Approaches to Selection and their Effect on Fitness Modelling in an Estimation of Distribution Algorithm

Alexander E. I. Brownlee, John A. W. McCall, Qingfu Zhang and Deryck F. Brown

Abstract—**Selection is one of the defining characteristics of an evolutionary algorithm, yet inherent in the selection process is the loss of some information from a population. Poor solutions may provide information about how to bias the search toward good solutions. Many Estimation of Distribution Algorithms (EDAs) use truncation selection which discards all solutions below a certain fitness, thus losing this information. Our previous work on Distribution Estimation using Markov networks (DEUM) has described an EDA which constructs a model of the fitness function; a unique feature of this approach is that because selective pressure is built into the model itself selection becomes optional. This paper outlines a series of experiments which make use of this property to examine the effects of selection on the population. We look at the impact of selecting only highly fit solutions, only poor solutions, selecting a mixture of highly fit and poor solutions, and abandoning selection altogether. We show that in some circumstances, particularly where some information about the problem is already known, selection of the fittest only is suboptimal.**

I. Introduction

Selection is one of the defining characteristics of an evolutionary algorithm, yet inherent in the selection process is the loss of some information from a population. This is particularly the case in truncation selection, commonly used in Estimation of Distribution Algorithms, which is the focus of the work presented in this paper.

Selection operators have previously been developed to deliberately increase the chance of including some low fitness solutions to increase diversity and avoid premature convergence; tournament selection being a classic example. Further approaches to combating diversity loss in the context of EDAs have included mutation of probabilistic distributions reported in [1]-[3]. Recent work on combating diversity loss due to sampling and selection in an EDA was also presented in [4]. The results we report here compliment that work by revealing some of the circumstances under which diversity loss caused by selection impacts on the fitness modelling capability of the algorithm. We present variations of truncation selection which can be used to allow greater diversity within a population.

Further related work was presented in [5]. That paper looked at an algorithm using a Gaussian distribution and suggested that using individuals discarded by selection in estimating the distribution would result in a closer fit to the fitness landscape. [6] and [7] present an approach which uses information from the whole population in the selection step. [8] looks at the influence of selection on finding dependencies between variables (structure or linkage learning) in an EDA, also important in modelling the fitness function.

The EDA *Distribution Estimation Using Markov networks (DEUM)* [9]-[11] constructs a model of a fitness function and samples this model to locate an optimum. It incorporates fitness into the model and because of this does not require a selection operator to guide the search. However, a traditional selection operator may be used in combination with it to bias the fitness model produced. In this paper we extend our previous work by employing this property to analyze the effects of selection on the information about the fitness function contained in a population. DEUM constructs a best fit to the population it is given; this uses the information available to construct a model of the fitness function. This model can be compared to the true fitness function using fitness prediction; we can then infer from this comparison how much information about fitness was present in the population. It is not always essential to construct an accurate model of the fitness function to optimize; often coefficients will be required in the model to reflect small changes in fitness that have no effect on the ranking of individuals by fitness. This is similar to the idea of benign interactions [12] and may be related to the concept of unnecessary interactions [13]. Spurious correlations [14], [15] are also related to this, but are false relationships in the model resulting from selection rather than interactions present in the fitness function but not required for optimisation.

A number of other algorithms [16]-[18] have been developed which also employ fitness modelling, typically as a surrogate fitness function to reduce the number of fitness evaluations.

Some recent work indicates that simple algorithms which ignore high levels of complexity perform reasonably well on complex fitness functions, particularly in combination with other simple algorithms such as hill climbers [19] and genetic algorithms [20]. However, it is still useful to explore in depth when selection may or may not be beneficial and what role it plays in the evolutionary process.

In our previous works on DEUM [9]-[11], [21] we empirically determined when the use or absence of selection would produce better results. Here we look at this issue systematically. Thus in this paper we have two aims: to

Manuscript received December 14, 2007.

Alexander Brownlee (corresponding author), John McCall and Deryck Brown are with the School of Computing, The Robert Gordon University, St Andrew St, Aberdeen, AB25 1HG, UK (phone: +44 (0)1224 262472; fax: +44 (0)1224 262727; e-mail: {sb,jm,db}@comp.rgu.ac.uk).

Qingfu Zhang is with the Department of Computer Science, The University of Essex, Colchester, UK (e-mail: qzhang@essex.ac.uk).

examine the effects of selection on our algorithm, and from this to infer clues as to its wider effect on evolutionary algorithms.

This paper is structured in the following order. In Section II we explain in further detail how we use a Markov network to construct a model of a fitness function, and how the quality of this model can be measured using the fitness prediction technique. In Section III we outline the structure of our experiments. In Sections IV and V we report and analyze the results of our experiments with selection on fitness models with perfect and imperfect structures. Finally in Section VI we draw our conclusions.

II. Fitness Modelling using Markov Networks

A. Construction of the Fitness Model

DEUM uses a Markov network (also known as a Markov Random Field) to model fitness as an energy distribution over the solution space. A Markov network is an undirected graphical model, in contrast to the directed graphical models such as Bayesian networks used by many EDAs such as hBOA [22] and EBNA [23]-[24]. Markov networks are also used in MN-FDA [25]; this algorithm also incorporates a structure learning step whereas DEUM currently does not.

A Markov network models a set of random variables as nodes on a graph, and interactions between those variables as edges. It is characterised by a property known as Markovianity, which states that the distribution of any node can be completely defined by the values of its neighbouring nodes. The Markov network may be viewed as a set of cliques, a clique simply being any fully connected subgraph of the graphical model. This allows a joint probability distribution for the Markov network to be defined in terms of the Gibbs distribution:

$$p(x)=\frac{f(x)}{\sum_y f(y)}\equiv\frac{e^{-U(x)/T}}{\sum_y e^{-U(y)/T}} \quad (1)$$

U(x) is a sum of *clique potential functions*, each of which models the neighbourhood relationship between variables in one particular clique on the graph. The summations are over all possible solutions *y*. *T* is a temperature coefficient, which remains set to 1 in all of our current experiments.

As with many evolutionary algorithms DEUM models a set of individuals. Each individual $x=\{x_1,x_2,...,x_n\}$ is a particular set of values which can be applied to the set of variables $X=\{X_1,X_2,...,X_n\}$ in a problem. Each individual is assigned a fitness to denote the quality of solution it represents.

In [26] it was shown that an equation for each individual in a population may be derived from the joint probability distribution shown in (1). This relates solution fitness to an energy function calculated from the values taken by variables in a set of individuals:

$$-\ln(f(x))=U(x) \quad (2)$$

Here, *f(x)* is the fitness of an individual *x* and *U(x)* is the energy function derived from alleles. *U(x)* fully specifies the joint probability distribution, so can be regarded as a probabilistic model of the fitness function. Minimising *U(x)* is equivalent to maximising *f(x)*. In this context we call the model the *Markov Fitness Model (MFM)*.

In early versions of DEUM, the Markov network used a univariate structure, and there was only one clique for each variable. In [10] and [11] we introduced extra cliques for bivariate and then trivariate interactions. Theoretically this can be extended to cover all multivariate interactions, although in practice this will be limited by available space and processing power as the number of terms in the model will grow exponentially with the order of interaction.

This gives us an energy function for each individual which can be expressed as:

$$U(x) = \begin{array}{l} c+\sum_{i=0}^{n}\alpha_i x_i+\sum_{i=0}^{n}\sum_{j=0}^{n}\alpha_{ij}x_i x_j \\ +\sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{k=0}^{n}\alpha_{ijk}x_i x_j x_k \ldots \end{array} \quad (3)$$

where each α is a parameter associated with a clique on the Markov network, c is a constant representing the *zero-clique* of background energy in the Markov network, *n* is the number of variables in each individual and x_i represents the value of variable *i* in the solution *x*. (terms for up to trivariate interactions are shown) Here {-1,1} are used as the values of x_i in place of {0,1} to ensure arithmetical symmetry between values. The set of α values completely model the distribution.

An example is helpful to illustrate this. A simple MAXSAT problem has the set of predicates in (4).

$$(x_1\vee\bar{x}_2\vee x_3)\wedge(x_2\vee x_3)\wedge(\bar{x}_4\vee x_2) \quad (4)$$

The negations may be ignored when considering the relationships between predicate variables giving us the undirected graphical structure shown in Fig. 1.

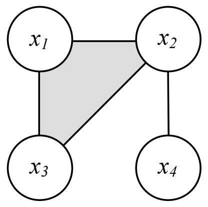

Fig. 1. Relationships between variables. x_1, x_2 and x_3 share a three way interaction (shaded); two way interactions are shown by lines

In the general energy function (5) for this problem, we have a constant, a term for each of the predicate variables x_i, a term for the bivariate interactions shown as edges on the graph and a term for the trivariate interaction shown by the shaded area.

$$U(x)=\begin{array}{l} c+\alpha_1x_1+\alpha_2x_2+\alpha_3x_3+\alpha_4x_4+\alpha_{12}x_1x_2 \\ +\alpha_{13}x_1x_3+\alpha_{23}x_2x_3+\alpha_{24}x_2x_4+\alpha_{123}x_1x_2x_3 \end{array} \quad (5)$$

An individual $x=\{0011\}$, with fitness $f(x)=2$ would thus have the energy function shown in (6).

$$-\ln(2)=c-\alpha_1-\alpha_2+\alpha_3+\alpha_4+\alpha_{12}-\alpha_{13}-\alpha_{23}-\alpha_{24}+\alpha_{123} \quad (6)$$

To determine the Markov network parameters, a random population is formed in the normal manner for an evolutionary algorithm. The energy function for each individual is formed resulting in a set of equations relating α values, energy (derived from fitness) and alleles. Singular value decomposition (SVD) [27] is used to solve the system of simultaneous equations and determine the unknown α values. Either the entire population or a selected subset can be used in this process – selective pressure comes from energy minimisation in the model rather than traditional selection operators. The model can then be sampled to generate a new population for the next generation; however for the experiments outlined in this paper, we will simply measure its fitness modelling capacity.

B. *Fitness Prediction*

We have seen that DEUM models the fitness function directly. In addition to sampling model to find an optimum we can also use it to predict the fitness of individuals; we can use this ability to measure how closely the MFM is modelling the fitness function.

Predicting the fitness of an individual is a simple task given a previously constructed model. The bitstring is encoded as before so that for each x_i, 0 is coded as -1 and 1 remains unchanged. These values are substituted into the energy function (3) to give a predicted energy $U(x)$ for the individual. The predicted fitness can then be calculated thus:

$$f(x) = e^{-U(x)} \quad (7)$$

We can then compare the predicted fitness values of a population with the true fitness values using the product moment correlation coefficient [28], which is calculated according to:

$$r = \frac{\sum(x-\bar{x}).(y-\bar{y})}{\sqrt{\sum(x-\bar{x})^2.\sum(y-\bar{y})^2}} \quad (8)$$

Where x is the set of true fitnesses, $\bar{x}$ is the mean of this set, y is the set of predicted fitnesses and $\bar{y}$ is the mean of this set. Here, we use this to measure the quality of fitness models being constructed. We have also performed some experiments using Spearman's rank correlation coefficient [28] – results using this measure show similar trends to those using product moment correlation and for brevity are not reported here.

This procedure is as follows:

1. Generate random initial population
2. Select a subset σ of the population
3. Use σ to build MFM
4. For each member of s, repeat:
   4.1. Mutate one bit in the individual
   4.2. Use MFM to predict fitness of individual
   4.3. Use fitness function to determine true fitness
5. Calculate the product moment correlation coefficient between the predicted and true fitnesses
6. Generate a complete new random population equal in size to the first, and for each member of this population:
   6.1. Use MFM to predict fitnesses
   6.2. Use fitness function to determine true fitness
7. Calculate correlation between the predicted and true fitnesses

We call the correlation figures obtained the fitness prediction correlation (FPC). We refer to the correlation obtained in step 5 as C_m and that in step 7 as C_r.

It is important to explain the relevance of the two different correlation measures. C_r measures the ability of the model to predict the fitness of randomly generated individuals; it follows that if this is a strong correlation then the MFM is closely modelling the fitness function. C_m is a useful figure in the wider context of evolutionary algorithms. Typically this will be reasonably good – even in situations where the model structure is missing key interactions – it represents the ability of the MFM to predict fitness of solutions "near to" the current population. This is important in the context of the proximate optimality principal [29] which assumes that good solutions have similar structure. Based on this idea, EAs will generally move to a population which closely resembles at least part of the current one. Related work on fitness of neighbouring solutions is described in [30].

C. *What makes a good fitness model?*

A correlation coefficient of 0.7 is typically considered to represent a strong positive correlation between two sets of values [31]. In our research using Markov networks to model fitness we have found a strong link between the population size used to build the MFM and its FPC. A typical example (is this case applied to a 400 bit onemax problem) is shown in Fig. 2. This example is from a run where the whole population was used to build the model (that is, no selection). Similar behaviour is exhibited if selection is present.

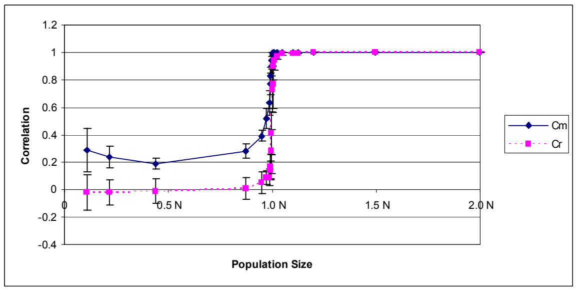

Fig. 2. Increase in fitness prediction correlation with increasing population size

Here we see how both FPC values vary with increasing population size from 0 up to 2N, where N is the number of terms in the MFM. Where the population size is less than N, we describe the system as *underspecified*; where it is greater we describe it as *overspecified*. In this paper, when we describe the system as over specified we used a population size on 1.1N; when underspecified we give the population size as a proportion of N.

D. Selection

The selection operator being examined here is truncation selection, commonly used in EDAs. It selects the fittest p * n solutions from the population, where p is a value between 0 and 1 and n is the population size. In comparison with other selection operators, truncation selection selects a solution only once (unless it is duplicated in the population), and there is zero probability of poor fitness individuals being selected.

Here we use two variants of truncation selection: bottom selection and top & bottom selection. We call the standard truncation selection top selection. Bottom selection selects the least fit p * n and top & bottom selection selects the fittest (p/2) * n and least fit (p/2) * n. Indirectly, we have a fourth operator; when the number of selected solutions equals the population size, we effectively have no selection.

III. Experimental Method

The experiments all follow the same procedure:

1. For each selection operator, and each selection proportion:
   1.1. Generate random initial population
   1.2. Select a subset σ of the population
   1.3. Use σ to build MFM
   1.4. Calculate C_m and C_r

Considering the discussion on specified and underspecified models in the Section II, it is important that the number of individuals used to build a model remains constant. This is achieved by choosing the proportion of the population that will be selected, then generating a population of suitable size to result in the chosen number being selected. This allows us to observe the effects of selection on the model building without matters being confused by the under/over-specification issue. In all experiments, the proportions p of the population selected were 0.01, 0.02, 0.05, 0.1, 0.2, 0.28, 0.5, 0.67, 0.83 and 1 (that is, no selection). Each experiment was repeated 30 times and the mean and standard deviation for each FPC calculated. Each experiment presented here looks at a specific instance of a fitness function which reflects the general pattern seen over the range of that particular function.

IV. Perfect Model Structures

A. Outline

The first series of experiments use what we describe as a perfect model structure. In this case, the MFM includes all interactions defined by the problem. This does not include all possible interactions but does include those which influence the absolute fitness value but are not required by the model to locate an optimum. By using a perfect model structure it is possible to set the model parameters to completely match the fitness function. This allows us to examine the effects of selection separate to the effect of missing or superfluous interactions in the model. We will look at the more realistic situation of imperfect model structure in section V.

The interactions included in the model structure are defined completely by the problem being examined. This is only possible for predefined test problems where the interactions are explicit. In the case of onemax there are no interactions. For the Ising problem, an interaction exists wherever there is a coupling between two spin variables. In the case of 3-CNF MAXSAT, each clause of three variables results in the addition of a 3-way interaction, and three pairwise interactions.

We will look at both fully specified and underspecified models. That is, where the number of individuals used to build the model is greater than and less than N, the number of terms in the MFM.

B. Results Using Fully Specified Models

Firstly we look at a classic and extensively studied univariate problem, onemax. There are no interactions between variables and the MFM includes only univariate terms, so the model has a total of N=101 parameters including the constant. The results are shown in Fig. 3.

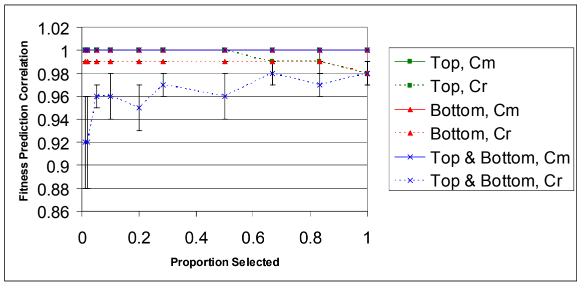

Fig. 3. Results for a 100 bit onemax problem, selecting 1.1N solutions (overspecified).

We can see that the MFM is able to model the fitness function extremely well (note the values on the y-axis); C_r is always higher than 0.8 and comes close to 1. Top selection yields the best model quality, which falls off as the proportion of the population selected increases. Bottom selection shows a similar trend, starting with a slightly lower C_r. In contrast, top & bottom selection results in the poorest model which improves as a larger proportion of the population is included. C_m is consistently close to 1 in all cases.

Fig. 4 shows the results for a 2D Ising Spin Glass problem, as described in [32]. Here problem instances have interactions between pairs of variables, giving a model size of N=301. We previously ran experiments optimizing this problem in [10].

We see the same trends appearing again; top selection gives high values for both C_r and C_m. These begin to drop as the proportion of solutions selected approaches 1. Bottom selection gives slightly lower values but follows the same gradual decrease. Top & bottom selection gives the lowest values with a gradual increase as more of the population is selected.

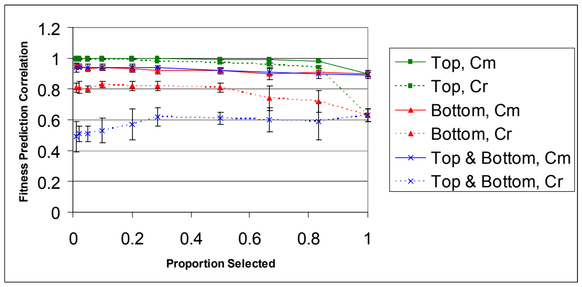

Fig. 4. Results for a 100 bit Ising Spin Glass problem, including all interactions in model and selecting 1.1N solutions (overspecified).

We now move on to a 3-CNF MAXSAT problem, presented in Fig. 5. Here we look at 100 bit instances from the phase transition region of the problem. The instances were obtained from SATLIB [33] and were used in the optimization experiments we described in [11]. These have interactions between up to three variables (we call these trivariate interactions) which are completely defined by the clauses in the problem specification. N varies because of differing numbers of bivariate interactions in each instance – across instances used in the experiment N=1667 with a standard deviation of 10.

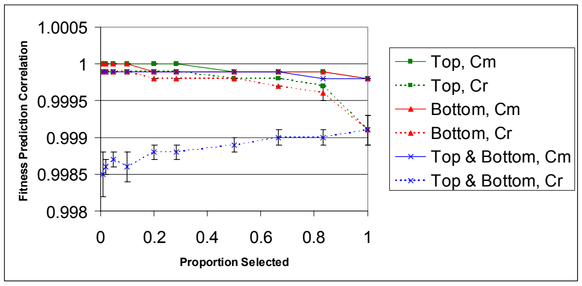

Fig. 5. Results for a 100 bit MAXSAT problem, including all interactions in model and selecting 1.1N solutions (overspecified).

We see very similar results to those for the onemax problem, though with all correlation values considerably closer to 1. Again top selection gives the best C_r values, while bottom selection follows a similar trend, slightly lower. Top & bottom selection gives the lowest values for C_r. Also like onemax, C_m values are consistently close to 1 for all selection types.

C. Results Using Underspecified Models

As illustrated in the Section II, an underspecified MFM does not generally model the fitness function well. However, some of our earlier experiments with DEUM have produced good results with what we have now determined to be too small a population. One possibility here is that selection can be used to focus the model on important parts of a population – reinforcing the limited information in a small population.

Again, the first experiment in this section looks at a 100 bit onemax problem (N=101), shown in Fig. 6. Here, the number of solutions selected equals 0.1N; considerably underspecified.

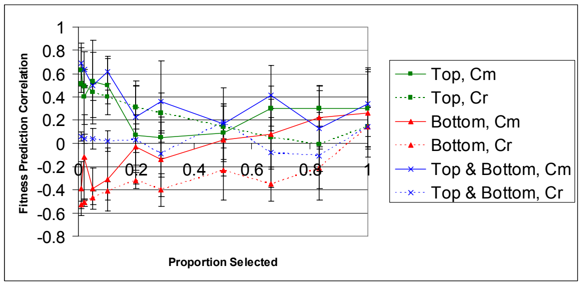

Fig. 6. Results for a 100 bit onemax problem, selecting 0.1N solutions (underspecified).

One point of particular note is that when using top selection the fitness prediction capability for random individuals at some points appears better than that for mutated individuals similar to those in the selected set, though set against a background of large standard deviations. We believe than the explanation for this lies in the "best-fit" behaviour of SVD, used to construct the MFM, when supplied with such a heavily underspecified system. Looking at the C_r values separately, we see that C_r appears to be highest when using top selection, top & bottom selection gives a correlation of approximately 0 between true and predicted fitnesses, and bottom selection shows an inverse correlation with true fitness. This is perhaps a result in keeping with intuition.

Perhaps most interesting in this context is that when selecting a small proportion of the population, top & bottom selection produces a model with a higher fitness modelling capability than top selection and bottom selection. Given the large standard deviations this is hard to interpret however. Similar results were obtained when the selected size was 0.5N.

In Fig. 7 we move on to the Ising problem (N=201). Again the number of solutions selected equals 0.1N.

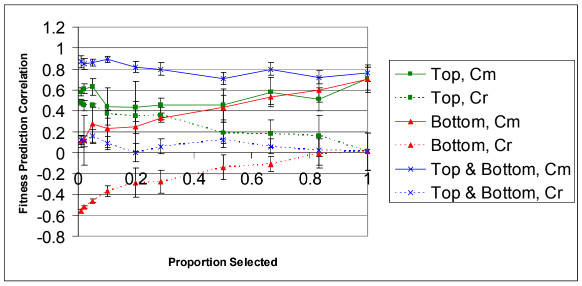

Fig. 7. Results for a 100 bit Ising Spin Glass problem, including all interactions in the model and selecting 0.1N solutions (underspecified).

Here we see a similar trend to that seen on onemax, although with less noise. For C_r, top selection gives a better model that the other operators. Results for C_m give a strong indication that top & bottom selection is producing a better model of fitness than top selection, which falls off slightly as the proportion selected increases. Top selection and bottom selection show similar results – a lower C_m which increases as a larger proportion of the population is selected.

In Fig. 8 we now show the results for MAXSAT (N=1667, SD 10) with a selected size of 0.1N.

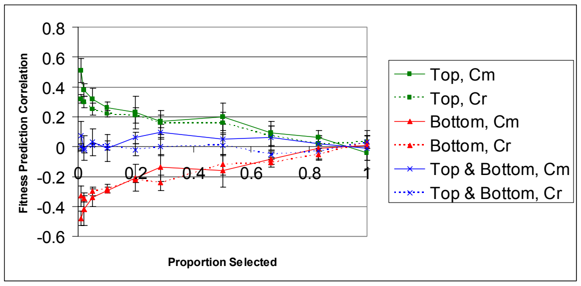

Fig. 8. Results for a 100 bit MAXSAT problem, including all interactions in the model and selecting 0.1N solutions (underspecified).

Here we can see selection exerting a strong influence on the quality fitness model produced. Top selection gives good values for both C_r and C_m, whereas top & bottom selection and bottom selection give close to 0 and negative correlations respectively. These results are perhaps closest to what we would intuitively expect the operators to produce for all problems. The strong influence of selection here is likely to be a reflection of the problem's complexity – with so little information in the problem, emphasis of strong solutions is required to build a good model of fitness.

D. Summary

In this section we have seen that with a perfect model structure, and a selected population large enough to give a fully specified system, use of the traditional top selection operator makes only a small difference to the fitness model achieved. Indeed, almost as much information about the fitness function was found in the lower part of the population as in the upper part; and a good model was obtained without any selection at all. It is possible that there is a trade-off – with the full population there is too much information and over fitting occurs; this being reduced by discarding a small number of solutions (selection of a high proportion). It is also worth observing the strong correlations seen between predicted and true fitness for random individuals – particularly for random individuals – showing the strength of the MFM approach to modelling fitness.

In the context of an underspecified MFM, we see a different picture. Top selection is important in building the general fitness model – C_r values only approach strong positive correlation when using this operator. However, the model is able to predict fitness of neighbouring solutions better when using top & bottom selection – we believe this is because this operator sharpens the information already present in the population, without completely discarding either poor or fit solutions.

V. Imperfect Model Structures

A. Outline

We now move on to experiments using imperfect model structures. Here the MFM does not include all the terms required to perfectly model the fitness function. Such structures are important because they are comparable with structures built by learning algorithms such as independence tests, which will inevitably miss some interactions. As the model is unable to perfectly match the fitness function it is not necessarily the case that sampling it to find an optimum will yield the global optimum of the fitness function. For optimization this means that we would no longer use the approach of building the model once then sampling it to find the optimum as in our previous work [10], [11], [21]. Instead, as is common with EDAs we can use the ability to model neighbouring populations to push the population gradually towards the optimum. In this case, selection is more likely to play a bigger role.

B. Experimental Results

In these experiments we cannot, of course, investigate onemax because it has no variable interactions to omit. Thus we begin by looking at the Ising problem, shown in Fig. 9.

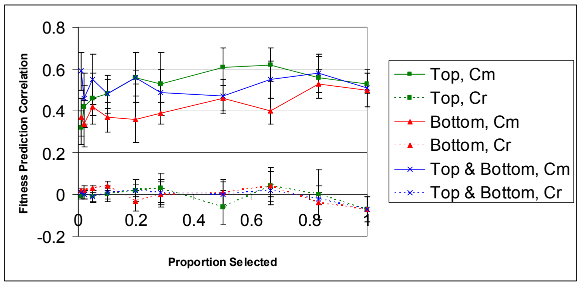

Fig. 9. Results for a 100 bit Ising Spin Glass problem, omitting all interactions from the model and selecting 1.1N solutions

These results are produced by the univariate model applied to the Ising problem, resulting in N=101. Firstly we can see that completely ignoring interactions between variables results in a poor model compared to those in Section IV – C_r values regardless of selection type are all close to 0 indicating that the model is unable to predict fitness of random solutions. However, for all three selection operators we have a relatively high C_m value, although in all cases it is below the 0.7 which as discussed in II-C would be considered a strong correlation. This indicates that modelling of nearby solutions is good but prone to error. The three operators give similar results, with top selection giving marginally higher C_m values.

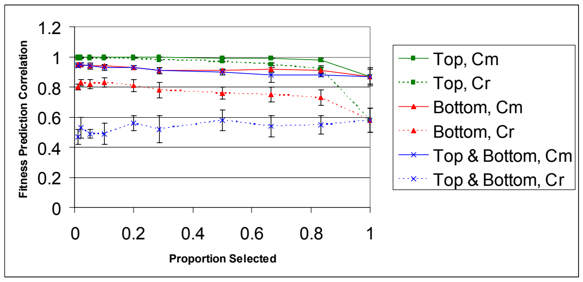

Fig. 10. Results for a 100 bit Ising Spin Glass problem, with a random 10% of interactions removed from the model and selecting 1.1N solutions (overspecified).

In Fig. 10 we show results for a decimated model applied to the Ising problem. In this case, 10% of the interactions have been removed from the model structure at random, giving N=381. Here, top selection gives both the highest C_m and C_r values, falling away as the proportion of the population selected increases. Bottom selection follows the same trend, at a lower level. Top & bottom selection is comparable to bottom selection for C_m, yet much poorer for C_r – increasing as the proportion selected increases.

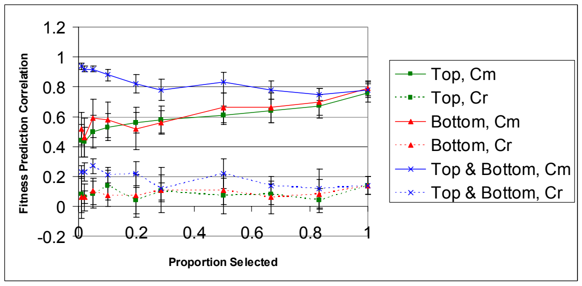

Fig. 11. Results for a 100 bit MAXSAT problem, omitting all interactions from the model and selecting 1.1N solutions (overspecified).

In Fig. 11 we see a univariate model applied to the MAXSAT problem, giving N=101. With a large number of high order interactions missing from the model, as one might expect the C_r value is poor for all forms of selection. However, more interesting is the result for mutated individuals. A similar amount of information is gained using either top selection or bottom selection; however the best results are found when using top & bottom selection, particularly with a small proportion of the population.

In Fig. 12 we see the same problem with all univariate and trivariate interactions included in the model (that is, only bivariate interactions omitted – resulting in N=431 for all instances). Again we see that top & bottom selection yields a more accurate fitness model of neighbouring solutions than the other two methods. We have also repeated this experiment with only trivariate interactions removed and on a decimated model with of a random 10% of interactions removed. Both give similar results.

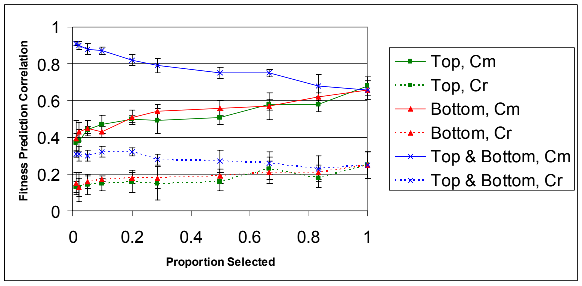

Fig. 12. Results for a 100 bit MAXSAT problem, omitting all bivariate interactions from the model and selecting 1.1N solutions (overspecified).

C. Summary

As with underspecified systems with perfect model structures, when we use an imperfect structure with a fully specified system we also see an improvement in model quality by using some form of selection. For the Ising problem, top selection gives the best model; for MAXSAT, top & bottom selection gives the best results. As might be expected, the models are unable to predict fitness of random individuals; however with appropriate choice of selection operator the model shows a strong correlation between predicted and true fitness values of solutions near the existing population. This is important in guiding the search gradually toward an optimum.

For underspecified systems this effect was also observed to a limited extent. In the wider context of EAs we believe that this compliments the other work [5]-[7] which shows that selection operators which include information from the wider population can yield better performance. Indeed, these results would indicate that there is considerable information about fitness in the poorer parts of a population which could be beneficial to reproduction operators (whether they be probabilistic models or more traditional operators). They also show that selection plays an important role in gathering this information from the population.

VI. Conclusions

In this paper we have described a series of experiments investigating the effect of different approaches to selection on the ability of an EDA to model fitness. This is further to the use of selective pressure in balance with modelling to achieve efficient optimisation. We have seen that while the traditional truncation selection often results in a good fitness model, comparable fitness models can also be produced when selecting different parts of the population (or indeed all of it).

From the results presented here, it is clear that selection is particularly important when the fitness model is unlikely to perfectly match the fitness function – as is the case when variable interactions have been learned from data rather than given as part of the problem. In such situations selection improves the ability of the algorithm to accurately model neighbouring areas of the search space – aiding the movement towards a global optimum. This is likely to be of use in optimising Estimation of Distribution Algorithms. It shows that a well-tuned computationally cheap selection operator can be used to improve the performance of a simpler model resulting in a more efficient algorithm relative to one using a more complex or perfect model.

We have also seen that in some circumstances the pure “top selection” operator can be outperformed by a “top & bottom selection” which selects and equal number of high and poor fitness solutions. This is in line with other work on selection – from the introduction of operators like tournament selection to more recent work on EDAs. It reinforces the idea that a considerable improvement in performance may be obtained by correct choice of selection operator.

It will be interesting to extend this work to cover a much larger range of problems, to determine more precisely when the different approaches to selection should be used. It will also be useful to look at the effects of other selection operators such as fitness proportionate selection and tournament selection.

REFERENCES

[1] A. Ochoa and M.R. Soto, “Linking Entropy to Estimation of Distribution Algorithms,” in J.A. Lozano, P. Larrañaga, Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms. Springer-Verlag, 2006, pp.1-38.

[2] H. Handa, “Estimation of Distribution Algorithms with Mutation,” in Evolutionary Computation in Combinatorial Optimization. Springer, 2005 , pp. 112-121.

[3] T. Mahnig and H. Mühlenbein, “Optimal Mutation Rate using Bayesian Priors for Estimation of Distribution Algorithms,” in Proceedings of the First Symposium on Stochastic Algorithms: Foundations and Applications (SAGA-2001). Springer, 2001, pp.33-48.

[4] J. Branke, C. Lode and J.L. Shapiro, "Addressing sampling errors and diversity loss in UMDA," in GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation, pp. 508-515, 2007.

[5] P. Pošík and V. Franc, "Estimation of Fitness Landscape Contours in EAs," in GECCO 2007 - Proceedings of the 9th annual conference on Genetic and Evolutionary Computation, pp. 562-569, 2007.

[6] T. Miquélez, E. Bengoetzea and P. Larrañaga, “Evolutionary Computation based on Bayesian Classifiers,” in the International Journal of Applied Mathematics and Computer Science, 2004 vol 14 issue 3, pp. 101-115.

[7] T. Miquélez, E. Bengoetzea, A. Mendiburu and P. Larrañaga, “Combining Bayesian Classifiers and Estimation of Distribution Algorithms for Optimization in Continuous Domains,” in Connection Science, 2007, vol. 19 issue 4, pp. 297-319.

[8] C.F. Lima, M. Pelikan, D.E. Goldberg, F.G. Lobo, K. Sastry and M. Hauschild, “Influence of Selection and Replacement Strategies on Linkage Learning in BOA,” in Proceedings of the 2007 Congress on Evolutionary Computation, IEEE Press, 2007, pp. 1083-1090.

[9] S.K. Shakya, "DEUM: A framework for an Estimation of Distribution Algorithm based on Markov Random Fields," PhD Thesis, The Robert Gordon University, Aberdeen, UK, 2006.

[10] S.K. Shakya, J.A.W. McCall and D.F. Brown, "Solving the Ising Spin Glass Problem using a bivariate EDA based on Markov Random Fields," in Proceedings of IEEE Congress on Evolutionary Computation (IEEE CEC 2006), 2006.

[11] A. Brownlee, J. McCall and D. Brown, "Solving the MAXSAT Problem using a Multivariate EDA based on Markov Networks," A late breaking paper in GECCO '07: Proceedings of the 2007 conference on Genetic and Evolutionary Computation, 2007.

[12] L. Kallel, B. Naudts and R. Reeves, “Properties of Fitness Functions and Search Landscapes,” in L. Kallel, B. Naudts and A. Rogers, “Theoretical Aspects of Evolutionary Computing,” Springer-Verlag, 2000, pp. 177-208.

[13] M. Hauschild, M. Pelikan, C.F. Lima and K. Sastry, "Analyzing probabilistic models in hierarchical BOA on traps and spin glasses," in GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation, pp. 523-530, 2007.

[14] H. Mühlenbein and T. Mahnig, “Evolutionary Optimization using Graphical Models,” in New Generation Computing, 18, pp.157-166, 2000

[15] R. Santana, P. Larrañaga, J.A. Lozano (2007) Challenges and open problems in discrete EDAs. Technical Report. EHU-KZAA-IK-1/07. University of the Basque Country.

[16] C.F. Lima, M. Pelikan, K. Sastry, M. Butz, D.E. Goldberg and F.G. Lobo, “Substructural Neighbourhoods for Local Search in the Bayesian Optimization Algorithm,” in Proceedings of the Parallel Problem Solving from Nature Conference (PPSN-IX), Springer, 2006, pp. 232-241.

[17] K. Sastry, C. Lima and D.E. Goldberg, “Evaluation Relaxation using Substructural Information and Linear Estimation,” in Proceedings of the 8th annual Conference on Genetic and Evolutionary Computation (GECCO-2006), ACM Press, 2006, pp. 419-426.

[18] A. Orriols-Puig, K. Sastry, D.E. Goldberg, and E. Bernad´o-Mansilla, “Substructrual Surrogates for Learning Decomposable Classification Problems: Implementation and First Results,” IlliGAL Report No. 2007010, Illinois Genetic Algorithms Laboratory, The University of Illinois at Urbana-Champaign, March 2007.

[19] Q. Zhang, J. Sun and E. Tsang, "Combinations of Estimation of Distribution Algorithms and Other Techniques," International Journal of Automation & Computing, pp. 273-280, July, 2007.

[20] Q. Zhang, J. Sun and E. Tsang, "An evolutionary algorithm with guided mutation for the maximum clique problem." IEEE Trans.Evolutionary Computation, vol. 9, pp. 192-200, 2005.

[21] S.K. Shakya, J.A.W. McCall and D.F. Brown, "Incorporating a Metropolis method in a Distribution Estimation using Markov Random Field Algorithm," in Proceedings of IEEE Congress on Evolutionary Computation (IEEE CEC 2005), pp. 2576-2583, 2005.

[22] M. Pelikan, “Hierarchical Bayesian Optimization Algorithms,” Springer Verlag, 2005.

[23] R. Exteberria and P. Larrañaga, “Global Optimization with Bayesian Networks,” in II Symposium on Artificial Intelligence, CIMAF99, Special Session on Distributions and Evolutionary Optimization, pp. 332-339.

[24] P. Larrañaga, R. Exteberria, J.A. Lozano and J.M. Peña, “Combinatorial Optimization by Learning and Simulation of Bayesian and Gaussian Networks,” Technical Report KZZA-IK-4-99, Department of Computer Science and Artificial Intelligence, The University of the Basque Country, 1999.

[25] R. Santana, “Estimation of Distribution Algorithms with Kikuchi Approximations,” in Evolutionary Computation, vol. 13, issue 1, 2005, pp. 67-97.

[26] D.F. Brown, A.B. Garmendia-Doval and J.A.W. McCall, "Markov Random Field Modelling of Royal Road Genetic Algorithms," in Selected Papers from the 5th European Conference on Artificial Evolution, pp. 65-76, 2002.

[27] W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, Numerical Recipes: The Art of Scientific Computing, Cambridge, UK: Cambridge University Press, 1986.

[28] T. Lucey, Quantatitive Techniques: An Instructional Manual, Eastleigh, Hampshire, UK: D. P. Publications, 1984.

[29] F. Glover and F. Laguna, Tabu Search, Norwell, MA, USA: Kluwer Academic Publishers, 1997.

[30] P. Collard, S. Verel and M. Clergue, "Local search heuristics: Fitness Cloud versus Fitness Landscape," Poster at the 2004 European Conference on Artificial Intelligence (ECAI04), pp. 973-974, 2004.

[31] D. Rowntree, Statistics without tears : a primer for non-mathematicians, Harmondsworth, UK: Penguin, 1981, pp. 199.

[32] R. Kindermann and J.L. Snell, Markov Random Fields and their applications, Providence, RI: American Mathematical Society, 1980.

[33] H.H. Hoos and T. Stützle, "SATLIB: An Online Resource for Research on SAT," in SAT 2000, I.P. Gent, H.V. Maaren and T. Walsh, IOS Press, 2000, pp. 283-292.