

**Influence of Selection and Replacement Strategies
on Linkage Learning in BOA**

**Cláudio F. Lima, Martin Pelikan,
David E. Goldberg, Fernando G. Lobo,
Kumara Sastry, and Mark Hauschild
IlligAL Report No. 2007013
April, 2007**

Illinois Genetic Algorithms Laboratory
University of Illinois at Urbana-Champaign
117 Transportation Building
104 S. Mathews Avenue Urbana, IL 61801
Office: (217) 333-2346
Fax: (217) 244-5705

Influence of Selection and Replacement Strategies on Linkage Learning in BOA

Cláudio F. Lima¹, Martin Pelikan², David E. Goldberg³, Fernando G. Lobo¹,
Kumara Sastry³, and Mark Hauschild²

¹Informatics Laboratory (UALG-ILAB)
Department of Electronics and Computer Science Engineering
University of Algarve, Campus de Gambelas, 8000-117 Faro, Portugal
{clima, flobo}@ualg.pt

²Missouri Estimation of Distribution Algorithm Laboratory (MEDAL)
Department of Mathematics and Computer Science
University of Missouri at St. Louis, St. Louis MO 63121
pelikan@cs.umsl.edu, mwh308@admiral.umsl.edu

³Illinois Genetic Algorithms Laboratory (IlliGAL)
Department of Industrial and Enterprise Systems Engineering
University of Illinois at Urbana-Champaign, Urbana IL 61801
{deg,ksastry}@uiuc.edu

April 3, 2007

Abstract

The Bayesian optimization algorithm (BOA) uses Bayesian networks to learn linkages between the decision variables of an optimization problem. This paper studies the influence of different selection and replacement methods on the accuracy of linkage learning in BOA. Results on concatenated m-k deceptive trap functions show that the model accuracy depends on a large extent on the choice of selection method and to a lesser extent on the replacement strategy used. Specifically, it is shown that linkage learning in BOA is more accurate with truncation selection than with tournament selection. The choice of replacement strategy is important when tournament selection is used, but it is not relevant when using truncation selection. On the other hand, if performance is our main concern, tournament selection and restricted tournament replacement should be preferred. These results aim to provide practitioners with useful information about the best way to tune BOA with respect to structural model accuracy and overall performance.

1 Introduction

Unlike traditional evolutionary algorithms (EAs), the Bayesian optimization algorithm (BOA) (Pelikan, Goldberg, & Cantú-Paz, 1999; Pelikan, 2005) replaces the standard crossover and mutation operators by building a probabilistic model of promising solutions and sampling from the corresponding probability distribution. This feature allows BOA and other advanced estimation of distribution algorithms (EDAs) (Larrañaga & Lozano, 2002; Pelikan, Goldberg, & Lobo, 2002) to

automatically identify the problem decomposition and important problem substructures, leading to superior performance for many problems when compared with EAs that use fixed, problem-independent variation operators.

Although the main feature of BOA and other EDAs is to perform efficient mixing of key substructures or building-blocks (BBs), they also provide additional information about the problem being solved. The probabilistic model of the population, that represents (in)dependencies among decision variables, is an important source of information that can be exploited to enhance the performance of EDAs even more, or to assist the user in a better interpretation and understanding of the underlying structure of the problem. Examples of using structural information from the probabilistic model for another purpose besides mixing are fitness estimation (Sastry, Pelikan, & Goldberg, 2004; Pelikan & Sastry, 2004; Sastry, Lima, & Goldberg, 2006), induction of global neighborhoods for mutation operators (Sastry & Goldberg, 2004; Lima, Pelikan, Sastry, Butz, Goldberg, & Lobo, 2006), hybridization and adaptive time continuation (Lima, Sastry, Goldberg, & Lobo, 2005; Lima, Pelikan, Sastry, Butz, Goldberg, & Lobo, 2006), substructural niching (Sastry, Abbass, Goldberg, & Johnson, 2005), offline (Yu & Goldberg, 2004) and online (Yu, Sastry, & Goldberg, 2007) population size adaptation.

In this paper we analyze the structural accuracy of the probabilistic models of BOA and their ability to represent underlying problem substructures. In particular, we use concatenated deceptive trap functions—where the optimal model is known, and accurate linkage learning is critical—to investigate the influence of different selection and replacement strategies on the quality of BOA models. The results show that as far as structural accuracy is concerned, truncation selection should be preferred over tournament selection, and the choice of replacement strategy matters when tournament selection is used. The results also show that if the objective is to obtain near-optimal solutions with high reliability using minimal number of function evaluations, then tournament selection with restricted tournament replacement is the best strategy for BOA.

The paper is structured as follows. The next section gives an outline of BOA. Section 3 motivates the importance of this work and makes a short survey on related work. In Section 4, the experimental setup used for measuring the structural accuracy of the probabilistic models is introduced. Section 5 analyzes the influence of selection in structural linkage learning, while in Section 6 the influence of the replacement method is studied. The paper ends with a summary and major conclusions.

2 Bayesian Optimization Algorithm

Estimation of distribution algorithms (Larrañaga & Lozano, 2002; Pelikan, Goldberg, & Lobo, 2002) replace traditional variation operators of EAs by building and sampling a probabilistic model of promising solutions to generate the offspring population. The Bayesian optimization algorithm (Pelikan, Goldberg, & Cantú-Paz, 1999; Pelikan, 2005) uses Bayesian networks as the probabilistic model to capture the (in)dependencies between the decision variables of the optimization problem.

BOA starts with an initial population that is usually generated at random. In each iteration, selection is performed to obtain a population of promising solutions. This population is then used to build the probabilistic model for the current generation. After the model structure is learned and its parameters estimated, the offspring population is generated by sampling from the distribution of modeled individuals. The new solutions are then incorporated into the original population by using any standard replacement method. The next iteration proceeds again from the selection phase until some stopping criteria is satisfied.

Bayesian networks (BNs) (Pearl, 1988) are powerful graphical models that combine probability

theory with graph theory to encode probabilistic relationships between variables of interest. 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 data to be modeled 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 $X = (X_1, X_2, \dots, X_\ell)$ 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 .

In BOA, both the structure and the parameters of the probabilistic model are searched and optimized to best fit the data (set of promising solutions). 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. The quality of a given network structure is quantified by using popular scoring metrics for BNs such as the Bayesian information criterion (BIC) (Schwarz, 1978) or the Bayesian-Dirichlet metric (BD) (Cooper & Herskovits, 1992; Heckerman, Geiger, & Chickering, 1994).

The parameters of a Bayesian network can be represented by a set of conditional probability tables (CPTs) specifying the conditional probabilities for each variable given all possible instances of the parent variables Π_i . Alternatively, these conditional probabilities can be stored in the form of local structures such as decision trees or decision graphs, allowing a more efficient and flexible representation of local conditional distributions. In this work, we use BNs with decision trees and the K2 metric with model-complexity penalty (a variant of BD metric) (Chickering, Heckerman, & Meek,).

The hierarchical BOA (hBOA) was later introduced by Pelikan and Goldberg (Pelikan & Goldberg, 2001; Pelikan, 2005) and resulted from the combination of BNs with local structures with a simple yet powerful niching method to maintain diversity in the population, known as restricted tournament replacement (RTR) (Harik, 1995). hBOA is able to solve hierarchical decomposable problems, in which the variable interactions are present at more than a single level.

3 Motivation and Related Work

While BOA is 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 the probabilistic 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 overfitting.

Analyzing the dependency groups captured by the Bayesian network with decision trees, it can be observed that while all important linkages are detected, spurious linkages are also incorporated in the model. By spurious linkage we mean additional variables that are considered together with a correct linkage group. While the structure of the BN captures such excessive complexity, the corresponding conditional probabilities nearly express independency between the spurious variables and the correct linkage, therefore not affecting the capability of sampling such variables as if they were almost independent.

Although the performance of BOA is not greatly affected by this kind of overfitting, several efficiency enhancement techniques for EDAs (Sastry, Pelikan, & Goldberg, 2004; Pelikan & Sastry, 2004; Sastry, Lima, & Goldberg, 2006; Sastry & Goldberg, 2004; Lima, Pelikan, Sastry, Butz, Goldberg, & Lobo, 2006; Lima, Sastry, Goldberg, & Lobo, 2005; Sastry, Abbass, Goldberg, & Johnson, 2005; Yu & Goldberg, 2004; Yu, Sastry, & Goldberg, 2007) crucially rely on the structural accuracy of the probabilistic models. One such example is the exploration of substructural neighborhoods for local search in BOA (Lima, Pelikan, Sastry, Butz, Goldberg, & Lobo, 2006). While significant speedups were obtained by incorporating model-based local search, the scalability of this speedup decreased for larger problem sizes due to overly complex model structures learned in BOA. Therefore, it is important to understand in which conditions the structural accuracy of the probabilistic models in BOA and other multivariate EDAs can be maximized. So far, only few studies have been done in this direction (Wu & Shapiro, 2006; Correa & Shapiro, 2006; Hauschild, Pelikan, Lima, & Sastry, 2007). In the remainder of this section we take a brief look at these works.

Wu and Shapiro (Wu & Shapiro, 2006) investigated the presence of overfitting when learning the probabilistic models in BOA and its consequences in terms of overall performance when solving random 3-SAT problems. CPTs (to encode the conditional probabilities) and the corresponding BIC metric were used. The authors concluded that overfitting does take place and that there is some correlation between this phenomenon and performance. The reduction in overfitting was proposed by using an early stopping criteria during the learning process of BNs, which gave some improvement in performance.

The trade-off between model complexity and performance in BOA was also studied recently (Correa & Shapiro, 2006). Correa and Shapiro looked at the performance achieved by BOA as a function of a parameter that determines the maximum number of incoming edges for each node. This parameter puts a limit on the number of parents for each variable, simplifying the search procedure for a model structure. This parameter was found to have a strong effect on the performance of the algorithm, for which there is a limited set of values where the performance is maximized. These results were obtained using CPTs and the corresponding BD metric. We should note that in fact this parameter is crucial if CPTs are used with the BD metric, however this is not the case for more sophisticated metrics that efficiently incorporate a complexity term to introduce pressure toward simpler models. This can be done better with the BIC metric for CPTs, or with the K2 metric for the case of decision trees (Pelikan, 2005).

More recently, Hauschild et al. (Hauschild, Pelikan, Lima, & Sastry, 2007) analyzed the probabilistic models built by hBOA for two common test problems: concatenated trap functions and 2D Ising spin glasses with periodic boundary conditions. The authors verified that the models learned closely correspond to the structure of the underlying problem. In their analysis, Hauschild et al. used truncation selection and restricted tournament replacement. In this paper, we will show that the results from (Hauschild, Pelikan, Lima, & Sastry, 2007) do not carry over to other combinations of selection and replacement methods. Before presenting these results, we discuss the details of our empirical analysis.

4 Experimental Setup for Measuring Structural Accuracy of Probabilistic Models

This section details the experimental setup and measures used to investigate the structural accuracy of the probabilistic models in BOA.

4.1 Test Problem and Experimental Setup

To investigate the structural accuracy of linkage learning 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). In this way the evaluation of the model structure quality to correctly detect both dependencies and independencies is performed.

The test problem considered is the $m - k$ deceptive trap function where m is the number of concatenated k -bit trap functions. Trap functions (Ackley, 1987; Deb & Goldberg, 1993) are relevant to test problem design because they bound an important class of nearly decomposable problems (Goldberg, 2002). The Trap function used (Deb & Goldberg, 1993) 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, and k is the size of the trap function. Note that for $k \geq 3$ the trap function is fully deceptive (Deb & Goldberg, 1993), 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 will lead to exponential scalability (Thierens & Goldberg, 1993). Thus, all variables corresponding to each trap function form a linkage group or BB partition and should be treated together by the probabilistic model. 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. A trap function with size $k = 5$ is used in our experiments.

A bisection method is used to determine the minimal population size required to solve the problem (Sastry, 2001). For each experiment, 10 independent bisection runs are performed. Each bisection run searches for the minimal population size required to find the optimum in 10 out of 10 independent runs. Therefore, the results for the minimal sufficient population size are averaged over 10 bisection runs, while the results for the number of function evaluations used are averaged over 100 (10×10) independent runs.

4.2 Measuring Structural Accuracy of Probabilistic Models

For accurate linkage learning in BOA at least one of the variables of each trap subfunction should depend on all remaining $k - 1$ variables, so that all k corresponding variables can be processed together by the probabilistic model. For example, the following dependency relation, $(X_1 \leftarrow X_2, X_3, X_4, X_5)$, encodes a linkage group between all variables for the first 5-bit trap subfunction. In addition, the remaining $l - k$ variables should not be part of that same dependency relation. If this is the case, the extra variables act as spurious linkage. For example, for the dependency relation $(X_1 \leftarrow X_2, X_3, X_4, X_5, X_6, X_{11})$, X_6 and X_{11} are spuriously linked variables. In essence, the dependencies between the groups of k bits corresponding to each subfunction must be discovered, while the remaining dependencies should be avoided to maximize mixing and minimize model complexity.

At each generation four different measures are analyzed taking into account only dependency groups of order k or higher:

Proportion of BBs with correct linkage group is the proportion of BB partitions or subfunctions (out of m) that have a dependency group in the model that only contains the corresponding

k variables.

Proportion of BBs with spurious linkage group is the proportion of BB partitions or subfunctions (out of m) that have a dependency group in the model that contains the corresponding k variables plus some additional spuriously linked variables.

Proportion of BBs with a linkage group is simply the sum of the two previous statistics. This measure is useful to confirm if every BB partition or subfunction is represented in the model, whether with only correct or additional spurious dependencies.

Average size of spurious linkage is the average number of spurious variables in the dependency relations that have spurious linkage (only those relations greater than k are considered).

5 Influence of the Selection Method

In this section, the influence of the selection method used in BOA, as well as the corresponding selection pressure, is investigated from the standpoint of structural accuracy of the learned linkage by the probabilistic models. Specifically, we consider two widely used ordinal selection schemes: Tournament and truncation selection. The particular choice of these two selection operators is related to the fact that these two schemes are the most frequently used in EDAs. Also, a previous study (Blickle & Thiele, 1997) on the comparison of several selection schemes has shown that these two schemes differ significantly in relevant features such as selection variance and loss of diversity.

In tournament selection (Goldberg, Korb, & Deb, 1989; Brindle, 1981), 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. If the individuals selected for the current tournament are candidates for other tournaments then the selection is made with replacement. On the other hand, if an individual is selected without replacement it cannot participate in further tournaments. While the expected outcome for both alternatives is the same, the former is a less noisy process. Therefore, in this study we use tournament selection without replacement.

In truncation selection (Mühlenbein & Schlierkamp-Voosen, 1993) the best $\tau\%$ 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 pressure (ratio of maximum to average fitness in the population) is increased, which means an increase in the selection strength. For the purpose of studying the influence of different selection strategies, the replacement strategy is kept as simple as possible: the offspring fully replace the parent population.

As an initial experiment, we analyze the influence of selection pressure in the required population size and total number of function evaluations. Figure 1 shows the population size and number of evaluations required for different tournament sizes and different problem sizes. From $s = 2$ to $s = 5$, both population size and number of evaluations is reduced for increasing tournament size. For $s \geq 5$, the requirements increase significantly, in particular for larger problem sizes. From these results, it appears to exist a sweet spot for the optimal tournament size, somewhere between $s = 4$ and $s = 5$. These qualitative results agree with a recent study (Yu, Sastry, Goldberg, & Pelikan, 2006) about the influence of selection pressure in the population size requirements for entropy-based EDAs.

In Figure 2, the speedup obtained by using the optimal tournament size for each problem size

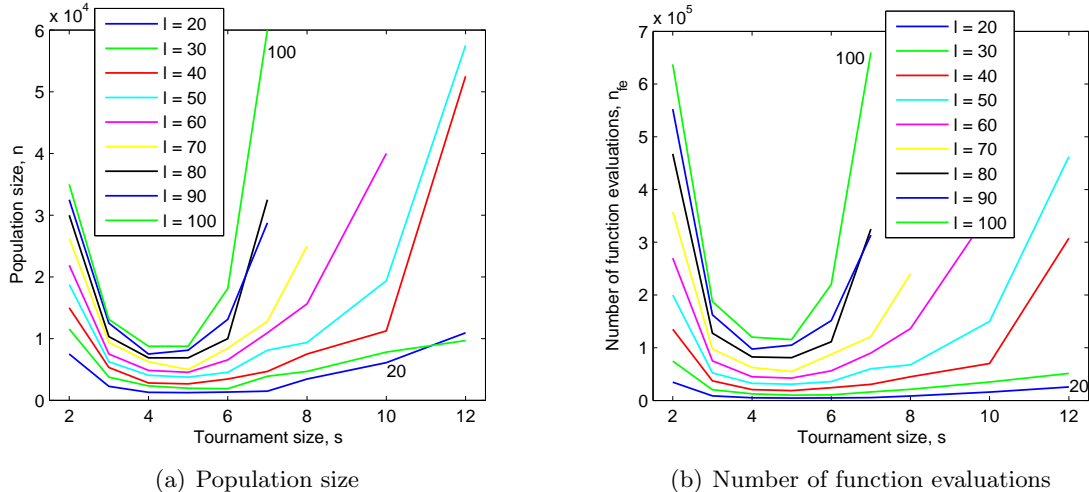


Figure 1: (a) Population size and (b) number of function evaluations required for different tournament sizes to solve concatenated 5-bit traps of varying string length ℓ .

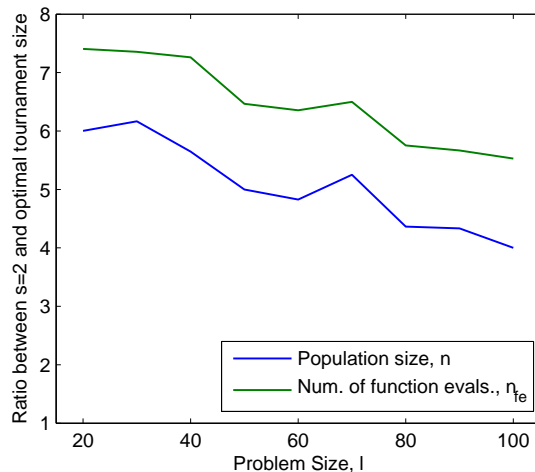


Figure 2: Speedup obtained when using the optimal tournament size for each problem size compared to the standard binary tournament ($s = 2$).

is plotted. This speedup is simply the ratio between the typical setting $s = 2$ and the optimal tournament size. The speedup decreases with increasing problem size, suggesting that this is not a scalable speedup and eventually will disappear for enough large problem sizes. Although it might be important to have the most appropriate tournament size for a given problem, this factor apparently loses relevance in terms of saving computational requirements as larger problems are considered.

We now turn to our head-to-head comparison between tournament and truncation selection having in mind the structural accuracy of the probabilistic models. In order to compare these two methods on a fair basis, different configurations for both methods with equivalent selection intensity are tested. The relation between selection intensity I , tournament size s , and truncation threshold τ is taken from (Blickle & Thiele, 1997) and is shown in Table 1.

Figures 3 and 4 show the linkage information captured by BOA during the run for tournament and truncation selection, respectively. The test problem considered is a concatenated trap function

Table 1: Equivalent tournament size (s) and truncation threshold (τ) for the same selection intensity (I) (Blickle & Thiele, 1997).

I	s	τ
0.56	2	66%
0.84	3	47%
1.03	4	36%

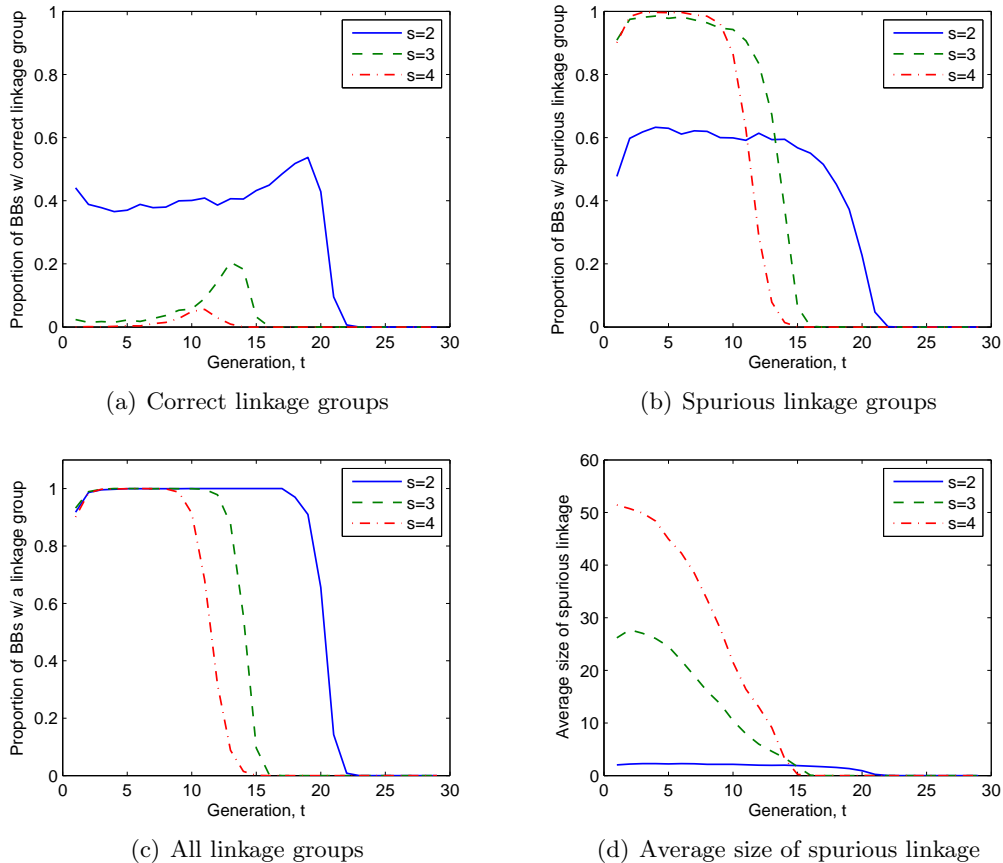


Figure 3: Linkage group information captured by the probabilistic model of BOA along the run for different tournament sizes, $s = \{2, 3, 4\}$, when solving $m = 24$ concatenated traps of order $k = 5$ ($\ell = 120$). Tournament selection and full replacement is used.

with $k = 5$ and $m = 24$, giving a total string length of $\ell = 120$.

For tournament selection, the proportion of BBs that have a correct linkage group (with exactly all corresponding k variables) represented in the model is quite low. Although for $s = 2$, nearly half of the BBs are still covered, for $s = 3$ and $s = 4$ this value approaches zero. Nevertheless, the BBs that are not covered with correct linkage groups are covered in spurious linkage groups, that have additional spuriously linked variables (other than the corresponding k variables). This can be observed in Figure 3 (c), where after the initial generation and until the end of the run, all BBs are represented (whether with only correct or additional spurious dependencies) in the probabilistic

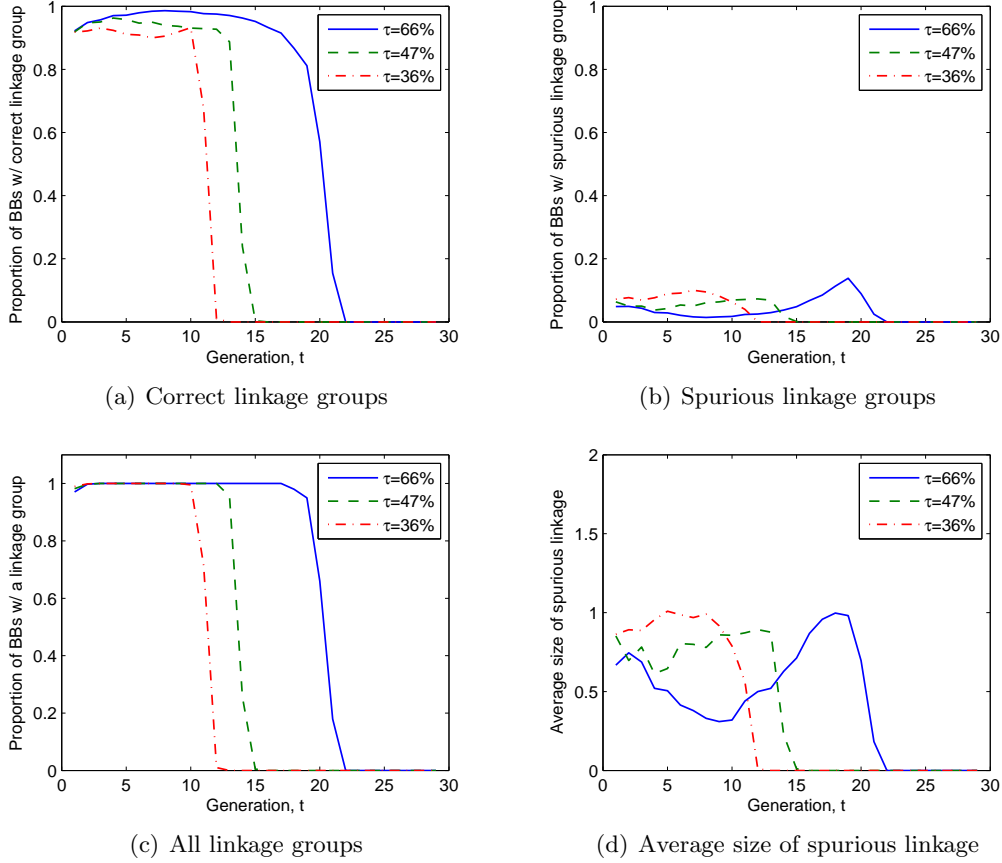


Figure 4: Linkage group information captured by the probabilistic model of BOA along the run for different truncation thresholds, $\tau = \{66\%, 47\%, 36\%\}$, when solving $m = 24$ concatenated traps of order $k = 5$ ($\ell = 120$). Truncation selection and full replacement is used.

model, a necessary condition to be able to solve the problem.

In Figure 3 (d), a drastic difference between binary tournament and higher tournament sizes can be noted for the average size of spurious linkages. In fact, the number of spurious variables added to the dependency groups is so high that one might wonder how BOA can sample new solutions efficiently. Analyzing the models in more detail, it can be seen that while the structure learned is much more complex than the underlying structure of the trap subfunctions, the parameters of the model nearly express independence between spurious and correlated variables. The detection of these weak dependencies is in part due to random fluctuations in the population (sample of limited size), that act as noise on the process of learning the real dependencies. Note that selection is performed at the individual-level rather than at the substructural or BB-level. Therefore, a top individual does not necessarily has mostly good substructures, which induces the learning process of BNs into some uncertainty, that is more pronounced in the initial generations. As the run proceeds and this source of noise is reduced by the iterative process of select-model-sample, the spurious linkage size reduces significantly.

For truncation selection the results are significantly better. With this selection method, BOA is able to represent almost 100% of the BBs with accurate linkage groups of order $k = 5$, while the size of spurious linkage is practically insignificant. Also, note that increasing the selection pressure (reducing the truncation threshold) hardly affect the accuracy of the linkage information.

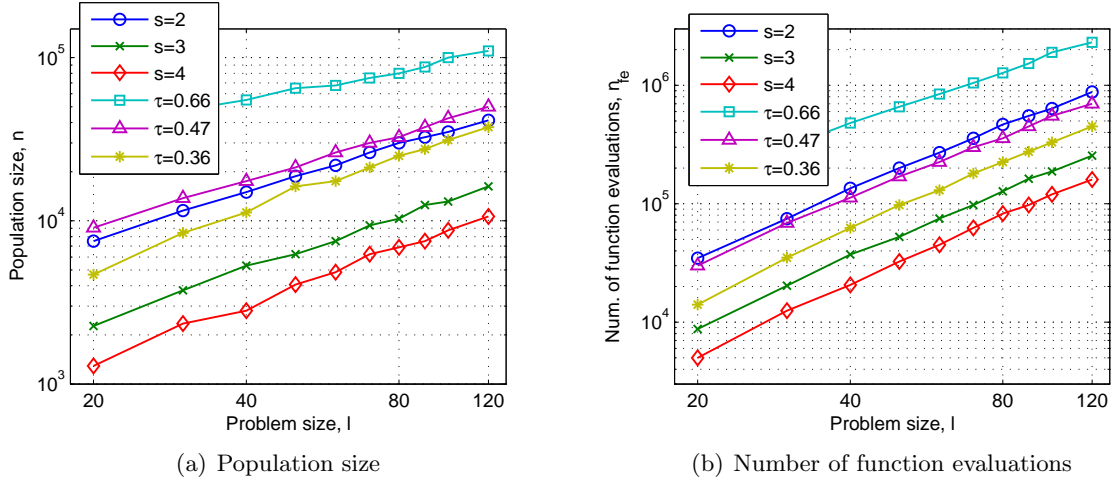


Figure 5: (a) Population size and (b) number of function evaluations required for different selection strategies when solving concatenated 5-bit traps of varying total string length ℓ . Full replacement is used. Although truncation selection requires larger population sizes, using tournament selection with equivalent population sizes to those required by truncation does not significantly improve the linkage information.

This completely different behavior between these two selection schemes lead us to take a look at their scalability behavior for population size, number of function evaluations, and average size of spurious linkage. That is what is shown in figures 5 and 6. The computational requirements for truncation are higher than for tournament selection by a significant however constant factor. Nevertheless, if we compare tournament selection with $s = 2$ and truncation selection with $\tau = 36\%$, the requirements for truncation are now smaller while in terms of structural accuracy truncation is still much better.

Further experiments (not plotted), where tournament selection was tested with the same population size used for truncation with the same selection pressure, showed that tournament selection only improves by a small factor the linkage information and is still much worse than truncation. This suggests that the difference observed isn't simply a matter of having enough population size, but more about the different way these selection operators work. In a detailed study (Blickle & Thiele, 1997) about the comparison of several selection schemes, it was shown that truncation and tournament selection are in fact quite different in terms of selection variance and loss of diversity.

Truncation selection has a higher loss of diversity and lower selection variance (for the same selection intensity) than tournament selection. Although we might expect that a lower loss of diversity and a higher selection variance would be desirable to avoid premature convergence, from the standpoint of EDAs where the probabilistic models are learned at every generation, a faster and clear distinction between good individuals and just above average individuals reduces the noise faced by the learning process of BNs.

There is also another important difference between these two operators. While in tournament selection the number of copies of an individual is proportional to its rank¹, in truncation no particular relevance is given to very good individuals, because all individuals selected get exactly one copy into the mating pool. This might present an interesting characteristic from the standpoint of BN learning.

¹The best individual gets exactly s copies, while other top individuals get on average a value close to s .

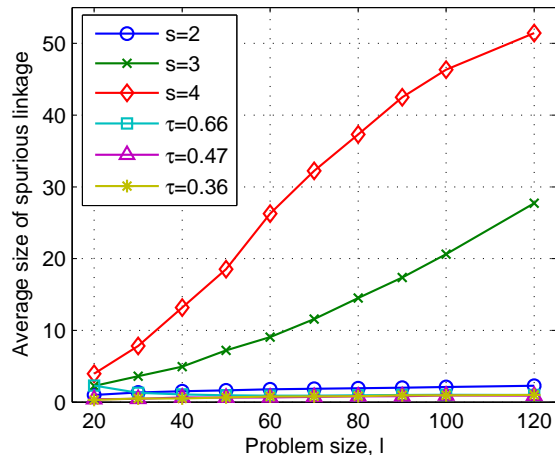


Figure 6: Average size of the spurious linkage for different selection strategies when solving concatenated 5-bit traps of varying total string length ℓ . Full replacement is used.

6 Influence of the Replacement Method

In this section, we analyze the influence of the replacement method used in BOA with respect to the accuracy of the probabilistic models versus overall performance. Three different replacement strategies are considered: full replacement (FR), elitist replacement (ER), and restricted tournament replacement (RTR).

In full replacement, the offspring population completely replaces the parent population at the end of each generation, therefore there is no overlap between these populations. For elitist replacement a given proportion of the worst individuals of the parent population is replaced by new individuals. A typical strategy is to replace the worst 50% individuals of the parent population by offspring, keeping the best 50% individuals for the next generation. Finally, a niching method called restricted tournament replacement (RTR) (Harik, 1995; Pelikan, 2005) is also tested. With RTR, each new solution X is incorporated into the original population using the following procedure:

1. Select a random subset of individuals W with size w from the original population.
2. Let Y be the solution from W that is most similar to X (in terms of genotypic distance).
3. Replace Y with X if the later is better, otherwise discard X .

The window size w is set to $w = \min\{\ell, n/20\}$ (Pelikan, 2005), where ℓ is the problem size and n is the population size. Note that RTR is the replacement method used in hBOA.

Figure 7 shows the results obtained for the different replacement strategies, using binary tournament selection. It can be seen that for all replacement methods a significant proportion of the BBs are not well represented in the models learned by BOA. Although all BBs have a linkage group that relates all k variables of interest, most of them have spurious linkage. This scenario is equally true for RTR and ER 50%, while for the FR method the linkage information captured is slightly more accurate. Figure 7 (d) shows the average size of spurious linkage, where RTR is clearly the worst option with respect to structural accuracy of the probabilistic models, while ER 50% performs better than RTR but still worse than FR, for which the average size is relatively constant and never goes beyond two. Note that the replacement strategy does not have the same impact as tournament size in the spurious linkage size.

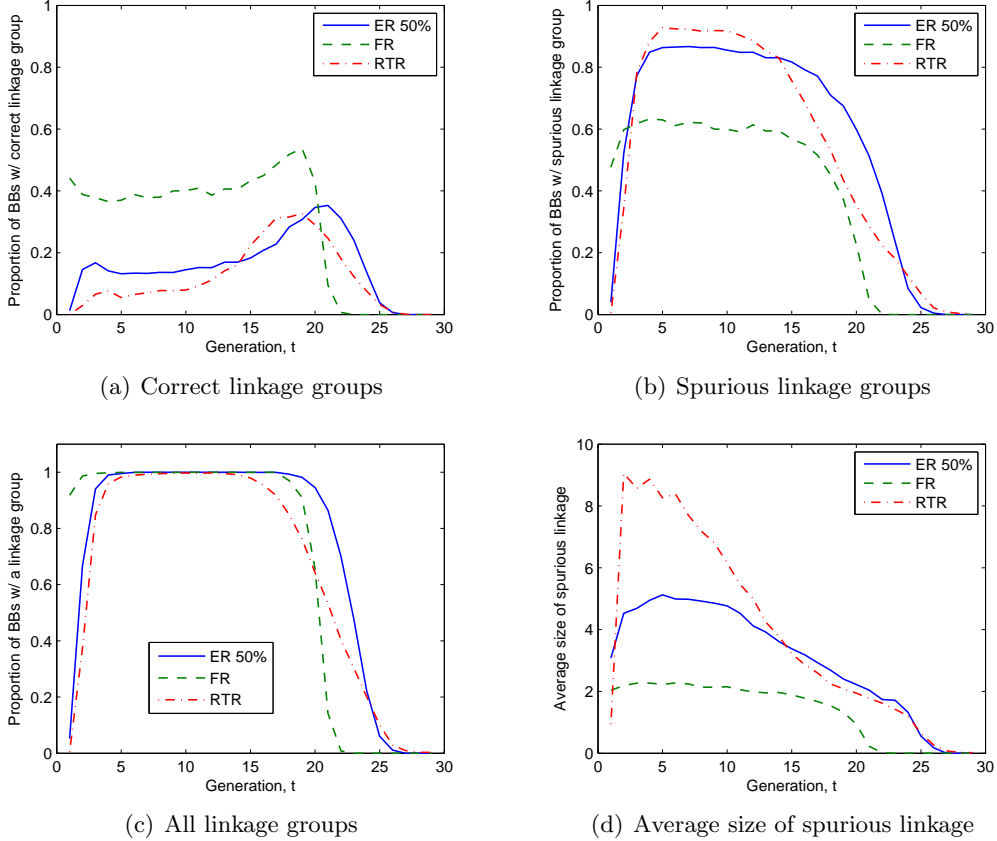


Figure 7: Linkage group information captured by the probabilistic model of BOA along the run for different replacement methods when solving $m = 24$ concatenated traps of order $k = 5$ ($\ell = 120$). ER 50% stands for the replacement of the worst 50% parents, FR for full replacement, and RTR for restricted tournament replacement. Binary tournament selection is used.

In figures 8 and 9, the scalability of the replacement methods is depicted. Additional values for ER were tested to investigate the progressive influence of the proportion of elitism on the structural accuracy of the learned models. RTR is clearly the strategy that requires smaller population size and fewer evaluations, however at the cost of higher spurious linkage sizes than the remaining replacement methods. This is due to the niching capabilities of RTR. By preserving diversity in the population, BOA can solve the given problem with smaller population sizes and consequently with fewer evaluations. However, the quality of the model is not the best because the drawback of using tournament selection is aggravated with smaller population sizes and increased diversity due to niching. For ER, as the elitist proportion is reduced the structure captured by the models gradually improves, until we get to the case of FR where the best result is obtained in terms of model accuracy. Note that for ER, where the best proportion of individuals is always kept in the population, the model is not required to be as accurate as in FR, where the sampled individuals fully replace the original population, in which case the quality of sampled solutions have a stronger influence on the probability of success to find the optimum.

While these results were obtained for binary tournament selection, additional experiments (not plotted) were performed with truncation selection. In this case, the replacement strategies were found not to have a significant impact on model accuracy and all methods performed very similar

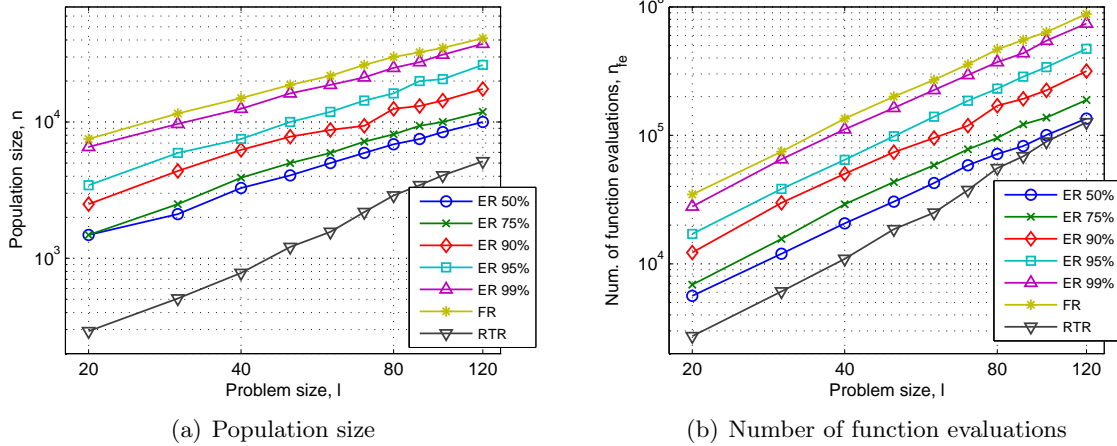


Figure 8: (a) Population size and (b) number of function evaluations required for different replacement strategies when solving concatenated 5-bit traps with varying total string length l . Binary tournament selection is used.

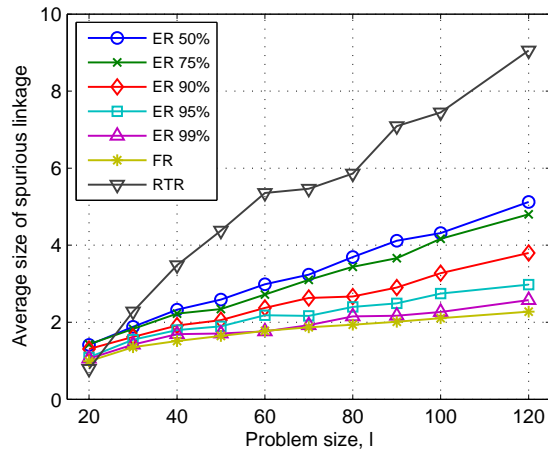


Figure 9: Average size of the spurious linkage for different replacement strategies when solving concatenated 5-bit traps with varying total string length l . Binary tournament selection is used.

to truncation selection with FR (see figures 4, 5, and 6).

7 Summary and Conclusions

In this work we have empirically analyzed the influence of selection and replacement strategies on the structural accuracy of linkage learning in BOA, for concatenated m - k deceptive trap functions. In essence, using truncation instead of tournament selection is much better for the purpose of having accurate structural linkage information. Although truncation selection requires larger population sizes, using tournament selection with equivalent population sizes to those required by truncation does not significantly improve the linkage information. For the same purpose, the replacement strategy was found to be relevant only if tournament selection is used, in which case the full replacement of the parents by their offspring is the most appropriate strategy. On the other hand,

if overall performance (number of function evaluations) is our main concern, tournament selection and restricted tournament replacement are clearly the best options.

The results presented in this paper provide important information to practitioners about the trade-off between the parameters used in BOA (and consequent computational cost) and the degree of accuracy for the learned linkage information, as well as the best way to tune BOA with respect to this goal.

Acknowledgments

This work was sponsored by the Portuguese Foundation for Science and Technology (FCT/MCES) under grants SFRH/BD/16980/2004 and POSC/EEA-ESE/61218/2004, the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant FA9550-06-1-0096, the National Science Foundation under NSF CAREER grant ECS-0547013, ITR grant DMR-03-25939 at Material Computation Center, UIUC. 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.

References

- Ackley, D. H. (1987). *A connectionist machine for genetic hill climbing*. Boston: Kluwer Academic.
- Blickle, T., & Thiele, L. (1997). A comparison of selection schemes used in genetic algorithms. *Evolutionary Computation*, 4(4), 311–347.
- Brindle, A. (1981). *Genetic algorithms for function optimization*. Doctoral dissertation, University of Alberta, Edmonton, Canada. Unpublished doctoral dissertation.
- Chickering, D. M., Heckerman, D., & Meek, C. *A Bayesian approach to learning Bayesian networks with local structure* (Technical Report MSR-TR-97-07). Redmond, WA: Microsoft Research.
- Cooper, G. F., & Herskovits, E. H. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9, 309–347.
- Correa, E. S., & Shapiro, J. L. (2006). Model complexity vs. performance in the bayesian optimization algorithm. In Runarsson, T. P., et al. (Eds.), *PPSN IX: Parallel Problem Solving from Nature, LNCS 4193* (pp. 998–1007). Springer.
- Deb, K., & Goldberg, D. E. (1993). Analyzing deception in trap functions. *Foundations of Genetic Algorithms 2*, 93–108.
- Goldberg, D. E. (2002). *The design of innovation - lessons from and for competent genetic algorithms*. Norwell, MA: Kluwer Academic Publishers.
- Goldberg, D. E., Korb, B., & Deb, K. (1989). Messy genetic algorithms: Motivation, analysis, and first results. *Complex Systems*, 3(5), 493–530. Also IlliGAL Report No. 89003.

- Harik, G. R. (1995). Finding multimodal solutions using restricted tournament selection. *Proceedings of the Sixth International Conference on Genetic Algorithms*, 24–31.
- Hauschild, M., Pelikan, M., Lima, C. F., & Sastry, K. (2007). *Analyzing probabilistic models in hierarchical BOA on traps and spin glasses* (MEDAL Report No. 2007001). St. Louis, MO: University of Missouri at St. Louis.
- Heckerman, D., Geiger, D., & Chickering, D. M. (1994). *Learning Bayesian networks: The combination of knowledge and statistical data* (Technical Report MSR-TR-94-09). Redmond, WA: Microsoft Research.
- Larrañaga, P., & Lozano, J. A. (Eds.) (2002). *Estimation of distribution algorithms: a new tool for evolutionary computation*. Boston, MA: Kluwer Academic Publishers.
- Lima, C. F., Pelikan, M., Sastry, K., Butz, M., Goldberg, D. E., & Lobo, F. G. (2006). Substructural neighborhoods for local search in the Bayesian optimization algorithm. In Runarsson, T. P., et al. (Eds.), *PPSN IX: Parallel Problem Solving from Nature, LNCS 4193* (pp. 232–241). Springer.
- Lima, C. F., Sastry, K., Goldberg, D. E., & Lobo, F. G. (2005). Combining competent crossover and mutation operators: a probabilistic model building approach. In Beyer, H., et al. (Eds.), *Proceedings of the ACM SIGEVO Genetic and Evolutionary Computation Conference (GECCO-2005)* (pp. 735–742). ACM Press.
- Mühlenbein, H., & Schlierkamp-Voosen, D. (1993). Predictive models for the breeder genetic algorithm: I. Continuous parameter optimization. *Evolutionary Computation*, 1(1), 25–49.
- Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. San Mateo, CA: Morgan Kaufmann.
- Pelikan, M. (2005). *Hierarchical Bayesian Optimization Algorithm: Toward a new generation of evolutionary algorithms*. Springer.
- Pelikan, M., & Goldberg, D. E. (2001, 7-11 July). Escaping hierarchical traps with competent genetic algorithms. In Spector, L., et al. (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)* (pp. 511–518). San Francisco, CA: Morgan Kaufmann. Also IlliGAL Report No. 2000020.
- Pelikan, M., Goldberg, D. E., & Cantú-Paz, E. (1999). BOA: The Bayesian Optimization Algorithm. In Banzhaf, W., et al. (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99* (pp. 525–532). San Francisco, CA: Morgan Kaufmann. Also IlliGAL Report No. 99003.
- Pelikan, M., Goldberg, D. E., & Lobo, F. (2002). A survey of optimization by building and using probabilistic models. *Computational Optimization and Applications*, 21(1), 5–20. Also IlliGAL Report No. 99018.
- Pelikan, M., & Sastry, K. (2004). Fitness inheritance in the bayesian optimization algorithm. In Deb, K., et al. (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2004), Part II, LNCS 3103* (pp. 48–59). Springer.
- Sastry, K. (2001). *Evaluation-relaxation schemes for genetic and evolutionary algorithms*. Master’s thesis, University of Illinois at Urbana-Champaign, Urbana, IL. Also IlliGAL Report No. 2002004.
- Sastry, K., Abbass, H. A., Goldberg, D. E., & Johnson, D. D. (2005). Sub-structural niching in estimation distribution algorithms. ACM Press.

- Sastry, K., & Goldberg, D. E. (2004). Designing competent mutation operators via probabilistic model building of neighborhoods. In Deb, K., & et al. (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2004), Part II, LNCS 3103* (pp. 114–125). Springer. Also IlliGAL Report No. 2004006.
- Sastry, K., Lima, C. F., & Goldberg, D. E. (2006). Evaluation relaxation using substructural information and linear estimation. In Keijzer, M., et al. (Eds.), *Proceedings of the ACM SIGEVO Genetic and Evolutionary Computation Conference (GECCO-2006)* (pp. 419–426). ACM Press.
- Sastry, K., Pelikan, M., & Goldberg, D. E. (2004). Efficiency enhancement of genetic algorithms via building-block-wise fitness estimation. In *Proceedings of the IEEE International Conference on Evolutionary Computation* (pp. 720–727). Also IlliGAL Report No. 2004010.
- Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6, 461–464.
- Thierens, D., & Goldberg, D. E. (1993). Mixing in genetic algorithms. In Forrest, S. (Ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms* (pp. 38–45). San Mateo, CA: Morgan Kaufmann.
- Wu, H., & Shapiro, J. L. (2006). Does overfitting affect performance in estimation of distribution algorithms. In Keijzer, M., et al. (Eds.), *Proceedings of the ACM SIGEVO Genetic and Evolutionary Computation Conference (GECCO-2006)* (pp. 433–434). ACM Press.
- Yu, T.-L., & Goldberg, D. E. (2004). Dependency structure matrix analysis: Offline utility of the dependency structure matrix genetic algorithm. In Deb, K., et al. (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2004), Part II, LNCS 3103* (pp. 355–366). Springer.
- Yu, T.-L., Sastry, K., & Goldberg, D. E. (2007). Population size to go: Online adaptation using noise and substructural measurements. In Lobo, F. G., et al. (Eds.), *Parameter Setting in Evolutionary Algorithms* (pp. 205–224). Springer.
- Yu, T.-L., Sastry, K., Goldberg, D. E., & Pelikan, M. (2006). *Population sizing for entropy-based model building in genetic algorithms* (IlliGAL Report No. 2006020). Urbana, IL: University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory.