 0's followed by 48 contiguous 1's etc. Watson et al showed that hillclimbing performs extremely badly on this problem [23].

To establish a performance profile for a simple evolutionary search technique on this problem, a set of tests were run using a simple EA (described later) over a range of population sizes (20 through 500) and mutation rates (1e–7 exponentially through to 0.83), noting the fitness of the best solution found, and the number of evaluations taken to first find it out of a limit of 20,000 evaluations. Each trial was repeated 50 times and the mean number of evaluations used is shown in Figure 1. This clearly shows a tri-modal performance profile, particularly at lower population sizes, and was first reported in [20]. Since this result was first seen, an extensive set of further experiments has been run with higher evaluation (time) limits, and variations of selection strategy and crossover operator.

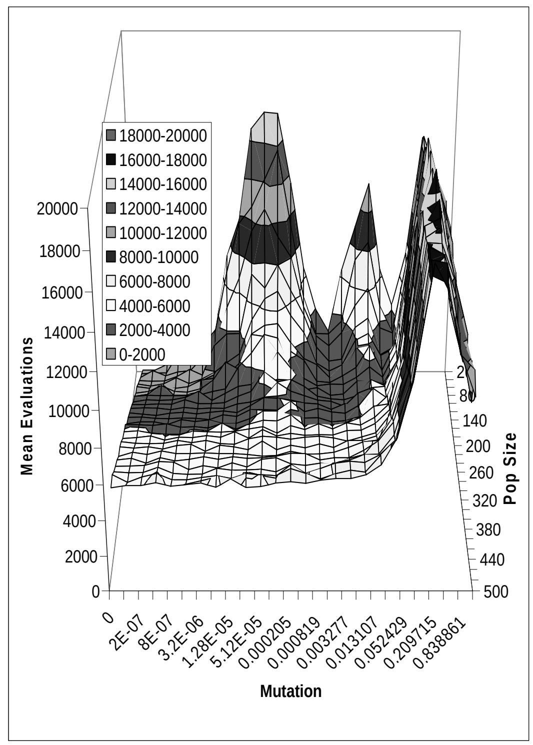

Figure 1 – H-IFF 64 Performance Profile

Some results and observations from these experiments are reported on in this paper, together with evidence that these phenomena are also exhibited in the performance profile of an example real-world problem.

3 METHOD

Unless otherwise stated, all results in this paper were generated using a steady state EA [3] with 3-way single tournament selection (with automatic replacement of the worst member of the tournament) with a fixed number of evaluations, one-point crossover with probability 1.0, and 'new random allele' (NRA) mutation. Each experiment was repeated 50 times, noting the evaluation number at which the best solution found in the run was first generated, and the fitness of that best solution. All

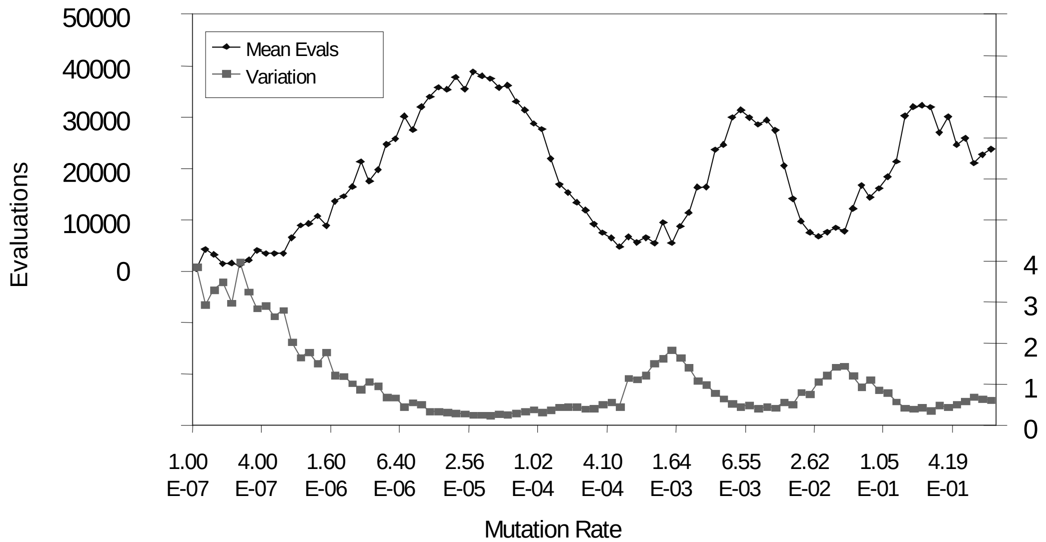

Figure 2 – Mean Evaluations and Variation at 50k evals

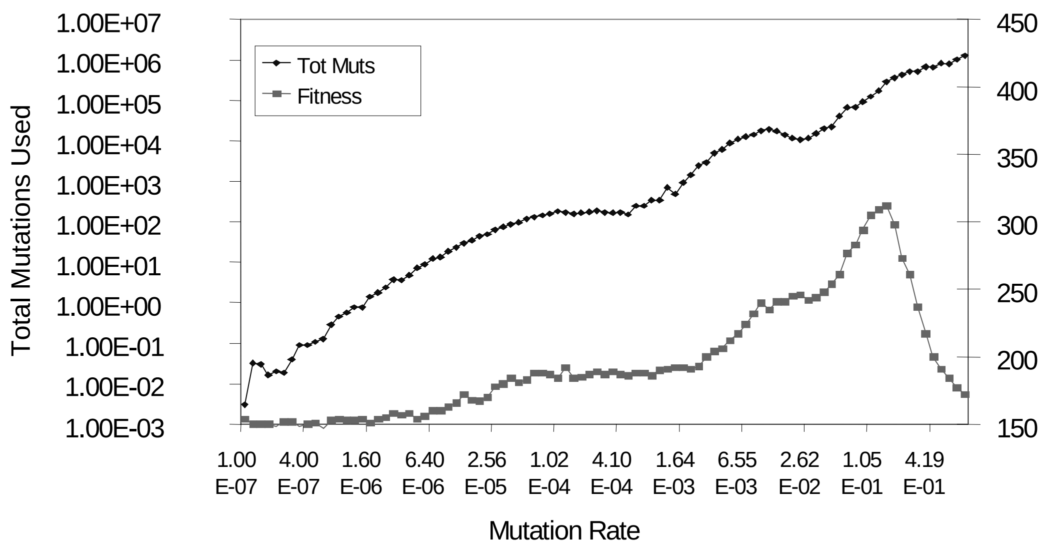

Figure 3 – Mutations used and Fitness at 50k evals

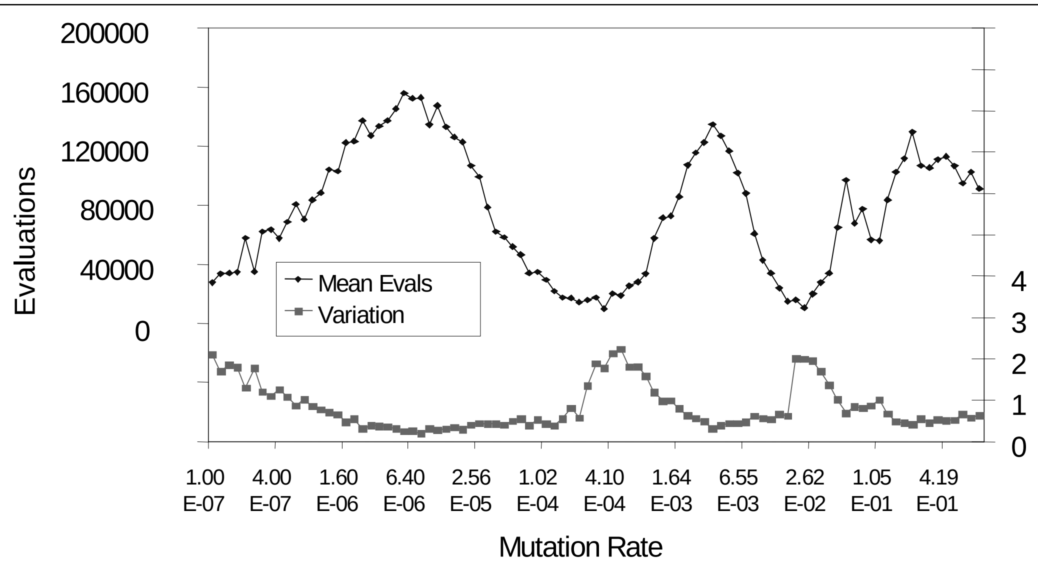

Figure 4 – Mean Evaluations and Variation at 200k evals

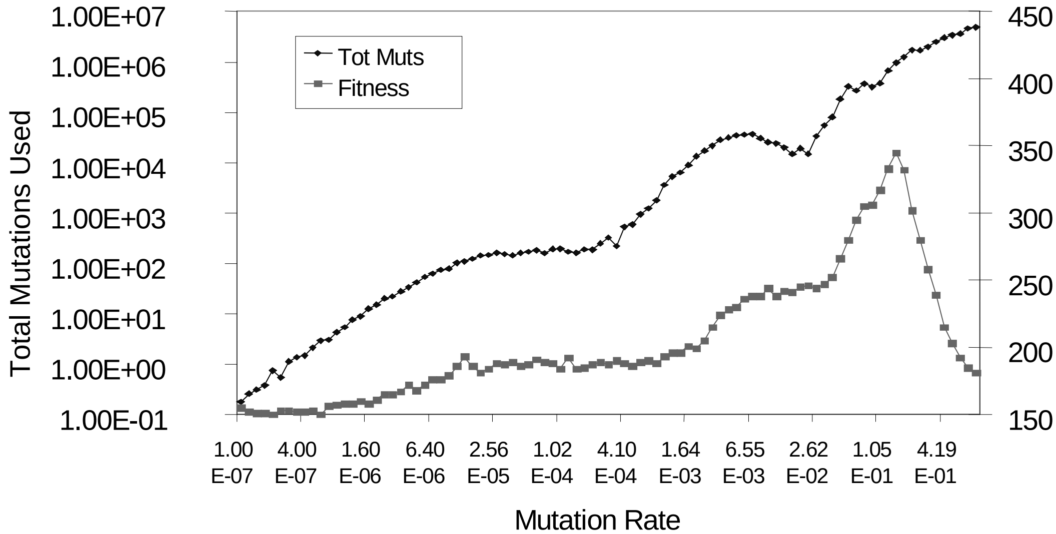

Figure 5 – Mutations used and Fitness at 200k evals

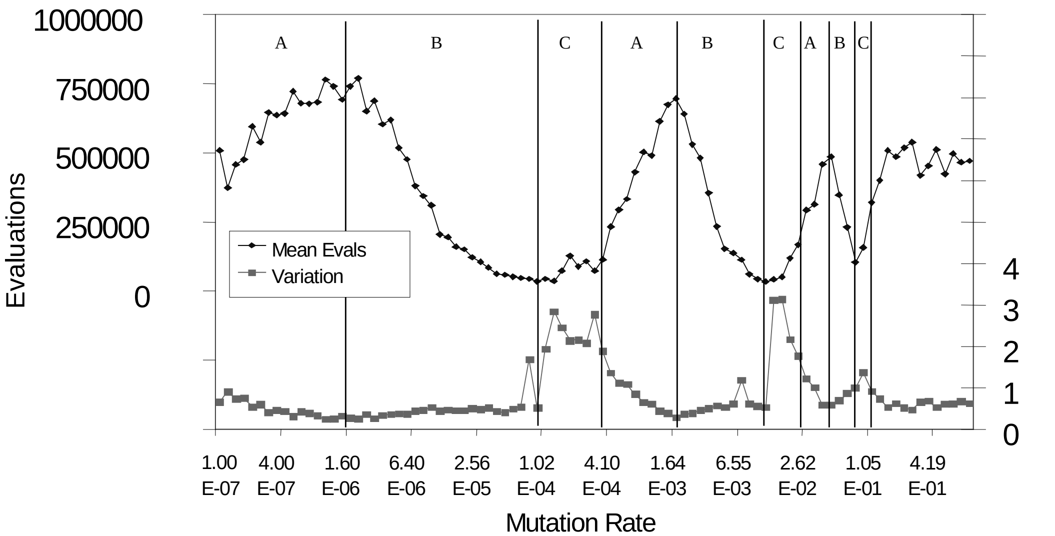

Figure 6 – Mean Evaluations and Variation at 1M evals

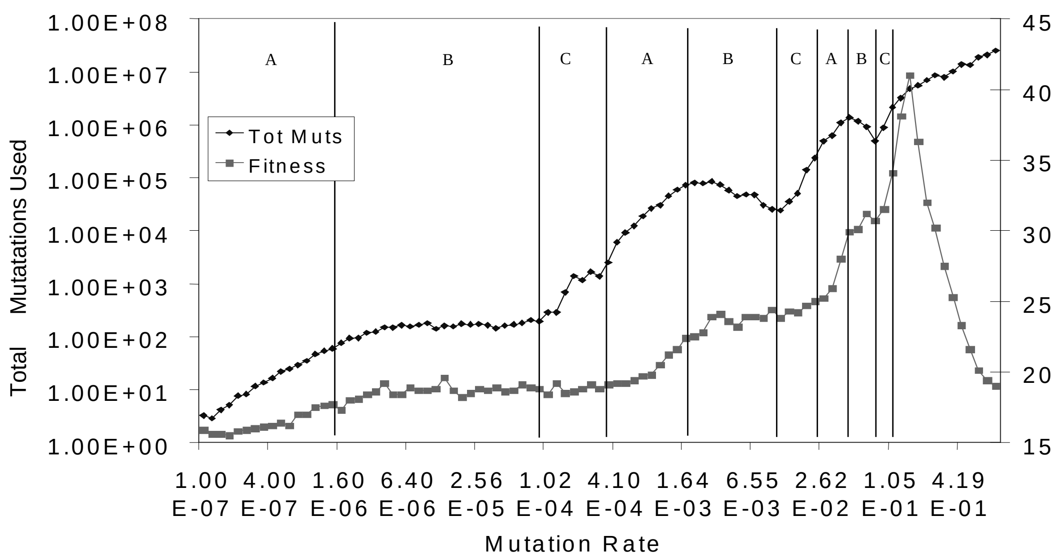

Figure 7 – Mutations used and Fitness at 1M evals

Runs were then repeated over a range of mutation rates from 0, 1e–7, then doubling at each step to 0.83. In some cases, a four-fold finer range of mutation rates was used.

For most experiments, a population size of 20 was used, whilst for others, a series of experiments were carried out at population sizes ranging from 20 to 500 members in steps of 20. For the H-IFF problem, a string length of 64 was used, whilst the ADDMP used a natural integer representation with 10 genes each having an allele range of 1 to 10.

Two other values are also plotted within the results: the 'coefficient of variation' is defined as the standard deviation of the 50 results, divided by the mean, which gives an indication of process instability; secondly the 'total mutations used', which is estimated as being the product of the mean number of evaluations used, the mutation rate per gene and the chromosome length.

4 RESULTS

Figure 1, as described in Section 2, clearly shows a tri-modal performance profile in terms of mean evaluations exploited at different mutation rates and population sizes for the 64 bit H-IFF problem. Clearly, at very low mutation rates, the algorithm stalls almost immediately, with premature convergence having depleted the limited amount of genetic diversity available to it from its initial randomly generated population. As population size is increased, the number of evaluations utilisable before this situation occurs is seen to rise linearly with population size. This result has already been seen on other problems and is documented in [13,17,18]. It is also important to note however that average fitness of best solution found also increases with population size in an asymptotic fashion.

However, for a fixed population size, as the mutation rate is increased, the number of evaluations exploitable is seen to initially rise. This reaches a peak at a mutation rate of around 5.1e–5, beyond which the number of evaluations utilised is seen to fall. This represents the algorithm finding solutions of a particular quality for which a specific range of mutation rates are particularly useful. In other words, the algorithm can use the increased mutation rate to find particular solutions in fewer evaluations leading to the trough feature at a rate of 1.6e–3. However this range of mutation rates is not optimal for finding higher fitness solutions and thus as rates are increased further, a second rising and falling of the number of evaluations used is observed between mutation rates of 1.3e–2 and 5.2e–2. Eventually, mutation rates are sufficiently high as to hinder progress, and the number of evaluations is seen to rise for a third time, with even higher rates causing the algorithm to deteriorate into random search. These phenomena are seen to also exist at higher population sizes, at least as high as 100 in this case, and at the same rates of mutation.

The above explanation is explored over the next six figures, as plots of the mean number of evaluations, coefficient of variation, mean fitness, and total number of mutations used are shown against various mutation rates in experiments limiting trials to 50,000, 200,000 and 1 million evaluations for each of the 50 runs.

Figure 2 shows the performance profile for a population size of 20 with the EA allowed 50,000 evaluations. Distinct peaks in the mean number of evaluations can be seen at mutation rates of 2.56e–5, 5.51e–3 and 2.10e–1 with an apparent anomaly at 6.23e–2. Troughs occur at rates of 4.87e–4 and 2.62e–2 which at first appear to correspond to peaks in the co-efficient of variation of the 50 runs. Given that a limit of 50,000 evaluations was imposed, this was not at first surprising and it was initially assumed that the troughs in the variance represented the fact that, over the 50 runs, the algorithm was simply hitting the limitation of 50,000 evaluations with increased frequency, thus reducing variation. However, detailed examination even of Figure 2 shows this not to be the case as where the mean can be seen to fall from around 30,000 at a mutation rate of 1.0e –4 to around 5,000 at a rate of 5.0e –4, no significant rise in variation occurs, suggesting that this simple explanation is invalid. Indeed Figures 4 and 6 (at 200,000 evaluations and 1,000,000 evaluations respectively) show this clearly to be the case.

Figure 3 shows plots of the 'total mutations used' and the mean fitness of 'best' solutions found over the 50 runs at each mutation rate. As hypothesised, mean fitness can be seen to increase in distinct steps as mutation rates are increased with plateaux occurring between mutation rates of 6.1e–5 through 2.3e–3, and 9.5e–3 through 3.1e–2. For mutation rates within these ranges, no significant improvement in mean fitness is seen, however the number of evaluations taken to find these solutions is seen to vary dramatically. The product of the mutation rate, number of evaluations used and chromosome length is also plotted and also shows a step like profile. This indicates that for certain ranges of rates of mutation, it is the number of mutations used that is important, and not necessarily the rate at which they are applied. This suggested the possibility that an unusual feature of the random number generator used in the program was being exhibited, and so the experiments were entirely replicated in a different programming language and using a different random number generator. This repeated set of experiments produced similar results with the same characteristics. It is therefore highly unlikely that 'random number generator' characteristics are the cause of this phenomenon and a more plausible explanation is discussed in Section 5.

It can also be seen from Figure 3, that the steps in each plot are subject to a form of 'phase shift' and do not occur at the same ranges of mutation rate. This will be explored in more detail later in this paper.

Figures 4 and 5 show similar performance profiles when trials are allowed up to 200,000 evaluations. However the mutation rates inducing the first two peaks in mean evaluations can now be seen to have fallen to 5.38e–6 and 3.28e–3 respectively, and the minor anomaly in Figure 2 is seen to expand to a distinct 'peak / trough' feature at around 5.24e–2. The mutation rate inducing the best overall fitness performance remains at 1.48e–1, however mean fitness can be seen to have improved significantly from around 310 to 350. Interestingly, the value of total number of mutations used defining the plateaux in Figures 3 and 5 can be seen to be similar.

Figures 6 and 7 show results for runs allowed up to 1 million evaluations. Superimposed onto these plots is a banded classification of repeated performance phases (A, B and C). In each of these, the performance of the algorithm can be characterised by differing effects on the mean number of evaluations used, process variation and mean fitness.

In the rightmost region denoted 'A', mean evaluations can be seen to rise, variation falls, total mutations used rises and average fitness rises. Here the algorithm is using increased mutation to explorer wider regions of the search space.

In band 'B', mean evaluations are seen to fall, process variation remains low (though slightly rising), and mean fitness remains constant, as does total mutations used. Here the algorithm is unable to use mutation to break out of certain local optima which can be found ever quicker as mutation rate increases. The left hand edge of band 'B' is therefore desirable as it gives solutions in lower numbers of evaluations with little deterioration in process variation.

In band 'C', mean evaluations starts to rise, process variation increases dramatically, total number of evaluations needed rises however there is little or no increase in mean fitness. This is a far from ideal set of circumstances, as the number of evaluations used varies widely for no improvement in mean fitness.

Crucially, as mutation rates are increased further, this pattern of phase 'A', 'B' and 'C' performance can be seen to repeated at least 3 times. Beyond this third iteration, insufficiently fine mutation rates have been sampled to show if further repetition exists; but this will be addressed in future work.

As before, features of the performance profile at 1 million evaluations occur at lower mutation rates than at either 200,000 and 50,000 evaluations. The 'third peak anomaly' previously occurring at mutation rates around 5.24e–2 is seen to be a distinct peak in Figure 6. The mutation rate inducing highest mean fitness is still around 1.48e–2 but now produces mean fitness of around 410 out of a maximum of 448, indicating a significant increase in the number of runs finding one of the 2 global optima. Once again, mutation rates beyond this value cause rapid deterioration into random search with poor results.

Further experiments are in progress, with different crossover operators (2 point, and Uniform [21]), and other selection strategies including higher tournament sizes, elitist, generational breeder [8] strategies and deterministic crowding [6]. These are all showing the features reported here, However it should be noted that these other algorithms display more marked differences at higher population sizes, as would be expected.

Finally, the experiment at 1 million evaluations was repeated on an instance of the real-world ADDMP problem. Figure 8 shows a multi-modal mean evaluation performance profile over the same range of mutation rates and population sizes as those used in Figure 1 on the H-IFF problem. It can also be seen that these features are persistent over a range of population sizes (here as high as 500), although with reducing amplitude. Particularly with the H-IFF result (Figure 1), this effect of increased population size should be expected, allowing crossover to play an increasingly important role in the search process, thus attenuating the effects of mutation.

Detailed examination of the coefficient of variation, total mutations used and mean fitness at each mutation rate also show strong similarities to those seen on H-IFF, but with less obvious structure and recurrence. This is perhaps not surprising since the ADDMP search space cannot be expected to possess such regularity as H-IFF. Further experiments are also underway with other standard test problems and results from these will be submitted for publication in due course.

5 DISCUSSION

Consideration of these results leads us to the following explanation, based on the fact that as mutation rate increases, emphasis shifts from a majority of single bit mutations per chromosome towards significant numbers of 2 bit mutations, and then 3 bit mutations, and so forth. At very low mutation rates, single bit mutation per chromosome occurs with increasing frequency, leading to increasing mean number of valuations but with decreasing variance (Phase A). As this problem has many local optima, it does not lend itself to single bit mutation hill climbers and thus the length of any useful random walk is severely limited. This walk will be achieved independently of the total number of mutations allowed and hence the plateaux in total mutations used being at the same level (around 170 mutations) in Figures 3, 5 and 7 (Phase B).

Once the usefulness of single bit mutations seems to have been exhausted, no further progress can be made until the expected number of 2 bit mutations becomes significant enough to make an impact on the search. This is of course dependent on the total number of evaluations allowed in the run, and will occur at lower mutation rates for higher evaluation limits. Indeed in all 3 cases, this transition is seen to commence at mutation rates which give an estimated number of around four 2 bit mutations by the end of the run.

As mutation is further increased, the introduction of 2 bit mutation can initially be expected to occur (if at all) significantly later on in the run than the number of evaluations needed to fully exploit single bit mutations. Hence this causes an increase first in the variance (Phase C), and then a significant increase in the mean number of evaluations used (Phase A) as variance falls again and the associated rise in mean fitness occurs. As mutation rates approach optimal for useful 2 bit mutation, the mean number of evaluations used drops (Phase B).

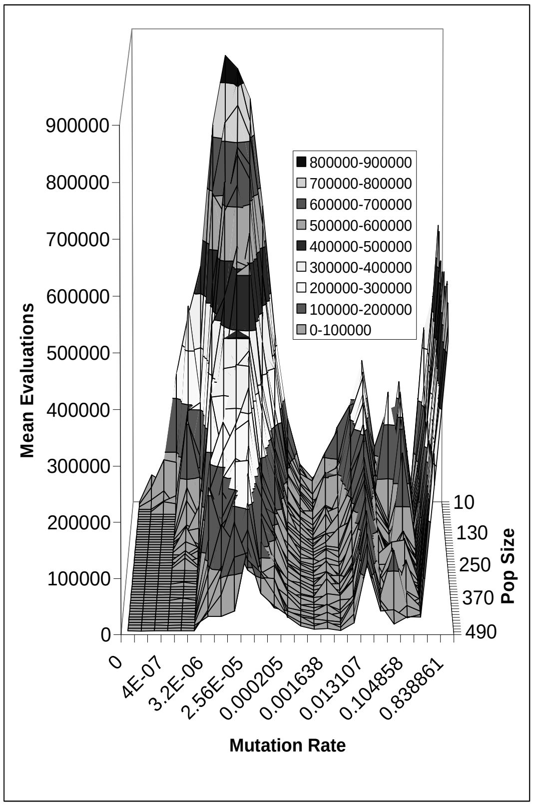

Figure 8 – ADDMP Performance Profile

As the usefulness of 2 bit mutation is exhausted, the cycle is again repeated as 3 bit mutations start to become significantly available. Again, at higher evaluation limits, this will occur at lower mutation rates. As the mutation rates favouring increased levels of bit mutation get closer together, it is likely that significant numbers of 4 bit mutations will arise before useful 3 bit mutations have been exhausted, hence these phases can be expected to merge as mutation rates increase.

6 CONCLUSIONS

The H-IFF problem has yielded a complex performance profile for simple evolutionary search strategies, particularly at low population sizes. The rate of mutation applied can be seen to dramatically affect several aspects of algorithm performance including mean evaluations until convergence, process variation; and mean fitness found.

Response to different mutation rates has been seen to fall into 3 distinct, repeating phases on this problem. There are ranges of mutation rates over which the algorithm can be seen to be predominantly exploring the search space, able to break free from particular local optima. There are ranges in which the algorithm is able to find solutions with particular qualities in fewer evaluations, without deterioration in process variation; and there are ranges in which performance shows sudden deterioration in process repeatability without significant increase in mean fitness.

These phases are seen to be repeated at different ranges of mutation rates, likely to be related to the ability of certain rates of mutation to be optimal for finding and breaking free of specific local optima in the problem search space.

Similar results have also been seen in other multimodal problems, and in particular in an example of a real-world industrial optimisation problem. As such, it can be seen that the investigation of performance profiles yields a complex collection of factors which should be taken into account in the context of parameter tuning. For example, in an industrial application, initial studies may reveal a locally optimal mutation rate which delivers an adequate level of fitness quickly, with exploration beyond that rate showing that, although better fitnesses are occasionally found, convergence time and variance grow towards unacceptable levels. As we can see, however, further exploration of the performance profile may then reveal a phase beyond this at which the better fitness is more reliably and quickly found. Also, the apparent suggestion of a three-phase nature to the search behaviour as mutation rate is increased could lead to ways of inferring the phase from online sampling of search behaviour, which could then be of use in adaptive strategies which attempt, for example, to maintain search within a phase B band.

Acknowledgements

The authors wish to thank British Telecommunications Plc for ongoing support for this research.

References

[1] T Bäck, *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, 1996

[2] K Deb and S Agrawal : *Understanding Interactions among Genetic Algorithm Parameters*. in Foundations of Genetic Algorithms 1998, Morgan Kaufmann.

[3] D Goldberg (1989), *Genetic Algorithms in Search Optimisation and Machine Learning*, Addison Wesley.

[4] J Holland, *Adaptation in Natural and Artificial Systems*, MIT press, Cambridge, MA, 1993

[5] Kauffman, S.A., *The Origins of Order: Self-Organization and Selection in Evolution*, Oxford University Press, 1993

[6] Mahfoud, S. W, *Niching methods for Genetic Algorithms*, University of Illinois PhD dissertation (document 95001), 1995.

[7] Z Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, 1996.

[8] H Mühlenbein and D Schlierkamp-Voosen (1994), *The Science of Breeding and its application to the Breeder Genetic Algorithm*, Evolutionary Computation 1, pp. 335-360.

[9] H Mühlenbein, *How genetic algorithms really work: I. Mutation and hillclimbing*, in R.Manner, B. Manderick (eds), Proc. of 2nd Int'l Conference on Parallel Problem Solving from Nature, Elsevier, pp 15-25.

[10] E van Nimwegen and J Crutchfield : *Optmizing Epochal Evolutionary Search: Population-Size Independent Theory*, in Computer Methods in Applied Mechanics and Engineering, special issue on Evolutionary and Genetic Algorithms in Computational Mechanics and Engineering, D Goldberg and K Deb, editors, 1998.

[11] E van Nimwegen and J Crutchfield : *Optmizing Epochal Evolutionary Search: Population-Size Dependent Theory*, Santa Fe Institute Working Paper 98-10-090, also submitted to Machine Learning, 1998.

[12] M Oates, D Corne and R Loader, *Investigating Evolutionary Approaches for Self-Adaption in Large Distributed Databases*, in Proceedings of the 1998 IEEE ICEC, pp. 452-457.

[13] M Oates and D Corne, *QoS based GA Parameter Selection for Autonomously Managed Distributed Information Systems*, in Procs of ECAI 98, the 1998 European Conference on Artificial Intelligence, pp. 670-674.

[14] M Oates and D Corne, *Investigating Evolutionary Approaches to Adaptive Database Management against various Quality of Service Metrics*, LNCS, Procs of 5th Intl Conf on Parallel Problem Solving from Nature, PPSN-V (1998), pp. 775-784.

[15] M Oates, *Autonomous Management of Distributed Information Systems using Evolutionary Computing Techniques*, Computing Anticipatory Systems, AIP Conf Procs 465, 1998, pp. 269-281.

[16] M Oates, D Corne and R Loader, *Skewed Crossover and the Dynamic Distributed Database Problem*, Artificial Neural Networks and Genetic Algorithms 1999, Dobnikar et al (eds), Springer pp 280-287.

[17] M Oates, D Corne and R Loader , *Investigation of a Characteristic Bimodal Convergence-time/Mutation-rate Feature in Evolutionary Search*, in Procs of Congress on Evolutionary Computation 99 Vol 3, IEEE, pp. 2175-2182

[18] Oates M, Corne D and Loader R, *Variation in Evolutionary Algorithm Performance Characteristics on the Adaptive Distributed Database Management Problem*, in Procs of Genetic and Evolutionary Computation Conference 99, Morgan Kaufmann, pp.480-487

[19] M. Oates, J. Smedley, D. Corne, R. Loader, *Bimodal Performance Profile of Evolutionary Search and the Effects of Crossover*, in Procs of 1999 Evonet Summer School on Theoretical aspects of Evolutionary Computation (in press).

[20] Oates M, Corne D and Loader R, *Multimodal Performance Profiles on the Adaptive Distributed Database Management Problem*, to appear in Procs of the EVOTEL 2000, the Second European Workshop on Evolutionary Computation in Telecommunications.

[21] G Syswerda (1989), *Uniform Crossover in Genetic Algorithms*, in Schaffer J. (ed), Procs of the Third Int. Conf. on Genetic Algorithms. Morgan Kaufmann, pp. 2 – 9

[22] Watson RA, Hornby GS, and Pollack JB, *Modelling Building-Block Interdependency*, LNCS, Procs of 5th Intl Conf on Parallel Problem Solving from Nature, PPSN-V (1998), pp. 97-106.

[23] Watson RA, Pollack JB, *Hierarchically Consistent Test Problems for Genetic Algorithms*, in Procs of Congress on Evolutionary Computation 99 Vol 2, IEEE, pp. 1406-1413

[24] Deb, K., Horne, J. and Goldberg, D.E., Multimodal Deceptive Functions, *Complex Systems* **7**, 131–153.