Toward an Understanding of the Quality and Efficiency of Model Building for Genetic Algorithms

Tian-Li Yu and David E. Goldberg

Illinois Genetic Algorithms Laboratory (IlliGAL)
Department of General Engineering
University of Illinois at Urbana-Champaign
104 S. Mathews Ave, Urbana, IL 61801
{tianliyu, deg}@illigal.ge.uiuc.edu

Abstract. This paper investigates the linkage model building for genetic algorithms. By assuming a given quality of the linkage model, an analytical model of time to convergence is derived. Given the computational cost of building the linkage model, an estimated total computational time is obtained by using the derived time-to-convergence model. The models are empirically verified. The results can be potentially used to decide whether applying a linkage-identification technique is worthwhile and give a guideline to speed up the linkage model building.

1 Introduction

Holland [1] suggested that operators learning linkage information to recombine alleles might be necessary for genetic algorithm (GA) success. Many such methods [2] have been developed to solve the linkage problem. The linkage model can be implicit (*e.g.* LLGA [3]) or explicit (*e.g.* LINC [4]), probabilistic (probabilistic model building GAs [5], or estimation of distribution algorithms [6]) or deterministic (*e.g.* DSMGA [7]). Those methods have different strength in identifying linkage and consume different computational time. Because some of the linkage-identification methods are computationally expensive, recently there is a trend of applying speedup techniques on those linkage-identification methods including parallelism [8]. Another possible speedup technique is the evaluation relaxation scheme [9]. In a unified point of view, a linkage-identification method with the evaluation relaxation technique applied can be considered as another linkage-identification method which is more efficient and possibly less accurate. With the same idea, a simple GA (sGA) can be thought as a GA with a linkage-identification method which does not consume additional computations and always *reports* tight linkages.

Two questions might frequently come into a GA researcher's mind: (1) On which classes of problems does a specific GA design have strength and weakness, and (2) is a speedup technique worthwhile to use? Take the Bayesian optimization algorithm (BOA) as an example. BOA works well on problems with

K. Deb et al. (Eds.): GECCO 2004, LNCS 3103, pp. 367–378, 2004.

linkages; however, for problems where linkage is not important, like OneMax, a sGA would perform better since building a Bayesian network is computationally time-consuming [10]. A linkage-identification method with the evaluation relaxation techniques is more efficient but possibly less accurate. The lower accuracy might elongate the convergence time, and speeding up the model-building process might result in a slower GA convergence. Although the above two questions seem loosely related, they boil down to the following more basic question: Given two different model builders with different accuracies and different computational costs, should one use a more accurate, but more computationally expensive model builder or a less accurate, but less computationally expensive model builder?

The purpose of this paper is to answer the above question, and by doing this, the results lead us toward a better understanding of the relationship between building linkage models and GA convergence. The paper first defines the errors of a linkage model, and then time to convergence is derived for a given number of errors of the linkage model. The time-to-convergence model is then used to derive the total computational time. A number of experiments are done to verify the models derived in the paper. Finally, discussions of the contributions and possible future work conclude this paper.

2 Time to Convergence for Linkage Model Building

This section derive the time-to-convergence model of GA by assuming a given linkage model quality. All derivations done by assuming an infinite population size and prefect mixing by population-wise crossover. For simplicity, we further assume that the problem contains a unique global optimum.

2.1 The Errors of a Linkage Model

The term "linkage" is widely used in GA field, but defining linkage is not an easy task. In this paper, the linkage can be loosely defined as follows. *If linkage exists between two genes, recombination might result in lowly fit offspring with high probability if those two genes are not transferred together from parents to offspring.* A group of highly linked genes forms a linkage group, or a building block (BB) in [2].

A linkage model is a model telling which genes form linkage groups. For instance, the boolean flags in LEGO [11], the genetic ordering in LLGA [12], the clustering model in eCGA [13], and the DSM clustering in DSMDGA [7] are all linkage models. Two different types of errors can happen when a linkage model is adopted to describe the genetic linkage. One is that the linkage model links those gene which are not linked in reality. The other is that the linkage model does not link those gene which are linked in reality. The first type of error does not disrupt correct BBs but they slow down BB mixing. Since perfect mixing (in short, BBs are uniformly distributed over the population on each particular position) is one of the presumptions of the time-to-convergence model used later

in the paper, this paper only focuses on the second type of error and leaves the first type of error as future work.

The quality of a linkage model can be quantified by the number of errors it makes. For example, consider a problem with four BBs, where $\{BB_1, BB_2, BB_3, BB_4\}$=$\{\{1,2,3\}, \{4,5,6\}, \{7,8,9\}, \{10,11,12\}\}$, and a linkage model $\{BB'_1, BB'_2, BB'_3, BB'_4, BB'_5\}$=$\{\{1,2\}, \{3,4,5,6\}, \{7,8\}, \{9,10,11\}, \{12\}\}$. By ignoring the first type of errors, the linkage model can be re-expressed as $\{\{1,2\}, \{3\}, \{4,5,6\}, \{7,8\}, \{9\}, \{10,11\}, \{12\}\}$. As a result, the linkage model produces 3 errors and only BB_2 is correctly identified.

2.2 Building Block Disruptions

According to [2], effectively mixing BBs is critical for a GA success. In most traditional GAs, BB mixing is done by performing crossover. However, if BBs are not correctly identified, crossover will also disrupt BBs (addressed in Holland's [1] schema theory). This subsection derives the upper bound of the expected number of BB disruptions given the number of errors of the linkage model.

Before derivations, it is convenient to define two terms, a *correct BB* and an *incorrect BB*. If the genes in a BB have the same values as those genes at the same locations of the globally optimal solution, the BB is called a correct BB; otherwise, it is called an incorrect BB. A BB disruption occurs when a correct BB becomes an incorrect BB after crossover.

By the assumption, after crossover is performed, a misidentified BB is recombined by two portions which come from two different BBs. When one portion comes from a correct BB and the other portion comes from an incorrect BB, a BB disruption probably occurs. To be conservative, we assume that recombining a portion of a correct BB with a portion of an incorrect BB always results in an incorrect BB. That happens when the most competitive incorrect BB is exactly the compliment of the correct BB. Likewise, we assume that recombining incorrect BBs always produces incorrect BBs.

Assume that there is a proportion p of correct BBs in the current population. For a randomly chosen BB, a BB disruption occurs when (1) it is misidentified, (2) it is a correct BB, and (3) it is going to be recombined with an incorrect BB. Therefore, the probability of a BB disruption occurrence is given by

$$\left(1-(1-\frac{1}{m})^e\right)p(1-p), \tag{1}$$

where m is the number of BBs in a chromosome. When the number of errors e is much smaller than the number of BBs m, it is valid to assume that only one errors occurs for each misidentified BB. The above equation can be approximated as

$$\frac{e}{m}p(1-p). \tag{2}$$

In a population of size N, there are total Nm BBs. The expected number of BB disruptions is then $Nep(1-p)$.

2.3 Time-to-Convergence Model

Mühlenbein and Schlierkamp-Voosen [14] gave the following time-to-convergence model for OneMax problem by assuming a infinite population size and prefect mixing.

$$t_{conv} = \left(\frac{\pi}{2} - \arcsin(2p_0 - 1)\right)\frac{\sqrt{l}}{I}, \tag{3}$$

where p_0 is the initial proportion of ones, I is selection intensity, and l is the length of chromosome. The time-to-convergence model can be derived from the following equation [14,15].

$$p_{t+1} - p_t = \frac{I}{\sqrt{l}}\sqrt{p_t(1-p_t)}, \tag{4}$$

where p_t is the proportion of ones at generation t.

Miller [16] extended the time-to-convergence model to problems with uniformly scaled BBs. If the linkage model successfully identifies every BB, by treating correct BBs as 1's and incorrect BBs as 0's, the problem is then similar to the OneMax problem. The only difference is that the growth of correct BBs are slower. The reason is that a chromosome with more correct BBs does not always have a higher fitness value. For example, {0" 0" 1} might have a lower fitness value than {0' 0' 0'}, where 1 represents a correct BB, 0' represents an incorrect BB with a high fitness value, and 0" represents another incorrect BB with a low fitness value. The growth of correct BBs is then modelled as follows.

$$p_{t+1} - p_t = \frac{I'}{\sqrt{m}}\sqrt{p_t(1-p_t)}, \tag{5}$$

where $I' \leq I$, and p_t is the proportion of correct BBs at generation t.

When the linkage model has some errors, the growth of correct BBs is slowed down. After selection, the growth of correct BBs is still governed by Equation 5, $p_{t,selected} = p_t + \frac{I'}{\sqrt{m}}\sqrt{p_t(1-p_t)}$. The proportion of disrupted BBs is given by

$$\frac{e}{m}p_{t,selected}(1 - p_{t,selected}). \tag{6}$$

Hence, the proportion of correct BBs for the next generation is

$$p_{t+1} = p_{t,selected} - \frac{e}{m}p_{t,selected}(1 - p_{t,selected}). \tag{7}$$

By approximating the BB disruption in Equation 6 as $\frac{e}{m}p_t(1-p_t)$ (the proportion of disrupted BBs is calculated according to the proportion of correct BBs before selection) and adopting $p(1-p) \leq \frac{1}{2}\sqrt{p(1-p)}$ for $0 \leq p \leq 1$, the proportion of disrupted BBs can be approximated as:

$$\frac{e}{2m}\sqrt{p_t(1-p_t)}. \tag{8}$$

The proportion of correct BBs in the next generation is then given by

$$p_{t+1} - p_t = (\frac{I'}{\sqrt{m}} - \frac{e}{2m})\sqrt{p_t(1-p_t)}. \tag{9}$$

Following a similar procedure in [14] and [15], one can derive the time-to-convergence model.

$$t_{conv} = \frac{\left(\frac{\pi}{2} - \arcsin(2p_0 - 1)\right)}{\frac{I'}{\sqrt{m}} - \frac{e}{2m}}. \tag{10}$$

The dimensionless model can be obtained as

$$\frac{t_{conv}(e=0)}{t_{conv}} = 1 - \frac{e}{2I'\sqrt{m}}. \tag{11}$$

The above equation also suggests that when $e \geq 2I'\sqrt{m}$, BB disruptions are severer than BB growths; the behavior of the GA is then like a random search and difficult to converge. Therefore, defining the *critical number of errors* $e_{critial}$ as the largest number of BBs that a linkage model could misidentify while a $(m-1)$-BB convergence is still possible, the following relation between $e_{critial}$ and m can be expressed as

$$e_{critial} = 2I'\sqrt{m}. \tag{12}$$

Since I' is only loosely related to the problem, the key idea of the above equation is that $e_{critial} = O(\sqrt{m})$.

3 Overall Computational Time

This section models the overall computational time using the time-to-convergence model in the previous section. The overall computational time is then used to derive an optimal decision of which linkage model should be used. A similar methods of modelling in this section can be found in [9].

Assuming the GA operators consume α computational time each generation, if a linkage model on average misidentifies e BBs and consumes β computational time, the overall computation time is then

$$T = t_{conv}(e)(\alpha + \beta). \tag{13}$$

Suppose we have two linkage-identification methods, M_1 and M_2, which misidentifies e_1 and e_2 BBs, and consumes α_1 and α_2 computational time, respectively. The ratio of the overall computational time of GAs which adopt those two linkage-identification methods is given by

$$\frac{T_{M_1}}{T_{M_2}} = \frac{2I'\sqrt{m} - e_2}{2I'\sqrt{m} - e_1} \cdot \frac{\alpha + \beta_1}{\alpha + \beta_2}. \tag{14}$$

If the ratio is smaller than 1, method M_1 should be used, and vice versa.

The above equation is difficult to use in practice mainly because the the number of BBs that a linkage model misidentifies is not easy to estimate. Nevertheless, the equation gives some insights and mathematical foundation to the following observations.

1. When the fitness function evaluation is computationally expensive ($\alpha >> \beta$), the first term ($\frac{2I'\sqrt{m}-e_2}{2I'\sqrt{m}-e_1}$) dominates the decision, and a time-consuming, but more accurate linkage-identification method is favored.

2. On the contrary, when the fitness function evaluation is relatively computationally inexpensive ($\alpha << \beta$), the second term ($\frac{\alpha+\beta_1}{\alpha+\beta_2}$) dominates the decision, and a less accurate, but computational efficient linkage-identification method is favored.

When errors are few ($e_1, e_2 << \sqrt{m}$) and the computational cost of the linkage model builder is relatively cheap compared to the GA operators ($\beta_1, \beta_2 << \alpha$), the above equation can be approximately simplified as

$$\frac{T_{M_1}}{T_{M_2}} = 1 + \frac{e_1 - e_2}{2I'\sqrt{m}} + \frac{\beta_1 - \beta_2}{\alpha}. \tag{15}$$

The above equation suggests the following definitions: The *quality* of a linkage model is $Q = 1 - \frac{e}{2I'\sqrt{m}}$ and the *relative cost* of a linkage model is $c = \frac{\beta}{\alpha}$. For any linkage model with $Q < 0$ or $e > 2I'\sqrt{m}$, the GA is difficult to converge. Given two linkage-identification methods M_1 and M_2 with qualities Q_1, Q_2 and their relative costs c_1, c_2 respectively, by defining $\triangle Q = Q_1 - Q_2$ and $\triangle c = c_1 - c_2$, the decision ratio becomes

$$\frac{T_{M_1}}{T_{M_2}} = 1 + (\triangle c - \triangle Q). \tag{16}$$

Therefore, if $\triangle c < \triangle Q$, M_1 is better; otherwise, M_2 is better. Consider M_2 as an evaluation relaxation version of M_1: M_2 is more computationally inexpensive but less accurate than M_1 ($\triangle c > 0$ and $\triangle Q > 0$). The evaluation relaxation is worthwhile only when $\triangle c > \triangle Q$, or in other words, when the save of relative cost is greater than the loss of quality.

4 Empirical Results

This section presents the experiments that empirically verify our models. This section first describes the design of the experiments, and then it shows the empirical results followed by discussions.

4.1 Experiment Design

The test function should be carefully chosen to fulfill the worst-case analysis: If a BB is misidentified, most likely it will be disrupted by crossover. Based on the reason, a MaxTrap function is used (for more details and analysis of a trap function, see [17]). A correct BB in a 5-bit trap is 11111, and its most competing BB is 00000. If the BB is misidentified, any crossover would result in incorrect BBs with some ones and zeros in them. By the nature of the trap function, those incorrect BBs would then tend to become the most competing BB, 00000.

In addition, two assumptions used in the time-to-convergence model derivations should be satisfied: (1) Infinite population-sizing and (2) perfect mixing. In implementation, a large-enough population size is used and uniform crossover is performed to ensure high mixing rate. In particular, a pair-wise BB-specific uniform crossover is used, the main difference between a population-wise BB-specific uniform crossover and the pair-wise BB-specific uniform crossover is that the amount of BB disruptions predicted in Equation 6 is reduced by a factor of 2, because the swap at any cross-site only is performed with a probability 0.5.

A 5×50 and a 5×100 MaxTrap are tested. The 5-bit trap is defined as:

$$trap_5(u) = \begin{cases} \frac{4-u}{5}, & u = 0, 1, 2, 3, 4 \\ 1, & u = 5 \end{cases}, \tag{17}$$

where u denotes the number of ones in the input 5-bit block.

A linkage model without any error ($e = 0$) gives the following BB information: $\{\{1, 2, 3, 4, 5\},\{6, 7, 8, 9, 10\},\cdots\}$. For $e > 0$, first, e BBs are randomly selected, their genes are randomly shuffled, and then those selected BBs are randomly split into two parts. For example, a randomly selection BB {6, 7, 8, 9, 10} might be shuffled as {6, 9, 10, 7, 8} and then split into {6, 9} and {10, 7, 8}. The processed BB information is then used to perform a pair-wise BB-specific uniform crossover. Tournament selection with tournament size $s = 2$ is used. According to Blickle and Thiele [18], the selection intensity $I \simeq 0.5763$. Assuming I' is a constant, I' can be then estimated by comparing the estimation of time to convergence given by Equation 10 and the empirical time to convergence for $e = 0$. As a result, $I' \simeq 0.752I \simeq 0.4334$. All experiments are averaged over 100 independent runs.

To approximate the asymptotic behavior of the time-to-convergence model, four times of the population size estimated by the gambler's ruin model [12] are used. For example, a population size of 1539 should supply enough BBs for GAs to process the 5×100 MaxTrap function. In the experiments, the population size is set to 6156.

4.2 Results and Discussions

The relationship between the number of BB disruptions and the correct BB proportion for different linkage model errors for the 5×100 MaxTrap is presented in Figure 1. The figure shows that the derived model (Equation 6) is more accurate when e is small. The reason is that Equation 6 is an overestimate which

ignores the possibility that correct BBs might be produced from the combination of two incorrect BBs. When e is larger, the crossover of 11111 and 00000 gives more incorrect BBs with some ones and zeros in them. The recombination of those BBs then has a higher probability to reproduce the correct BB, 11111. Note that the model indeed bounds the empirical data.

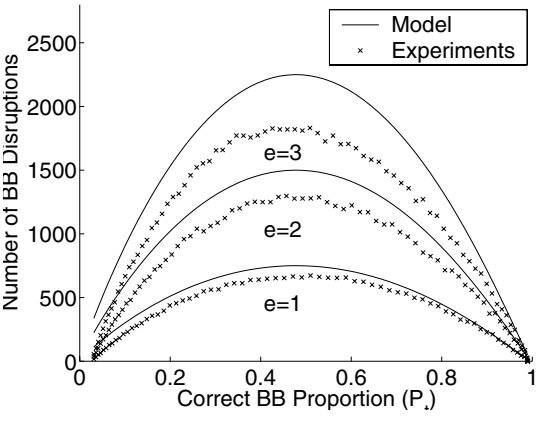

Fig. 1. Numbers of BB disruptions for the 5×100 MaxTrap. The BB disruption is severe in the middle of the GA run, and mild when the GA is merely or nearly converged.

It is easily seen that BB disruption is severe when roughly half of the BBs in the population are correct; Only few BB disruptions occur when the proportion of correct BBs is close to either 0 or 1. The observation suggests the following possible adaptive speedup scheme. Instead of recalculating the linkage model every generation, the linkage model is only updated every several generations at the beginning and the end of the GA run. Of course, it is non-trivial to estimate the degree of convergence of the GA for any given generation, and that leaves a room for the future work.

The time to convergence for different linkage model error e is shown in Figures 2 and 3. The convergence condition is that on average, there are $(m-1)$ correct BBs in each individual, where m is the number of BBs of the problem. As expected, since the number of BB disruptions is overestimated, the predicted time to convergence is also longer than the actual number of generations that the GA needs to converge. For a smaller e (compared with $\sqrt{m}$), the model predicts better. The empirical data agree better with the model for the 5×100 MaxTrap than the the 5×50 one, because for the same value of error, the error is relatively smaller compared to $\sqrt{m}$ for a larger m.

Finally, Equation 12 predicts that the GA could hardly converge for $e > 2I'\sqrt{m}$. In the experiments, because a pair-wise crossover is used and the swap is performed with a probability 0.5 on every cross-site, the number of BB dis-

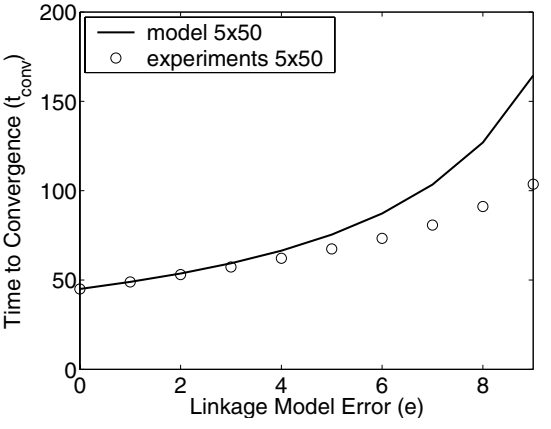

Fig. 2. Time to convergence for the 5x50 MaxTrap. The model overestimates the time to convergence. For a relatively smaller e, the model predicts better.

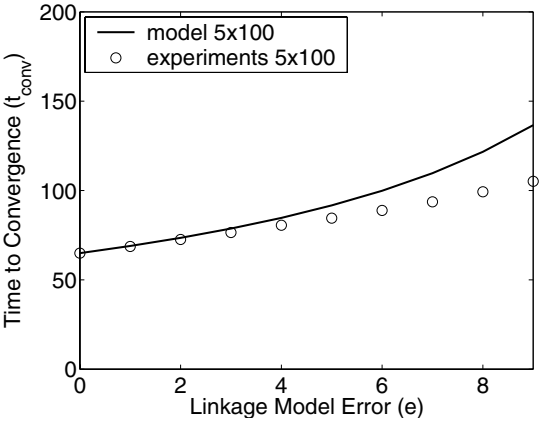

Fig. 3. Time-to-convergence for the 5x100 MaxTrap. The model predicts better than that for 5x50 because then number of errors e is relatively smaller compared with $\sqrt{m}$.

ruptions is only half as modelled in section 2, and hence $e_{critial} = 4I'\sqrt{m}$. If the linkage model contains more errors than $e_{critial}$, BB disruption rate is higher than BB growth rate, and the GA is difficult to converge. The critical number of errors versus the number of BBs is plotted in Figure 4. As shown in the figure, the results basically agree with the model: The critical number of error $e_{critical}$ grows proportionally to the square root of the number of BBs ($\sqrt{m}$). Since the number of BB disruptions is overestimated, the estimation of $e_{critical}$ should be a underestimation. However, due to the finite population size, the empirical $e_{critical}$ is smaller than predicted.

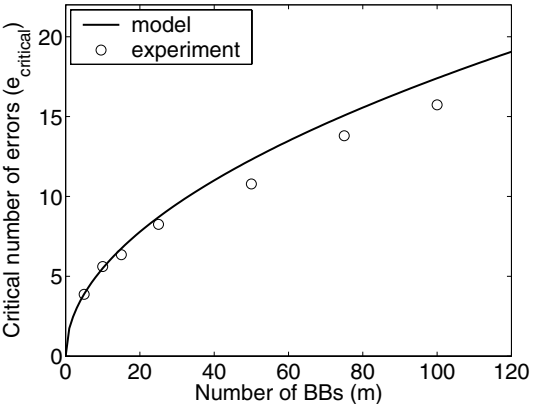

Fig. 4. The critical number of errors versus the number of BBs. Roughly $e_{critical} = O(\sqrt{m})$.

5 Conclusions and Future Work

In this paper, by assuming a infinite population size and perfect mixing, the time to convergence was derived for a given number of misidentified BB of the linkage model. The derivations gave several insights about the model-building for GAs:

1. The BB disruptions is severe when roughly half of the BBs are converged. The BB disruptions are mild when the GA is merely or fully converged.

2. Speedup might be achieved by applying evaluation relaxation techniques on the model-building techniques at the beginning and at the end of the GA run.

3. The critical number of errors (the largest number of BBs that a linkage model could misidentify while a $(m-1)$-BB convergence is still possible) grows proportional to the square root of the number of BBs ($e_{critical} = O(\sqrt{m})$).

As future work, we would like to integrate the the first type of errors discussed in section 2.1 into our models. Also, we are investigating how to estimate the number of errors of a linkage model. The number of errors might be estimated by observing the convergence behavior of the GA. If that is doable, we could perform the linkage-identification algorithm only when the number of errors exceeds some predefined threshold. By doing that, a speedup is obtained while the quality of solution is maintained.

Acknowledgment. This work was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant F49620-

03-1-0129, and by the Technology Research, Education, and Commercialization Center (TRECC), at University of Illinois at Urbana-Champaign, administered by the National Center for Supercomputing Applications (NCSA) and funded by the Office of Naval Research under grant N00014-01-1-0175. 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 Technology Research, Education, and Commercialization Center, the Office of Naval Research, or the U.S. Government.

References

1. Holland, J.H.: Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor, MI (1975)
2. Goldberg, D.E.: The design of innovation: Lessons from and for competent genetic algorithms. Kluwer Academic Publishers, Boston, MA (2002)
3. Harik, G.R., Goldberg, D.E.: Learning linkage. Foundations of Genetic Algorithms 4 (1996) 247–262
4. Munetomo, M., Goldberg, D.E.: Identifying linkage groups by nonlinearity/nonmonotonicity detection. Proceedings of the Genetic and Evolutionary Computation Conference 1999: Volume 1 (1999) 433–440
5. Pelikan, M., Goldberg, D.E., Lobo, F.G.: A survey of optimization by building and using probabilistic models. IlliGAL Report No. 99018, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, IL (1999)
6. Larrañaga, P., Lozano, J.A., eds.: Estimation of distribution algorithms: A new tool for evolutionary computation. Kluwer Academic Publishers, Boston, MA (2002)
7. Yu, T.-L., Goldberg, D.E., Yassine, A., Chen, Y.-p.: Genetic algorithm design inspired by organizational theory: Pilot study of a dependency structure matrix driven genetic algorithm. Proceedings of Artificial Neural Networks in Engineering 2003 (ANNIE 2003) (2003) 327–332 (Also IlliGAL Report No. 2003007).
8. Ocenasek, J., Schwarz, J., Pelikan, M.: Design of multithreaded estimation of distribution algorithms. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2003) (2003) 1247–1258
9. Sastry, K.: Evaluation-relaxation schemes for genetic and evolutionary algorithms. Master thesis, University of Illinois at Urbana-Champaign, Urbana, IL (2002)
10. Pelikan, M.: Bayesian optimization algorithm: From single level to hierarchy. Doctoral dissertation, University of Illinois at Urbana-Champaign (2002)
11. Smith, J., Fogarty, T.C.: Recombination strategy adaptation via evolution of gene linkage. Proceedings of the 1996 IEEE International Conference on Evolutionary Computation (1996) 826–831
12. Harik, G.R., Cantú-Paz, E., Goldberg, D.E., Miller, B.L.: The gambler's ruin problem, genetic algorithms, and the sizing of populations. Proceedings of 1997 IEEE International Conference on Evolutionary Computation (1997) 7–12
13. Harik, G.R.: Linkage learning via probabilistic modeling in the ECGA. IlliGAL Report No. 99010, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, IL (1999)

14. Mühlenbein, H., Schlierkamp-Voosen, D.: Predictive models for the breeder genetic algorithm: I. Continuous parameter optimization. Evolutionary Computation **1** (1993) 25–49
15. Thierens, D., Goldberg, D.E.: Convergence models of genetic algorithm selection schemes. In: Parallel Problem Solving fron Nature, PPSN III. (1994) 119–129
16. Miller, B.L.: Noise, sampling, and efficient genetic algorithms. Doctoral dissertation, University of Illinois at Urbana-Champaign, Urbana (1997)
17. Deb, K., Goldberg, D.E.: Analyzing deception in trap functions. Foundations of Genetic Algorithms 2 (1993) 93–108
18. Blickle, T., Thiele, L.: A mathematical analysis of tournament selection. Proceedings of the Sixth International Conference on Genetic Algorithms (1995) 9–16