From Mating Pool Distributions to Model Overfitting

Claudio F. Lima
DEEI-FCT
University of Algarve
Campus de Gambelas
8005-139, Faro, Portugal
clima.research@gmail.com

Fernando G. Lobo
DEEI-FCT
University of Algarve
Campus de Gambelas
8005-139, Faro, Portugal
fernando.lobo@gmail.com

Martin Pelikan
MEDAL
University of Missouri at St. Louis
One University Blvd.
St. Louis, MO 63121
pelikan@cs.umsl.edu

ABSTRACT

This paper addresses selection as a source of overfitting in Bayesian estimation of distribution algorithms (EDAs). The purpose of the paper is twofold. First, it shows how the selection operator can lead to model overfitting in the Bayesian optimization algorithm (BOA). Second, the metric score that guides the search for an adequate model structure is modified to take into account the non-uniform distribution of the mating pool generated by tournament selection.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

General Terms

Algorithms, Performance, Theory

Keywords

Estimation of distribution algorithms, Bayesian optimization algorithm, Bayesian networks, overfitting, selection.

1. INTRODUCTION

Estimation of distribution algorithms (EDAs) [11, 15] can be classified according to the complexity of their probabilistic models. Simpler EDAs use a model of simple and fixed structure and only learn the corresponding parameters. At the other side of the spectrum, are the Bayesian EDAs which use Bayesian networks (BNs) [14] to model complex multivariate interactions. While Bayesian EDAs are able to solve a broad class of nearly decomposable and hierarchical problems in a reliable and scalable manner, their probabilistic models oftentimes do not exactly reflect the problem structure. Because these models are learned from a sample of limited size (population of individuals), particular features of the specific sample are also encoded, which act as noise when seeking for generalization. This is a well-known problem in machine learning, known as model overfitting.

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.

In many situations, the knowledge of the problem structure can be as valuable as a high-quality solution to the problem. This is the case for several model-based efficiency enhancement techniques developed for EDAs that yield *super-multiplicative speedups* [9]. Another important situation is the *offline interpretation* of the probabilistic models to help develop fixed but structure-based operators for specific instances or classes of problems that have similar structure.

This paper investigates the influence of the selection procedure on model quality for the Bayesian optimization algorithm (BOA) [16, 15]. Selection is analyzed as the mating pool distribution generator, which turns out to have a great impact on Bayesian network learning. Particularly, it is shown that tournament selection generates the mating pool according to a power distribution that leads to model overfitting. However, if the metric that scores networks takes into account the resampling performed by tournament, the model quality can be highly improved and comparable to that of truncation selection which generates a uniform distribution, more suitable for BN learning.

The next section introduces relevant background to understand the purpose of the paper. Section 3 analyzes selection as the mating pool generator, while Section 4 models the metric gain when overfitting with tournament selection. Finally, a correction to the complexity penalty is proposed to avoid overfitting, and conclusions are presented.

2. PRELIMINARIES

2.1 Bayesian Optimization Algorithm

The Bayesian optimization algorithm (BOA) [16, 15] uses Bayesian networks (BNs) [14] to capture the (in)dependencies between the decision variables of the optimization problem. In BOA, the traditional crossover and mutation operators of genetic algorithms are replaced by (1) building a BN which model promising solutions and (2) sampling from the corresponding probability distribution to generate new solutions.

A BN is defined by its structure and corresponding parameters. The structure is represented by a directed acyclic graph where the nodes correspond to the variables of the problem and the edges correspond to conditional dependencies. The parameters are represented by the conditional probabilities for each variable given any instance of the variables that this variable depends on. More formally, a Bayesian network encodes the following joint probability distribution,

$$p(X) = \prod_{i=1}^{\ell} p(X_i|\Pi_i), \quad (1)$$

where $X = (X_1, X_2, \ldots, X_\ell)$ is a vector with all variables of the problem, Π_i is the set of *parents* of X_i (nodes from which there exists an edge to X_i), and $p(X_i|\Pi_i)$ is the conditional probability of X_i given its parents Π_i.

The parameters of a Bayesian network can be represented by a set of conditional probability tables (CPTs) or local structures. Using local structures such as decision trees allows a more efficient and flexible representation of local conditional distributions, improving the expressiveness of BNs. In this work we focus on BNs with decision trees.

The quality of a given network structure is quantified by a scoring metric. Here, we consider two popular metrics for BNs: the K2 metric [5, 10] and the Bayesian information criterion (BIC) [18]. It has been shown that the behavior of these metrics is asymptotically equivalent; however, the results obtained with each metric can differ for particular domains, particularly in terms of sensitivity to noise. In the context of EDAs, when using CPTs to store the parameters, the BIC metric outperforms the K2 metric, but when using decision trees or graphs, the K2 metric has shown to be more robust [15]. We will confirm this observation later in Section 2.3.

To learn the most adequate structure for the BN a greedy algorithm is usually used for a good compromise between search efficiency and model quality. We consider a simple learning algorithm that starts with an empty network and at each step performs the operation that improves the metric the most, until no further improvement is possible. The operator considered is the *split*, which splits a leaf on some variable and creates two new children on the leaf. Each time a split on X_j takes place at tree T_i, an edge from X_j to X_i is added to the network. For more details on BNs with local structures the reader is referred elsewhere [4, 7, 15].

2.2 Structural Accuracy of Probabilistic Models in Bayesian EDAs

Definition 1. The *model structural accuracy* (MSA) is defined as the ratio of correct edges over the total number of edges in the Bayesian network.

Definition 2. An *edge* is *correct* if it connects two variables that are linked according to the objective function definition.

Definition 3. Model overfitting is defined as the inclusion of *incorrect (or unnecessary) edges* to the Bayesian network.

To investigate the MSA in BOA, we focus on solving a problem of known structure, where it is clear which dependencies must be discovered (for successful tractability) and which dependencies are unnecessary (reducing the interpretability of the models).

The test problem considered is the $m - k$ trap function, where m is the number of concatenated k-bit trap functions. Trap functions [1, 6] are relevant to test problem design because they bound an important class of nearly decomposable problems [9]. The trap function used [6] is defined as follows

$$f_{trap}(u) = \begin{cases} k, & \text{if } u = k \\ k - 1 - u, & \text{otherwise} \end{cases} \quad (2)$$

where u is the number of ones in the string, k is the size of the trap function. Note that for $k \geq 3$ the trap function is fully deceptive [6] which means that any lower than k-order statistics will mislead the search away from the optimum. In this problem the accurate identification and exchange of the building-blocks (BBs) is critical to achieve success, because processing substructures of lower order leads to exponential scalability [19]. Note that no information about the problem is given to the algorithm; therefore, it is equally difficult for BOA if the variables correlated are closely or randomly distributed along the chromosome string. A trap function with size $k = 5$ is used in our experiments.

To focus on the influence of selection in model quality, the replacement strategy is kept as simple as possible, where the offspring fully replace the parent population.

For all experiments, we use the minimal population size required to solve the problem in 10 out of 10 independent runs. The population size is obtained by performing 10 independent bisection runs [17]. Therefore, the total number of function evaluations is averaged over 100 (10×10) runs.

2.3 Influence of Selection Strategy on MSA

The influence of the selection strategy in BOA has been discussed before [12]. Here, we review essential findings to the purpose of studying model overfitting and extend the experiments to the BIC metric. In particular, we consider two widely used selection schemes in EDAs: Tournament and truncation selection.

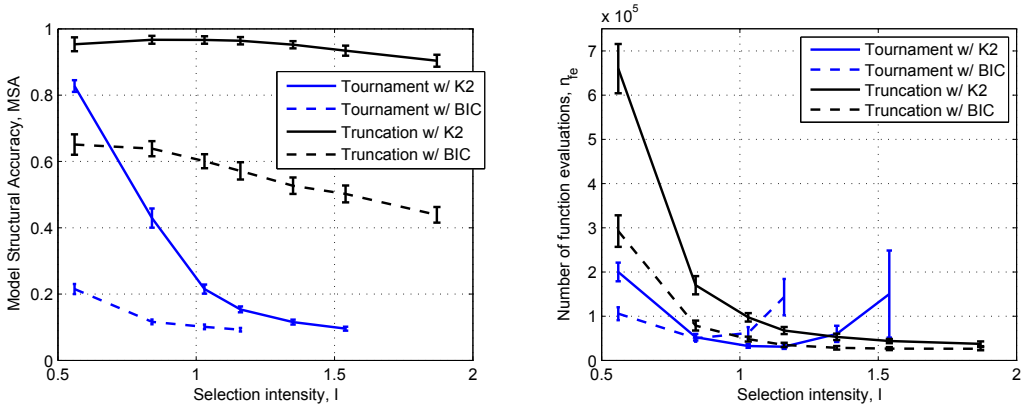
In tournament selection [8, 3], s individuals are randomly picked from the population and the best one is selected for the mating pool. This process is repeated n times, where n is the population size. There are two popular variations of tournament selection, with and without replacement. With replacement, the individuals are drawn from the population following a discrete uniform distribution. Without replacement, individuals are also drawn randomly from the population but there's the guarantee that every individual participates in exactly s tournaments. While the expected outcome for both alternatives is the same, the latter is a less noisy process. Therefore, in this study we use tournament selection without replacement.

In truncation selection [13] the best τ% individuals in the population are selected for the mating pool. This method is equivalent to the standard (μ, λ)-selection procedure used in evolution strategies (ESs), where $\tau = \frac{\mu}{\lambda} \times 100$.

Note that when increasing the size of the tournament s, or decreasing the threshold τ, the selection intensity is increased, which means an increase in the selection strength. In order to compare the two selection operators on a fair basis, different configurations for both methods with equivalent selection intensity are considered. The relation between selection intensity I, tournament size s, and truncation threshold τ is taken from [2] and is shown in Table 1.

Table 1: Equivalent tournament size (s) and truncation threshold (τ) for the same selection intensity (I).

I	0.56	0.84	1.03	1.16	1.35	1.54	1.87
s	2	3	4	5	7	10	20
τ(%)	66	47	36	30	22	15	8



(a) Model structural accuracy (b) Num. of function evaluations

Figure 1: Model structural accuracy and number of function evaluations for different selection-metric combinations when solving the 5-bit trap problem of size $\ell = 50$.

Figure 1 shows the model quality and number of function evaluations for different combinations of selection methods and scoring metrics. From a model quality perspective, it is clear that (1) truncation selection performs better than tournament selection and (2) K2 metric performs better than BIC metric. Note that with tournament selection, while for small values of s the number of evaluations decreases, after some value of s, the number of evaluations starts to increase again. Curiously, this happens when the MSA approaches 0.1.

3. SELECTION AS THE MATING POOL DISTRIBUTION GENERATOR

Like in traditional genetics, the selection mechanism is responsible for ensuring the survival of the fittest in the population. In the context of EDAs, this is one of the most important components inherited from the evolutionary computation framework. However, in EDAs, which have a strong connection with data mining and classification, the selection operator can also be viewed as the generator of the data set used to learn the probabilistic model at each generation. Since in EDAs we are interested in modeling the set of promising solutions, the selection operator indicates which individuals have relevant features to be modeled and propagated in the solution set (population of individuals). Before moving to the study of the selection strategy as the data set generator for learning the BNs, we make a simple analysis of the selection operators considered.

In terms of creating duplicate individuals in the population there are two responsible mechanisms. The selection operator explicitly assigns several copies of the same individual to the mating pool, where the number of copies is somewhat proportional to their fitness rank. This is the case for tournament, ranking, and proportional selection. Additionally, the model sampling procedure generates with a certain probability duplicates of the same individual, although selection implicitly controls how often this happens. Note that this probability will increase in time as the EDA starts focusing on more concrete regions of the search space. Clearly, the selection operator has some influence on this phenomenon as it explicitly regulates the convergence speed of the algorithm. Without loss of generality, consider that the replication of individuals done explicitly by the selection operator is the main source of duplicates in the population.

For the sake of simplicity, let us assume that all individuals have different fitness. Ordering the population by fitness, where the worst individual has rank 1 and the best has rank n, the probability that an individual with rank i wins a given tournament of size s is, for $i \geq s$, given by

$$p_i = \frac{\binom{i-1}{s-1}}{\binom{n-1}{s-1}} = \frac{(i-1)!(n-s)!}{(i-s)!(n-1)!} = \prod_{j=1}^{s-1} \frac{i-j}{n-j}, \qquad \text{for } s \geq 2. \tag{3}$$

Note that the worst $s-1$ individuals will never win a tournament, therefore for $i < s$, $p_i = 0$.

Given that in tournament selection without replacement each individual participates in exactly s tournaments, the expected number of copies (c_i) in the mating pool for an individual of rank i is simply

$$c_i = s\, p_i. \tag{4}$$

For $i \gg s$, and consequently $n \gg s$, the distribution of the expected number of copies c_i can be approximated by a power distribution with p.d.f.,

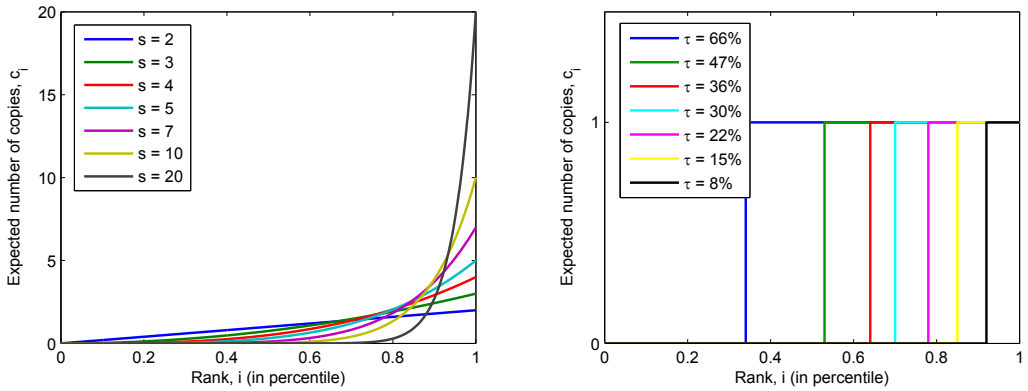
$$f(x) = \alpha\, x^{\alpha-1}, \quad 0 < x < 1, \alpha = s. \tag{5}$$

In this way, the distribution of c_i can be expressed for any population size, where the relative rank is given by $x = i/n$. Note that as the relative rank slightly decreases from 1 the corresponding number of expected copies rapidly decreases. This is particularly true for higher tournament sizes, when increasing the exponent of the power factor.

On the other hand, in truncation selection the expected number of copies for the selected individuals is one, which follows a uniform distribution with p.d.f.,

$$c_i = \begin{cases} 0, & \text{if } i < n\,(1 - (\tau/100)) \\ 1, & \text{otherwise.} \end{cases} \tag{6}$$

Figure 2 shows the distributions of expected number of copies for each individual with rank expressed in percentile.



(a) Tournament (b) Truncation

Figure 2: Distribution of the expected number of copies in the mating pool for (a) tournament and (b) truncation selection with different selection intensity values. Note that s and τ values generate the same selection intensity. Rank is expressed in percentile.

The difference between the two selection methods is notorious. While tournament selection assigns increasing relevance to top-ranked individuals according to a power distribution, truncation selection gives no particular preference to any of the selected individuals, all having the same frequency in the learning data set.

The differences between tournament and truncation distributions stress out two relevant features of any given selection method: (1) *window size*, which determines the proportion of unique individuals that are included in the mating pool, and (2) *distribution shape*, which determines the relevance of each selected individual in the mating pool, in terms of the number of copies. These features in a certain way control the tradeoff between exploration and exploitation in model structural learning in EDAs.

Clearly, tournament and truncation selection differ in both features. While the window size is deterministically defined in truncation selection—solutions above the threshold are included in the selected set and solutions below are not—in tournament selection, the choice of which individuals to include in the mating pool is a stochastic process (except for the best solution and the worst $s-1$), but also guided by fitness rank. The probability of inclusion rapidly decreases with rank, particularly for larger tournament sizes, as can be seen in Figure 2 (a). In terms of distribution shape, the two selection methods also differ significantly. Tournament selection gives higher emphasis to top-ranked solutions according to a power distribution with $\alpha = s$. This means that best solutions get approximately s copies in the mating pool, which forces the learned models to focus on particular features of these individuals, which contain good substructures, but also undesirable components due to stochastic noise.

Another way to look at tournament selection in comparison with truncation selection as the mating pool generator is recognizing that this selection procedure acts as a biased data resampling on an uniform data set. The uniform data set is the set of unique selected solutions (solutions that will win at least one tournament), similar to what happens in truncation, while the resampling is performed when top-ranked individuals participate in more than one tournament. This sort of resampling is clearly biased by fitness.

4. MODELING METRIC GAIN WHEN OVERFITTING

To analyze the effect of tournament size on resampling bias we must look at the cumulative distribution function (c.d.f.) of the power distribution, which is given by

$$F(x) = x^s, \quad 0 < x < 1, s \geq 1. \tag{7}$$

Note that for $s = 1$ we have a uniform distribution and there's no resampling, as the mating pool becomes a complete copy of the population. For $s \geq 2$, we can obtain the proportion of individuals in the mating pool with rank equal or less than x by simply calculating $F(x)$, or alternatively, the market share of the $(1-x)$ top-ranked individuals given by $1 - F(x)$ (right-side area of the c.d.f.).

The overfitting due to noise coming from top-ranked individuals is certainly more likely to happen if we think in a fairly small percentage of the population. Said differently, the smaller this proportion is, the more likely these individuals will contain the same misleading features that are induced by noise. On the other hand, this proportion should be significant enough in terms of relative frequencies so that it can influence the metric component that scores the likelihood of the model with respect to the data. How large or small should this proportion be depends obviously on the tournament size. For larger tournament sizes, this proportion is expected to be inferior to the case of smaller tournament sizes because the number of copies assigned to top individuals increases considerably. Therefore, we recognize that this proportion should be small, but the exact proportion will differ from situation to situation.

To better ilustraste our argument, Figure 3 shows the power c.d.f. for several proportions of top-ranked individuals. It can be seen that for small proportions ($\leq 1\%$) of top-ranked individuals, the expected proportion in the mating pool after selection grows approximately linearly with the tournament size. Note that as the proportion considered is more elitist, the slope of the linear relationship approaches the proportion itself. For example, when considering the best 0.1%, the market share after selection with $s = 50$ is $4.88\% \approx 0.1\% \times 50$.

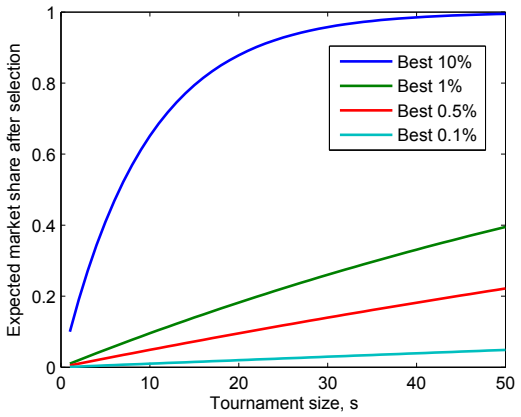


Figure 3: Expected market share of top-ranked individuals included in the mating pool after selection for infinite population size. For small proportions ($\leq$ 1%) of top-ranked individuals the relation between tournament size and the expected proportion in the mating pool after selection is approximately linear.

The bottom line of this rationale is to verify that, in the worst case, the noise in terms of counts or relative frequencies coming from the replication of top-ranked individuals grows linearly with the tournament size.

Consider now the possibility of adding an edge from a variable X_2 to another variable X_1 due to nonlinearities introduced by tournament selection, knowing that these two variables are in fact independent from each other. To investigate the influence of the resampling done by successive tournaments, we must derive the score metric for the network where an edge is added from X_2 to X_1. Given that both MDL and Bayesian metrics are decomposable, it is sufficient to look at the term corresponding to the node X_1. The metric gain obtained by splitting a leaf on X_2 in the tree encoding the parameters of X_1 and adding the corresponding edge to the network is given by

$$\begin{aligned} G_{metric} = \quad & ScoreAfter - ScoreBefore \\ & - ComplexityPenalty, \end{aligned} \tag{8}$$

where *ScoreAfter* is the metric score obtained after splitting the leaf into two new ones, *ScoreBefore* is the score obtained before the split (keeping X_1 independent from X_2), and *ComplexityPenalty* is the penalty associated with the increased complexity of adding one leaf to the tree. In BOA, if this gain is positive the split is accepted and the corresponding edge is inserted in the Bayesian network.

Considering the BIC metric, the metric gain corresponding to adding an edge from X_2 to X_1 is

$$\begin{aligned} G_{BIC} = \quad & m(X_1X_2 = 00) \log_2 \left(\frac{m(X_1X_2 = 00)}{m(X_2 = 0)} \right) \\ & + m(X_1X_2 = 10) \log_2 \left(\frac{m(X_1X_2 = 10)}{m(X_2 = 0)} \right) \\ & + m(X_1X_2 = 01) \log_2 \left(\frac{m(X_1X_2 = 01)}{m(X_2 = 1)} \right) \\ & + m(X_1X_2 = 11) \log_2 \left(\frac{m(X_1X_2 = 11)}{m(X_2 = 1)} \right) \\ & - m(X_1 = 0) \log_2 \left(\frac{m(X_1 = 0)}{n} \right) \\ & - m(X_1 = 1) \log_2 \left(\frac{m(X_1 = 1)}{n} \right) \\ & - \frac{1}{2} \log_2 (n) , \end{aligned} \tag{9}$$

where $m(X_1X_2 = x_1x_2)$ is the number of individuals in the population with $X_1X_2 = x_1x_2$ and n is the population size. Note that the first four terms correspond to *ScoreAfter*, the fifth and sixth terms express *ScoreBefore*, and the final term penalizes the score because of the complexity added to the BN. Denoting $m(X_1X_2 = x_1x_2)$ by $m_{x_1x_2}$ and recognizing that $m(X_1 = x_1) = m(X_1X_2 = x_10) + m(X_1X_2 = x_11)$, as well as $n = m_{00} + m_{01} + m_{10} + m_{11}$, the previous expression can be expressed as

$$\begin{aligned} G_{BIC} = \quad & m_{00} \log_2 \left(\frac{m_{00}}{m_{00} + m_{10}} \right) + m_{10} \log_2 \left(\frac{m_{10}}{m_{00} + m_{10}} \right) \\ & + m_{01} \log_2 \left(\frac{m_{01}}{m_{01} + m_{11}} \right) + m_{11} \log_2 \left(\frac{m_{11}}{m_{01} + m_{11}} \right) \\ & - (m_{00} + m_{01}) \log_2 \left(\frac{m_{00} + m_{01}}{m_{00} + m_{01} + m_{10} + m_{11}} \right) \\ & - (m_{10} + m_{11}) \log_2 \left(\frac{m_{10} + m_{11}}{m_{00} + m_{01} + m_{10} + m_{11}} \right) \\ & - \frac{1}{2} \log_2 (m_{00} + m_{01} + m_{10} + m_{11}) . \end{aligned} \tag{10}$$

Expressing in terms of relative frequencies, the gain can be expressed as

$$G_{BIC} = n \; G'_{BIC} - \frac{1}{2} \log_2(n), \tag{11}$$

where

$$\begin{aligned} G'_{BIC} = \quad & p_{00} \log_2 \left(\frac{p_{00}}{p_{00} + p_{10}} \right) + p_{10} \log_2 \left(\frac{p_{10}}{p_{00} + p_{10}} \right) \\ & + p_{01} \log_2 \left(\frac{p_{01}}{p_{01} + p_{11}} \right) + p_{11} \log_2 \left(\frac{p_{11}}{p_{01} + p_{11}} \right) \\ & - (p_{00} + p_{01}) \log_2 (p_{00} + p_{01}) \\ & - (p_{10} + p_{11}) \log_2 (p_{10} + p_{11}) . \end{aligned} \tag{12}$$

Next, we want to model the deviation from the actual frequencies in a uniformly distributed mating pool (in terms of copies) to biased frequencies towards the noise induced by the replication of top-ranked individuals (power distribution). First, consider the frequencies on the uniform mating pool to be $p_{00} = p_{01} = p_{10} = p_{11} = 0.25$, which reveals independence between X_1 and X_2. Then, and without loss of generality, we will assume that the these frequencies are deviated towards equally increasing p_{00}, p_{11} and equally decreasing p_{01}, p_{10}. This assumption relies on the fact that the decrease in entropy (corresponding to an increase in score) will be achieved faster than for other possible configurations of pairwise frequency deviation. In this way, we analyze the case that can upper bound other possible deviations.

Assuming that the deviation of the "true" frequencies is linear with respect to the tournament size, as argued before, the frequency deviation can be expressed as

$$\begin{aligned} p_{00} &\approx 0.25 + \Delta(s-1), \\ p_{01} &\approx 0.25 - \Delta(s-1), \\ p_{10} &\approx 0.25 - \Delta(s-1), \\ p_{11} &\approx 0.25 + \Delta(s-1), \end{aligned} \tag{13}$$

where Δ is the slope of the linear relationship plotted in Figure 3, therefore the exact value will depend on the proportion considered. Replacing (13) into (12) and denoting $(s-1)$ by s',

$$\begin{aligned} G'_{BIC} \approx \quad & (0.25 + \Delta s') \log_2 \left(\frac{0.25 + \Delta s'}{0.5} \right) \\ & + (0.25 - \Delta s') \log_2 \left(\frac{0.25 - \Delta s'}{0.5} \right) \\ & + (0.25 - \Delta s') \log_2 \left(\frac{0.25 - \Delta s'}{0.5} \right) \\ & + (0.25 + \Delta s') \log_2 \left(\frac{0.25 + \Delta s'}{0.5} \right) \\ & - 0.5 \log_2 (0.5) - 0.5 \log_2 (0.5) \,. \end{aligned} \tag{14}$$

Simplifying the previous equation, we have

$$\begin{aligned} G'_{BIC} \approx \quad & (0.5 + 2\Delta s') \log_2 \left(\frac{0.25 + \Delta s'}{0.5} \right) \\ & + (0.5 - 2\Delta s') \log_2 \left(\frac{0.25 - \Delta s'}{0.5} \right) + 1. \end{aligned} \tag{15}$$

Using the logarithm property $\log(a/b) = \log(a) - \log(b)$ and simplifying again, we get

$$\begin{aligned} G'_{BIC} \approx \quad & (0.5 + 2\Delta s') \log_2 (0.25 + \Delta s') \\ & + (0.5 - 2\Delta s') \log_2 (0.25 - \Delta s') + 2. \end{aligned} \tag{16}$$

Dividing both terms by 2,

$$\begin{aligned} \frac{1}{2} G'_{BIC} \approx \quad & (0.25 + \Delta s') \log_2 (0.25 + \Delta s') \\ & + (0.25 - \Delta s') \log_2 (0.25 - \Delta s') + 1. \end{aligned} \tag{17}$$

Looking at the function $x \log_2(x)$ for the interval $[0, 0.5]$, one can see that the first term in Equation 17 is relatively constant around -0.5. Therefore,

$$\frac{1}{2} G'_{BIC} \approx (0.25 - \Delta s') \log_2 (0.25 - \Delta s') + 0.5, \tag{18}$$

or alternatively,

$$G'_{BIC} \approx 2 \left(0.25 - \Delta s'\right) \log_2 \left(0.25 - \Delta s'\right) + 1. \tag{19}$$

The approximate expression for the metric gain G'_{BIC} due to overfitting of top-ranked individuals in tournament selection is plotted in Figure 4. A value of $\Delta = 0.001$ is used (best 0.1%). Since the schema proportions considered will vary from 0.25 to 0 or 0.5, the Δ value will basically define the increment/decrement step of that same proportions. For example, for a higher $\Delta = 0.005$ the approximate expression would be defined only for $s = [1, 50]$, instead of the plotted $s = [1, 250]$.

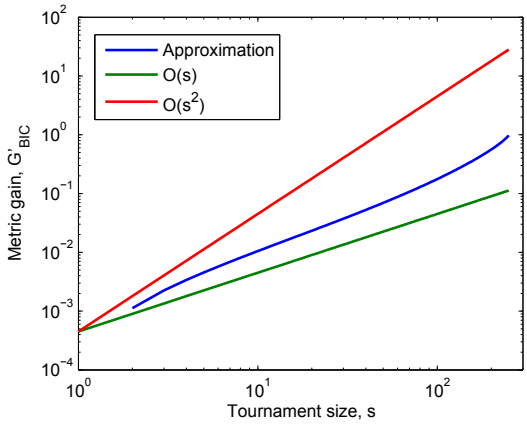


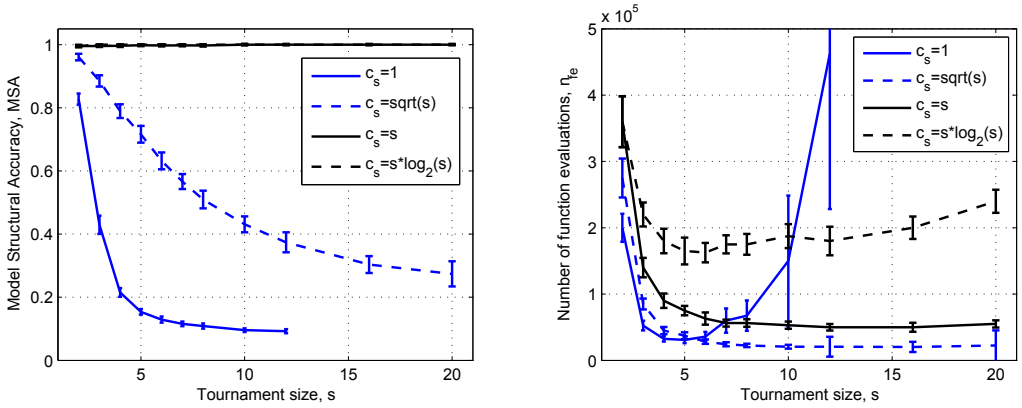
Figure 4: Approximated metric gain G'_{BIC} due to overfitting of top-ranked individuals in tournament selection. A value of $\Delta = 0.001$ is used (best 0.1%). The metric growth is somewhere between linear and quadratic, but closer to linear.

As can be seen, the metric gain grows close to linear in log-log scale, with the exception made for lower and higher values of s. This means a polynomial growth in linear scale, somewhere between linear and quadratic, which can be confirmed by comparison with reference curves. While the metric gain G'_{BIC} does not account for the factor n (population size) and the complexity penalty term $0.5 \log_2(n)$, it does tell us about the way the gain grows with respect to the tournament size s.

5. ADAPTING COMPLEXITY PENALTY

In this section we change the complexity penalty of the metric score in order to account for the power distribution nature of tournament selection. While the metrics considered have different backgrounds, the penalty associated with each leaf addition is exactly the same: $0.5 \log_2(n)$ [15]. This becomes clear if we compare the logarithm of the K2 metric with the BIC metric. Here, we aggravate this penalty by a factor that depends on the tournament size, using $0.5 c_s \log_2(n)$, where c_s is tournament size dependent. In this way, the greater the number of copies of top-ranked individuals in the mating pool, the more demanding we are in accepting an edge/leaf addition. From the previous section we know that the metric gain due to overfitting grows approximately as s, therefore we try different c_s values around s to investigate the corresponding response in terms of MSA and the number of function evaluations. We perform experiments for $c_s = \sqrt{s}, s, s \log_2(s)$ and compare them with the original penalty correction ($c_s = 1$).

Experiments for both BIC and K2 metrics were performed; however, due to lack of space we only show the results for the K2 metric. Figure 5 shows the model quality and corresponding evaluations for BOA with tournament selection using different complexity penalties. Already for $c_s = \sqrt{s}$, the model quality improves with respect to the standard case $c_s = 1$, but when considering $c_s = s$ and



(a) Model structural accuracy (b) Num. of function evaluations

Figure 5: Model quality and number of function evaluations for different penalty correction values $c_s = 1, \sqrt{s}, s, s\log_2(s)$ with the K2 metric.

$s\log_2(s)$ the improvement is much better. Increasing the penalty by a factor of s or higher takes model quality very close to 100%. However looking at the number of evaluations spent by each penalty, it is clear that $c_s = s\log_2(s)$ is too strong as a penalty because for larger s values it takes too many evaluations and the situation gets worse with increasing s. On the other hand, the $s-$penalty ($c_s = s$) shows to be an adequate penalty because while obtaining high-quality models the number of evaluations is kept constant after some tournament size. This point us out to another advantage of the $s-$penalty, because it allows to have a wider range of s values for which BOA performs well and at a relatively low cost. Similar results are obtained for the BIC metric, where the $s-$penalty is also the most adequate.

We now look at the behavior of tournament selection with the $s-$penalty for different problem sizes and compare it to truncation selection with the standard penalty. Figures 6 and 7 show BOA with tournament and truncation selection, respectively. Clearly, tournament selection with the $s-$penalty obtains better model quality than truncation selection with the standard penalty. Notice, however, that model quality is now plotted between 90% and 100%, because both methods obtain models of much better quality than tournament selection with the standard penalty. In terms of number of evaluations, tournament selection is still less expensive than truncation selection, but as selection intensity increases their costs become comparable. These results demonstrate that tournament selection is a good selection method for EDAs, like it is for GAs, as long as the scoring metric counterbalances the power distribution in the mating pool. The greater the tournament size is, the more demanding the metric score has to be in accepting edge/leaf additions.

6. CONCLUSIONS

This paper has addressed model overfitting in the context of the Bayesian optimization algorithm. The influence of selection methods in Bayesian network learning has been demonstrated by looking at the corresponding distributions in the set of selected solutions. The metric gain obtained when overfitting has been derived so that the complexity penalty of the scoring metric could be compensated in the same order of magnitude. By doing so, the model quality in BOA when using tournament selection has been considerably improved. While we did not consider other selection operators such as ranking or proportionate selection, the methodology developed in the paper should provide guidelines to account for the non-uniform distributions generated by these operators.

7. ACKNOWLEDGMENTS

This work was sponsored by the Portuguese Foundation for Science and Technology under grants SFRH-BD-16980-2004 and PTDC-EIA-67776-2006, the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant FA9550-06-1-0096, and the National Science Foundation under NSF CAREER grant ECS-0547013. The work was also supported by the High Performance Computing Collaboratory sponsored by Information Technology Services, the Research Award and the Research Board at the University of Missouri in St. Louis. The U.S. Government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright notation thereon.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research, the National Science Foundation, or the U.S. Government.

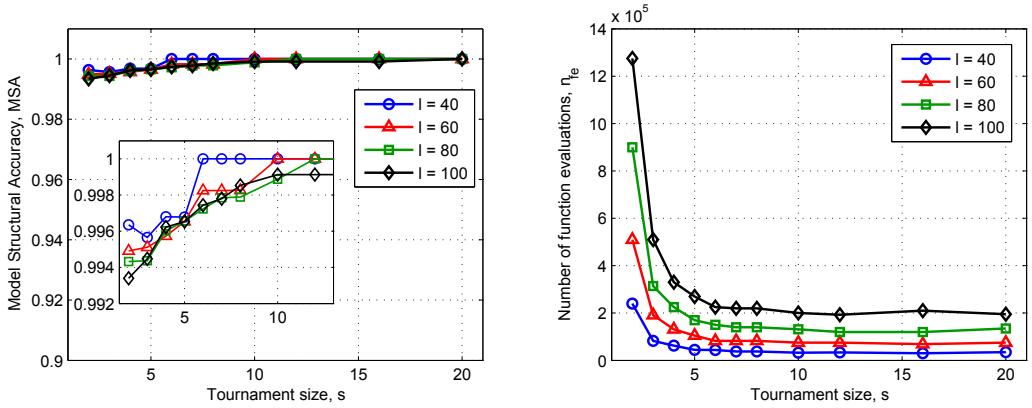
8. REFERENCES

[1] D. H. Ackley. *A connectionist machine for genetic hill climbing*. Kluwer Academic, Boston, 1987.

[2] T. Blickle and L. Thiele. A comparison of selection schemes used in genetic algorithms. *Evolutionary Computation*, 4(4):311–347, 1997.

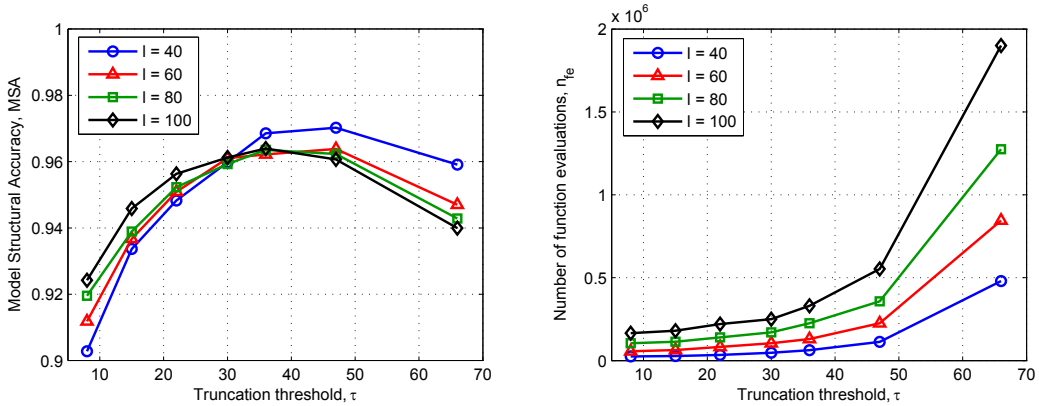
[3] A. Brindle. *Genetic Algorithms for Function Optimization*. PhD thesis, University of Alberta, Edmonton, Canada, 1981. Unpublished doctoral dissertation.

[4] D. M. Chickering, D. Heckerman, and C. Meek. A Bayesian approach to learning Bayesian networks with local structure. Technical Report MSR-TR-97-07, Microsoft Research, Redmond, WA.



(a) Model structural accuracy (b) Num. of function evaluations

Figure 6: Model quality and number of function evaluations for tournament selection with the K2 metric and $s-$penalty ($c_s = s$).



(a) Model structural accuracy (b) Num. of function evaluations

Figure 7: Model quality and number of function evaluations for truncation selection with the K2 metric.

[5] G. F. Cooper and E. H. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.

[6] K. Deb and D. E. Goldberg. Analyzing deception in trap functions. *Foundations of Genetic Algorithms 2*, pages 93–108, 1993.

[7] N. Friedman and M. Goldszmidt. Learning bayesian networks with local structure. *Graphical Models*, pages 421–459, 1999.

[8] D. E. Goldberg, B. Korb, and K. Deb. Messy genetic algorithms: Motivation, analysis, and first results. *Complex Systems*, 3(5):493–530, 1989.

[9] D. E. Goldberg and K. Sastry. *Genetic Algorithms: The Design of Innovation (2nd Edition)*. Springer, 2008.

[10] D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. Technical Report MSR-TR-94-09, Microsoft Research, Redmond, WA, 1994.

[11] P. Larrañaga and J. A. Lozano, editors. *Estimation of distribution algorithms: a new tool for Evolutionary Computation*. Kluwer Academic Publishers, Boston, MA, 2002.

[12] C. F. Lima, , D. E. Goldberg, M. Pelikan, F. G. Lobo, K. Sastry, and M. Hauschild. Influence of selection and replacement strategies on linkage learning in BOA. In K. C. Tan et al., editors, *IEEE Congress on Evolutionary Computation (CEC-2007)*, pages 1083–1090. IEEE Press, 2007.

[13] H. Mühlenbein and D. Schlierkamp-Voosen. Predictive models for the breeder genetic algorithm: I. Continuous parameter optimization. *Evolutionary Computation*, 1(1):25–49, 1993.

[14] J. Pearl. *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. Morgan Kaufmann, San Mateo, CA, 1988.

[15] M. Pelikan. *Hierarchical Bayesian Optimization Algorithm: Toward a New Generation of Evolutionary Algorithms*. Springer, 2005.

[16] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz. BOA: The Bayesian Optimization Algorithm. In W. Banzhaf et al., editors, *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99*, pages 525–532, San Francisco, CA, 1999. Morgan Kaufmann.

[17] K. Sastry. Evaluation-relaxation schemes for genetic and evolutionary algorithms. Master's thesis, University of Illinois at Urbana-Champaign, Urbana, IL, 2001.

[18] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6:461–464, 1978.

[19] D. Thierens and D. E. Goldberg. Mixing in genetic algorithms. In S. Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 38–45, San Mateo, CA, 1993. Morgan Kaufmann.