

**How Well Does A Single-Point Crossover
Mix Building Blocks with Tight Linkage?**

**Kumara Sastry
David E. Goldberg**

IlliGAL Report No. 2002013
May, 2002

Illinois Genetic Algorithms Laboratory (IlliGAL)
Department of General Engineering
University of Illinois at Urbana-Champaign
117 Transportation Building
104 S. Mathews Avenue, Urbana, IL 61801
<http://www-illigal.ge.uiuc.edu>

How Well Does A Single-Point Crossover Mix Building Blocks with Tight Linkage?

Kumara Sastry 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
{ksastry,deg}@uiuc.edu

Abstract

Ensuring building-block (BB) mixing is critical to the success of genetic and evolutionary algorithms. This study develops facetwise models to predict the BB mixing time and the population sizing dictated by BB mixing for single-point crossover. Empirical results are used to validate these models. The population-sizing model suggests that for moderate-to-large problems, BB mixing—instead of BB decision making and BB supply—bounds the population size required to obtain a solution of constant quality. Furthermore, the population sizing for single-point crossover scales as $O(2^k m^{1.5})$, where k is the BB size and m is the number of BBs.

1 Introduction

Since the inception of genetic algorithms (GAs), the importance of building blocks have been recognized (Holland, 1975; Goldberg, 1989b). Based on Holland’s notion of BBs, Goldberg proposed a design decomposition method for a successful design of GAs (Goldberg, 1991; Goldberg & Liepens, 1991; Goldberg, Deb, & Clark, 1992). This design decomposition currently consists of seven steps (Goldberg, in press) and can be stated as follows: (1) Know what GAs process—building blocks (BBs), (2) solve problems that are of bounded BB difficulty, (3) ensure an adequate supply of raw BBs, (4) ensure increased market share for superior BBs, (5) know BB takeover and convergence times, (6) ensure that BB decisions are well made, and (7) ensure a good mixing of BBs. Significant progress has been made in developing facetwise models for many of the above decomposition steps and the interested reader should consult Goldberg (in press) and the references therein for further details.

However, researchers have often overlooked the issues of BB identification and mixing/exchange, even though studies on selectorecombinative GAs have indicated that effective *identification* and *exchange* of BBs is critical to innovative success. Furthermore, existing facetwise models such as convergence-time and population-sizing models *assume* tight linkage. That is, alleles of a BB were assumed to be close to one another, and crossover operators are assumed to ensure necessary exchange of BBs with high probability. Even though, the assumption of tight linkage isolates the phenomenon of interest while bracketing the linkage problem, in real-world problems this is not the case, as we don’t know which alleles contribute to which BBs *a priori*. This necessitates the incorporation of mixing of BBs into GA dynamics.

It is therefore critical to understand mixing capability of popular recombination operators used in genetic and evolutionary algorithms. Dimensional models for BB mixing have been developed

for uniform crossover (Goldberg, Thierens, & Deb, 1993; Thierens & Goldberg, 1993) and similar analysis is yet to be done on single-point and two-point crossovers. For problems with loosely linked BBs, mixing behavior of multi-point crossover is bounded by the mixing model of uniform crossover. On the other hand, for problems with tightly linked BBs, mixing behavior of multi-point crossovers can be different from that of uniform crossover and separate dimensional models have to be developed to predict their BB mixing rates. Therefore the objective of this study is to develop a facetwise model to predict the mixing behavior of single-point crossover and to utilize this model to predict the mixing time and population sizing dictated by BB mixing.

This paper is organized as follows: The following section presents a brief literature review of studies on recombination-operator analysis. Section 3 states the assumptions used in developing the facetwise models. Section 4 defines the mixing problem and section 5 presents a BB mixing model for a problem with two building blocks. The results of section 5 are extended to m building blocks in section 6. The mixing models are used in conjunction with takeover-time and convergence-time models to develop a population-sizing model bounded by BB mixing in each of these two sections. Finally, section 7 summarizes the paper and presents key conclusions.

2 Literature Review

Holland made two key observations in his monograph (Holland, 1975): (1) BBs with tighter linkage (shorter defining length) have a selective advantage over those with loose linkage (longer defining length), and (2) operators to adapt linkage and to choose allele combinations might be necessary for GA success. Goldberg, Deb, and Clark (1992) proposed the design decomposition for a successful design of GAs, a core part of which is BB identification and mixing. However, success stories of GAs with fixed codings and crossover operators have somewhat masked the importance of investigating and designing operators that adapt linkage and efficiently mix BBs (Goldberg, in press).

On a parallel note, it is now known that many operators such as *elitism*, *niching*, *mating restriction*, *inversion*, and *reordering*, proposed to adapt linkage and to enhance GA performance do not achieve their goal. Thierens (1995) showed that elitism and niching either alone or in cohort do not address the mixing issue effectively. He also suggested that any crossover with fixed scheme is highly unlikely to adapt linkage. Bagley (1967) and Frantz (1972) provided empirical evidence to indicate that Holland's (1975) inversion operator does not learn linkage effectively. Goldberg and Bridges (1990) demonstrated that simple reordering operators have limited linkage-learning capability.

Failures with such operators suggest that the critical step in designing better crossover operators is to first analyze fixed crossover operators such as uniform, one-point, and n-point crossovers. Such analysis should tell us if fixed crossover operators are powerful enough to solve problems of bounded difficulty quickly, reliably, and accurately. They should also give us insights as to why such fixed crossover operators fail if they do fail. The question as to how well fixed crossover operators solve GA-easy as well as GA-hard problems is called the mixing question.

Many researchers have analyzed different aspects of fixed point crossovers. One such class of models is motivated from the schema theorem (Holland, 1975). These studies model schemata disruption of different crossover operators and are useful in comparing those operators (De Jong, 1975; Syswerda, 1989; Spears & De Jong, 1991; De Jong & Spears, 1990; De Jong & Spears, 1992). However, as shown elsewhere, these models do not address the mixing problem (Goldberg & Sastry, 2001). It should be noted that schema theorem only focuses on a single BB, but for a GA success (for innovation and successful recombination of best BBs), we must model some aspects about the likelihood of different BBs coming together.

Another class of models is motivated from quantitative genetics (Bailey, 1961; Bulmer, 1985; Falconer, 1989; McPeck, 1996). Central to these models is *Geiringer's theorem* (Geiringer, 1944) and *linkage (Robbin's) equilibrium* (Robbins, 1918). Such models quantify either *linkage disequilibrium* (distance from the linkage equilibrium) (Christiansen, 1989; Booker, 1993), or crossover-induced bias such as *positional bias*, *distributional bias*, *recombinative bias* and *schema bias* (Eshelman, Caruana, & Schaffer, 1989; Eshelman, 1991; Booker, 1993; Eshelman & Schaffer, 1995; Altenberg, 1995; Rana, 1999), and *relaxation time*, which measures the rate at which recombination operators converge to linkage equilibrium (Rabani, Rabinovich, & Sinclair, 1998; Prügel-Bennett, 2001). However, similar to schema disruption models, these models also do not address the mixing issue.

Unlike the above two class of models, another class of models combines all the facets of recombination operators (Bridges & Goldberg, 1987; Vose & Liepens, 1991; Nix & Vose, 1992; Prügel-Bennett & Shapiro, 1994; Rattray, 1996; Stephens & Waelbroeck, 1999) using tools such as difference equations, differential equations, Markov chains, and statistical mechanics. They tend to model workings of recombination exactly or more accurately than facetwise models. That is, they contain all the information about BB mixing that we are interested in. Such information is usually hidden among many other facets of GAs, making these models highly complex. This makes it very hard to extract information on BB mixing (or for that matter, any other single aspect of GAs) out of such models. Unless these models can be simplified, or their asymptotic behavior predicted, it is very difficult to use them to design better GAs or better operators.

Finally, there exist models that address the BB mixing issue in a more direct manner than those belonging to the above mentioned models (Goldberg, Thierens, & Deb, 1993; Thierens & Goldberg, 1993). These models predict *mixing* or *innovation* time. Here, mixing time is defined as the expected number of generations to obtain an instance of the target string. Such models are not only intuitive, but also easy to analyze and compare with other facetwise models using dimensional arguments. Such a comparison can lead to the construction of a *control map* clearly identifying different competing forces affecting the genetic search. Since these studies deal more directly with BB mixing issue, their key results are presented in the following paragraphs .

Goldberg, Thierens, and Deb (1993) addressed the issue of allele-wise mixing building upon Holland's suggestion that limiting probability of success of BB exchange is allele-wise random assortive (Holland, 1975). Goldberg, Thierens, and Deb (1993) calculated the mixing time, t_x , for allele-wise exchange, assuming a population of size n , geometric mean of proportion of target allele p , crossover probability p_c , and a string length ℓ :

$$t_x = \frac{1}{np_c p^\ell} \quad (1)$$

They compared this model with selection time model (Goldberg & Deb, 1991), drift time (Goldberg & Segrest, 1987) and cross-competition models to construct mixing boundary, drift boundary and cross-competition boundary. They identified the following key points in computing these boundaries:

- When mixing time is less than selection time, then good BB exchange is ensured.
- On the other hand, if mixing time is greater than selection time, then a GA will converge prematurely
- When the selection time is greater than the drift time then, GA drifts and converges to alleles with little or no selection pressure
- When selection pressure is high, BBs in different partitions might compete. Such a competition between non-competing alleles is called *cross competition*.

Goldberg, Thierens, and Deb (1993) used these boundaries to construct a control map on the crossover probability versus selection pressure ($p_c - s$) plane. They also verified their facetwise models and the control map with empirical evidence. They observed that simple GAs solving easy problems have a large sweet-spot (Goldberg, 1999). As a consequence of a large sweet-spot, a simple GA will successfully solve easy problems with a large range of parameters values to choose from.

The allele-wise mixing model of Goldberg, Thierens, and Deb (1993) was further expanded and used for GA design by Thierens (Thierens, 1995; Thierens, 1996). Specifically, he quantified the interplay between (1) string length and selection pressure, (2) disruption probability and string length, (3) population size and disruption probability, and (4) population size and string length. He showed that for a GA easy problem, the effect of mixing on population size dominates the effect of selection for small string lengths. For larger problems (longer string lengths), the effect of selection on population sizing dominates the effect of mixing.

Thierens and Goldberg (1993) extended the allele-wise mixing model to incorporate BB wise mixing. One of their objectives was to investigate if the sweet-spot remained large enough for boundedly difficult problems. Specifically, they considered *deceptive* trap problems with uniform building-block scale and size (Goldberg, 1987; Goldberg, 1989a; Deb & Goldberg, 1993). They developed a mixing-time model for the case of two BBs of size k and then extended it for the case of multiple BBs of size k . For a problem with m building blocks of size k , the mixing time is given by (Thierens & Goldberg, 1993; Thierens, 1995; Thierens, 1999),

$$t_x = c \frac{2^{\mu k} 2^m}{np_c m^{\frac{5}{2}}}, \quad (2)$$

where c and μ are constants. Furthermore, comparing this with selection time yields a population-sizing model to satisfy mixing:

$$n \ln n > c \frac{2^{\mu k} 2^m}{np_c m^{\frac{5}{2}}} \ln s. \quad (3)$$

The above result clearly indicates that the sweet-spot shrinks rapidly as the problem size increases. Their model also showed that for a GA success, the BBs have to be tightly linked in the problem coding structure. One of the main outcome of their analysis is that they provided sufficient evidence to indicate that, to solve boundedly difficult problems in polynomial time, operators that adapt linkage are required. In other words, as the problems become more difficult, population sizes must grow exponentially to ensure that a simple GA using fixed crossover operators converges to a good solution (Thierens, 1995; Thierens, 1999). This important result has led to the design of many successful designs of operators that identify and adapt linkage (Goldberg, in press).

Both allele-wise mixing models and the BB-wise mixing models are derived for uniform crossover and similar models for multi-point crossover is still lacking. For problems with loosely linked BBs, the mixing models for uniform crossover bounds the mixing behavior of multi-point crossover. However, for problems with tightly linked BBs, the scenario might be completely different and further investigation is necessary. This paper takes one step towards achieving this goal, where mixing behavior of single-point crossover on problems with tightly linked BBs is analyzed. Before developing the BB-wise mixing models, key assumptions made throughout this study is presented.

3 Assumptions

The primary assumptions used in this study are as follows. Many of these assumptions are made to ease the analytical burden and can be relaxed in a straightforward manner.

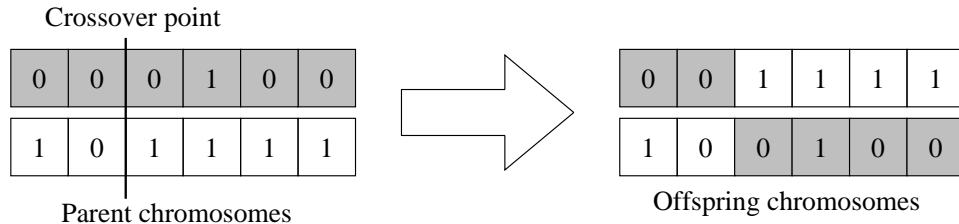


Figure 1: Illustration of a single-point crossover. A crossover site from the possible $\ell - 1$ sites is chosen randomly and alleles to the right of the crossover site are swapped to yield two offspring.

- **Selectorecombinative GAs:** Selectorecombinative means that only selection and recombination are used as the genetic operators and mutation is not used. Either truncation selection (Mühlenbein & Schlierkamp-Voosen, 1993) or binary tournament selection without replacement (Goldberg, Korb, & Deb, 1989) is employed. Single-point crossover is used as the recombination method. In single-point crossover, individuals in the population are paired randomly and recombination is performed on each pair with a probability equal to the crossover probability, p_c , to obtain two new offspring. A single crossover site is selected at random over the string length, and the alleles on the right side of the site are exchanged between the individuals (see figure 1).
- **Generational GAs:** In generationwise GAs, evolution occurs through discrete steps. In any given generation, all the individuals in the population are evaluated. The selection operator selects highly fit individuals and places them in a mating pool. Individuals in the mating pool are recombined to create new individuals which then replace the original population and the next generation begins.
- **Non-overlapping population of fixed size:** The number of individuals throughout the genetic search is kept constant. Furthermore, offspring population completely replaces parent population (for example, elitism is not used).
- **Fixed-length binary encoding:** The decision variables of the search problem are assumed to be encoded in a fixed-length binary string.
- **Uniformly scaled building blocks with tight linkage:** In this study, we restrict the notion of building block as the minimal-order schemata that contribute to the global optimum. Uniform scaling implies that the contribution of building blocks from different partitions to the overall fitness of the individual is the same. A tight linkage implies that alleles belonging to the same partition are positioned next to each other in a string. As mentioned earlier, single-point crossover behaves similar to uniform crossover on problems with loose linkage and the existing mixing model can be used to bound its behavior. In this study, however, we are interested in predicting the behavior of single-point crossover on problems with tight linkage.
- **Deceptive trap function:** The fitness function for m k -bit deceptive traps is defined as follows (Goldberg, 1987; Goldberg, 1989a; Deb & Goldberg, 1993):

$$f = \sum_{i=1}^m g(u_i), \quad (4)$$

where,

$$g(u_i) = \begin{cases} 1 & u_i = k, \\ (1.0 - \delta) \left(1 - \frac{u_i}{k-1}\right) & u_i < k \end{cases}, \quad (5)$$

and,

$$u_i = \sum_{j=1}^k x_{(i-1)k+j}, \quad (6)$$

where, x_i is the value of the i th allele, δ is the signal difference and is a constant. The Lower the value of δ , the more difficult the problem is for a GA. In this study the value of δ is set to 0.25.

Even though this study uses deceptive trap functions to validate the model, the results should apply to additively decomposable stationary fitness functions of bounded order (Goldberg, in press).

The aforementioned assumptions ease the analytical burden, and facilitate the development of facetwise models for mixing. Furthermore, a dimensionless argument in developing these models will enable us to develop algebraic relationships among key control parameters. In most cases, the models also provide a simple design methodology for practitioners. The motivation for developing such facetwise models is discussed elsewhere (Goldberg, in press).

4 Problem Definition

The section describes the problem addressed in this paper. Before doing so, it should be noted that there are two ways a crossover can increase the number of BBs on a particular string. One possibility is that the BB at a certain position is *created*. However, the likelihood of BBs being created reduces as the BB size increases. The other possibility is that the BBs are *mixed*. That is, the crossover operator transfers BBs at different positions from both parents to one of the offspring. For instance, a crossover can combine the following two strings, **bb####** and **##b###**, to yield following two offspring, **bbb###**, and **#####**. Here **b** refers to the BB and **#** refers to schemata other than the BB.

This aspect of crossover—its ability to recombine BBs in order to create better solutions—is of particular interest to us. Particularly, the rate at which a crossover exchanges BBs dictate the success of a GA run. Therefore, in this paper, we model the mixing rate of single-point crossover on search problems with tightly linked BBs. Specifically, we answer the following question: Given that the individuals in the population have m_c BBs, how long—in terms of number of generations—will it take to obtain individuals with $m_c + 1$ BBs. This time is defined as the *mixing time* and is denoted by t_x . To do so, we first consider the case of two BBs, and later extend it to the general m BB case in the next section. The dimensional model for mixing time will be used in conjunction with convergence-time model to develop a population-sizing model dictated by BB mixing.

5 Mixing Models: Two Building Blocks

The section presents dimensional models to predict the mixing behavior of single-point crossover on a problem with two building blocks. Here, we assume that individuals in the initial population have exactly one BB. Therefore, in the two-BB case, a single mixing event mixing two BBs from different partitions to yield the global optimum. That is, if the following two individuals are selected for crossover: **b#** and **#b**, and the crossover site is chosen as k , then one of the offspring is **bb**, which

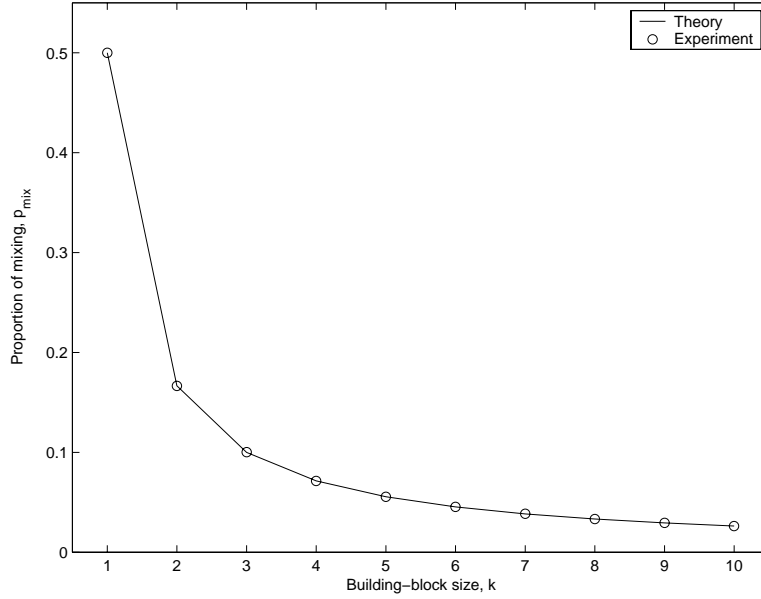


Figure 2: Empirical validation of the proportion of mixing model (Equation 7). Proportion of mixing for two BBs is plotted as a function of BB size, k . The empirical results are averaged over 5000 independent runs.

is the global optimum. Recognizing that a mixing event can occur in two scenarios ($\mathbf{b\#+\#b}$, or $\mathbf{\#b+b\#}$) out of the possible four crossover scenarios and if the crossover site is chosen between the BBs (crossover site is k), we can write probability of a mixing event as

$$p_{\text{mix}} = \left(\frac{2}{4}\right) \left(\frac{1}{2k-1}\right). \quad (7)$$

The above equation is compared to empirical results in figure 2 which plots the probability of mixing as a function of BB size, k . Figure 2 clearly validates the model (equation 7).

Equation 7 assumes that every individual in the population contains either a BB (all ones) or a deceptive attractor (all zeros). However, in the actual case, the probability of having a BB in a string is 2^{-k} . Therefore, the proportion of recombinations that can result in a mixing event is given by $2^{-2k} (2^k - 1)$. In such a case probability of mixing is given by

$$p'_{\text{mix}} = 2 \cdot 2^{-2k} (2^k - 1) \left(\frac{1}{2k-1}\right) \approx \frac{2^{-(k-1)}}{2k-1}. \quad (8)$$

Assuming a population size, n and a crossover probability, p_c , there are $\frac{n}{2} \cdot p_c$ recombinations in one generation. Therefore, them mixing time, t_x is

$$t_x = \frac{1}{\frac{n}{2} p_c p'_{\text{mix}}} = \frac{2^{k-1} (2k-1)}{n p_c}. \quad (9)$$

The above mixing time model is compared with the following convergence-time model (Mühlenbein & Schlierkamp-Voosen, 1993) to interrelated recombination with selection time:

$$t_{\text{conv}} = \frac{c_c \sqrt{\ell}}{I} \quad (10)$$

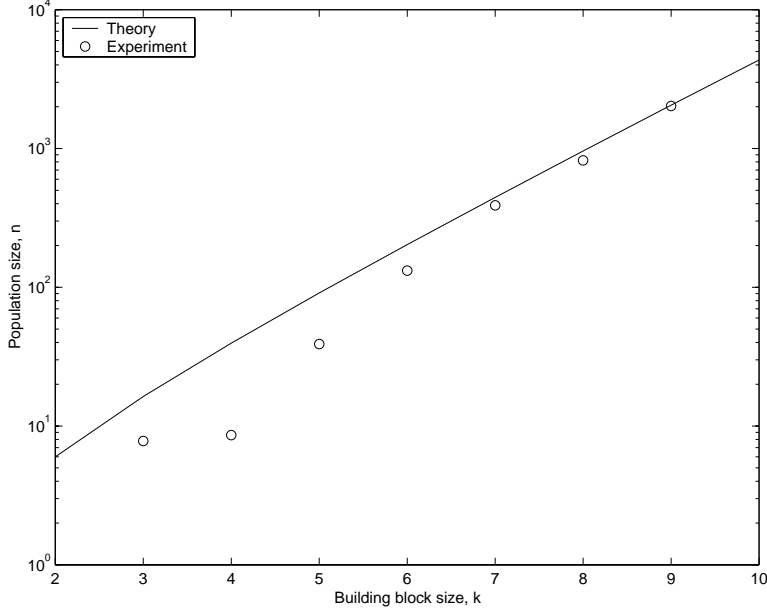


Figure 3: Empirical validation of the population-sizing model dictated by BB mixing (Equation 11). Minimum population size required for a GA success is plotted as a function of BB size, k . The minimum population size is determined by a bisection method and the empirical results are averaged over 25 independent bisection runs.

where, I is the selection intensity, which is a function of tournament size (Bäck, 1994), and c_c is a constant. Note that Goldberg, Thierens, and Deb (1993) and Thierens and Goldberg (1993) used takeover-time model (Goldberg & Deb, 1991), however both approaches (using convergence-time and takeover-time models) are equally valid. Recognizing that for innovative success the mixing time has to be less than the convergence time (recombination has to create the global optimum before the population converges to the local optimum), we can write,

$$t_x < t_{\text{conv}}$$

For the two-BB case, $\ell = 2k$, and the above equation can be written as

$$n > \frac{I}{c_c p_c} \left(\frac{2^{k-1}(2k-1)}{\sqrt{2k}} \right) \approx c_{x,2} \frac{I}{p_c} (2^k \sqrt{k}), \quad (11)$$

where $c_{x,2} = (\sqrt{2}c_c)^{-1}$ is a constant. Equation 11 predicts the population size dictated by BB mixing. It is verified with empirical results in figure 3. The empirical results are obtained as follows: A binary tournament selection and single-point crossover with $p_c = 1$ is used. A GA run is terminated when all the individuals in the population converge to the same fitness value. The average number of BBs correctly converged are computed over 50 independent runs. The minimum population required such that both BBs converge to the correct value is determined by a bisection method. The results shown in figure 3 are averaged over 25 such bisection runs.

The population-sizing model suggest that the population size increases exponentially with the BB size, k , linearly with the selection intensity, I , and inversely proportional to the crossover probability, p_c . This result is similar in functional form to the gambler's ruin based population-sizing model that accounts for BB decision making and initial BB supply (Harik, Cantú-Paz, Goldberg, & Miller, 1997).

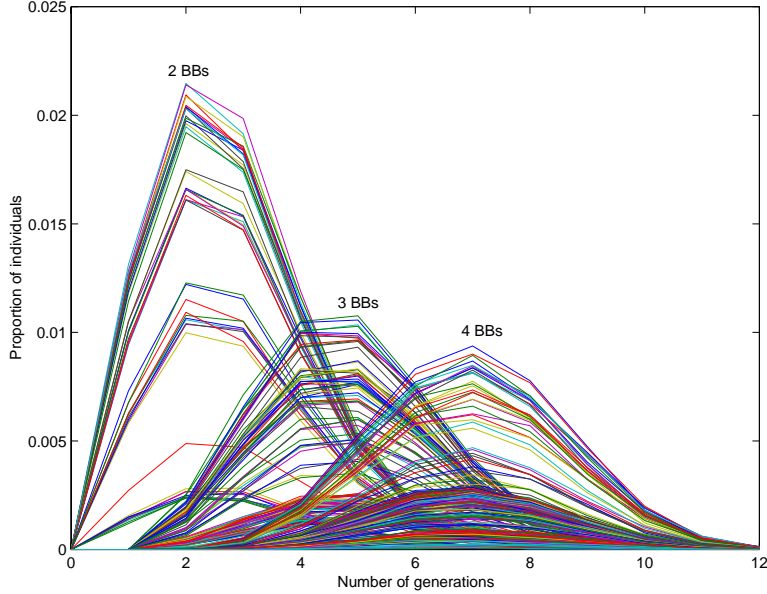


Figure 4: Proportion of individuals with correct BBs as a function of time. Empirical results indicate that mixing at a higher level ($m_c + 1$) takes over only after BBs of lower level (m_c) are covered. The results are for ten 4-bit deceptive trap functions

Equipped with the mixing models for the two BB case, we are now ready to predict them for several building block case, which is presented in the next section.

6 Mixing Models: m Building Blocks

In the previous section, the simplest possible mixing process was considered: The exchange of two building blocks. To extend it the analysis to m building blocks, we blend empirical observations with theory similar to the approach followed by Thierens and Goldberg (1993). The empirical results on m k -bit deceptive traps indicates two key behavior of single-point crossover:

1. All the building block configurations at a certain *mixing level* have to be discovered before a higher mixing level can be reached. Here a mixing level denotes the number of BBs present in an individual string. For instance an individual **b#b###** is at mixing level 2, and an individual **b#bb##** is at mixing level 3. This *ladder-climbing* phenomenon is illustrated in figure 4. The figure plots the proportion of all possible BB configuration at different mixing levels as a function of time. It can be clearly seen that when a certain mixing level is at its maximum value—when the proportion of all the BB configurations at a certain mixing level in the population is highest—proportion of individuals at higher mixing level is negligible. This phenomenon enables us to use the *ladder-climbing* model similar to the one used by Thierens and Goldberg (1993).
2. The proportion of individuals having good BB at their two ends is higher than those at other positions. In fact, the proportion gradually reduces along the position and reaches a minimum at the center. This phenomenon is clearly illustrated in figure 5, which plots the proportion of individuals having a good BB as a function of its position in the string and time. This is due to the fact that the probability of mixing is much higher when mixing two BBs at the

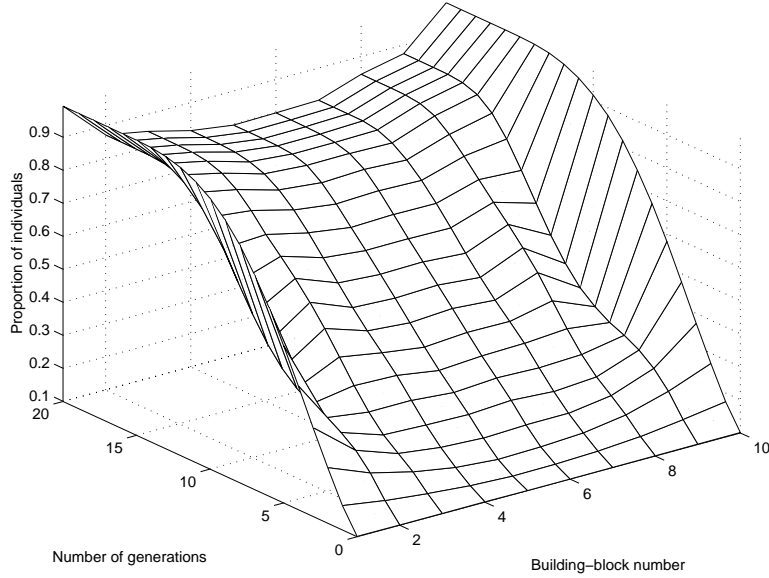


Figure 5: Proportion of individuals with correct BBs as a function of BB position and time. Mixing of one-point crossover is dependent on the BB position. It is highest on the string ends and lowest in the middle of the string. The results shown are for a 10 4-bit deceptive trap functions.

extreme points in the string. For example, the mixing probability when recombining $\mathbf{b\#\dots\#\#}$ and $\mathbf{\#\#\dots\#b}$ to yield $\mathbf{b\#\dots\#b}$ is $((m-2)k+1)/(mk-1)$ as opposed to recombining $\mathbf{b\#\#\dots\#}$ and $\mathbf{\#b\#\dots\#}$ to yield $\mathbf{bb\#\dots\#}$, in which case the mixing probability is $1/(mk-1)$.

This suggests increase in mixing level leads to the decrease in problem size affecting mixing. For example, when we are at mixing level 3, the problem size reduces by two BBs. This is because the majority of the individuals have two BBs at their ends and the third BB can lie anywhere in $m-2$ possible positions. Therefore, for a problem with m BBs at mixing level 3, the mixing events leading to mixing level 4 are equivalent to those of a problem with $m-2$ BBs at level 1 leading to level 2. Similarly, when we are at mixing level $m-1$, only 2 BBs in the middle of the string are not converged and contribute to mixing. This is similar to the two-BB case considered in the last section.

This property of single-point crossover called *length-reduction* phenomenon differs from that of uniform crossover.

The combined effect of *ladder-climbing* and *length reduction* illustrated in figure 6. The length-reduction phenomenon suggests that we can consider mixing probability as a function of number of BBs that are not converged at a given time and the BB size. Furthermore, we assume that among the BBs that are not converged the mixing level is always one.

Given that a problem has m BBs and we are currently at mixing level one, that is, every individual has one BB at some position. There are m^2 total recombination scenarios possible, out of which the following scenarios result in mixing:

Recombination scenario	$\mathbf{b\#\#\dots\#\#}$	$\mathbf{b\#\#\#\dots\#}$	\dots	$\mathbf{b\#\dots\#\#\dots\#}$	\dots	$\mathbf{b\#\dots\#\#}$
	$\mathbf{\#b\#\dots\#\#}$	$\mathbf{\#\#\b\#\dots\#}$	\dots	$\mathbf{\#\#\dots\#b\#\dots\#}$	\dots	$\mathbf{\#\#\dots\#b}$
Possible events	$2(m-1)$	$2(m-2)$	\dots	$2(m-i)$	\dots	2
Mixing Probability	$\frac{1}{mk-1}$	$\frac{k+1}{mk-1}$	\dots	$\frac{(i-1)k+1}{mk-1}$	\dots	$\frac{(m-2)k+1}{mk-1}$

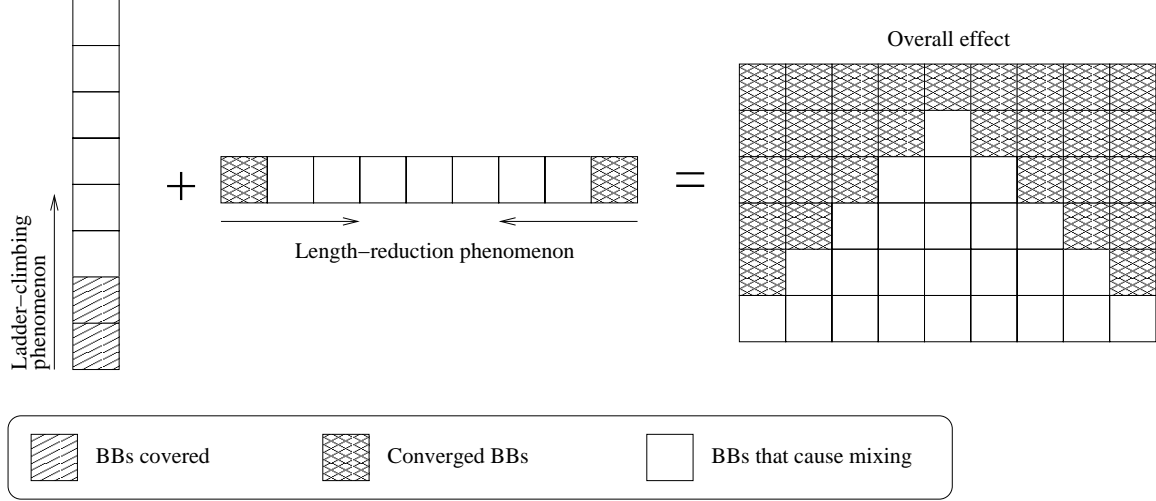


Figure 6: Illustration of the *ladder-climbing* and *length-reduction* phenomena and their combined effect. Ladder-climbing phenomenon suggests that we increase the mixing level one step at a time. Length-reduction phenomenon suggests that as the mixing level increases, the problem size dictating mixing shrinks. It also suggests that the string ends converge faster than the BBs at the middle of the string.

Therefore the overall mixing probability can be written as

$$p_{\text{mix}}(m) = \frac{2}{m^2} \sum_{i=1}^{m-1} \frac{2(m-i)((i-1)k+1)}{mk-1}, \quad (12)$$

where the factor of 2 comes because the mixing event can occur in either of the two strings. Using the following two properties

$$\sum_{i=1}^{m-1} i = \frac{1}{2}m(m-1), \quad \text{and} \quad \sum_{i=1}^{m-1} i^2 = \frac{1}{6}m(m-1)(2m-1),$$

in equation 12 can be simplified as follows:

$$p_{\text{mix}}(m) = \frac{2}{3} \cdot \frac{(m-1)[(m-2)k+3]}{m(mk-1)} \quad (13)$$

From the length-reduction phenomenon, we know that, at a given mixing level, m_c , the mixing probability is given by

$$p_{\text{mix}}|_{m_c} = p_{\text{mix}}(m - m_c + 1) = \frac{2}{3} \left[\frac{(m - m_c)((m - m_c - 1)k + 3)}{(m - m_c + 1)((m - m_c + 1)k - 1)} \right]. \quad (14)$$

For moderate-to-large problems, the mixing probability can be assumed to be constant with respect to the mixing level and can be approximated as follows:

$$p_{\text{mix}} \approx p_{\text{mix}}(m_c = 1) = \frac{2}{3} \left[\frac{(m-1)((m-2)k+3)}{m(mk-1)} \right]. \quad (15)$$

Comparison of equations 14, and 15 with empirical results is shown in figure 7. The figure plots the mixing probability as a function of mixing level. The results indicate that constant mixing

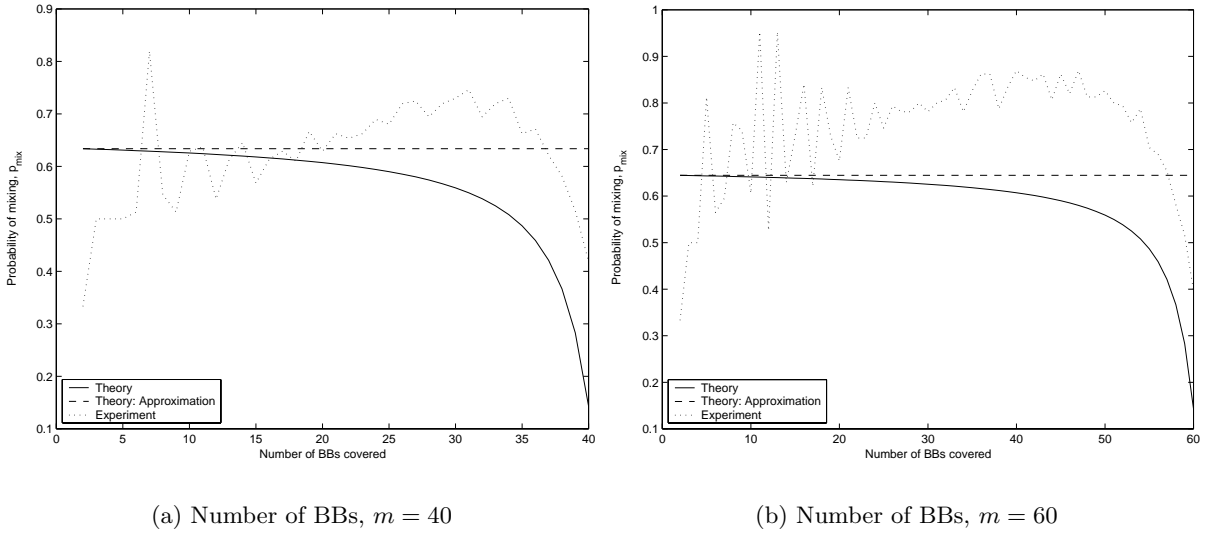


Figure 7: Comparison of proportion of mixing predicted by equations 14 and 15 with empirical results. The proportion of mixing is plotted as a function of number of covered BBs. The mixing proportion is approximated as constant with respect to the number of covered BBs.

probability assumption is similar to taking an averaging the mixing probability over different mixing level values. The empirical results are shown for a 4-bit deceptive trap with 40 and 60 BBs. The results are averaged over 500 independent runs.

Now the question remains as to how we reach mixing level $m_c + 1$ when we are at mixing level m_c . Assume that after n_x mixing events, we have n_x individuals at mixing level $m_c + 1$. The question to be answered is how many mixing events are needed to have all BBs covered at mixing level $m_c + 1$. Since all the n_x strings have $m_c + 1$ BBs, it can be expected that $m \left[1 - \left(1 - \frac{m_c + 1}{m} \right)^{n_x} \right]$ BBs are covered. By increasing the number of mixing events n_x , the probability that all BBs are covered approaches 1:

$$m \left[1 - \left(1 - \frac{m_c + 1}{m} \right)^{n_x} \right] > m(1 - \alpha), \quad (16)$$

where $\alpha < 1/m$. Rearranging the above equation and using the approximation, $\ln \left(1 - \frac{m_c + 1}{m} \right) \approx -\frac{m_c + 1}{m}$, we get

$$n_x > c_{mx} \frac{m}{m_c + 1}, \quad (17)$$

where $c_{mx} = -\ln \alpha$ is a constant.

The number of mixing events required to climb one step of the ladder therefore proportional to the number of building blocks. Similar to the mixing probability, we assume n_x to be constant with respect to the mixing level:

$$n_x \approx n_x (m_c = 1) = \frac{1}{2} c_{mx} m. \quad (18)$$

This approximation is slightly conservative, but bounds the mixing behavior of single-point crossover quite accurately as will become clear in the following paragraphs.

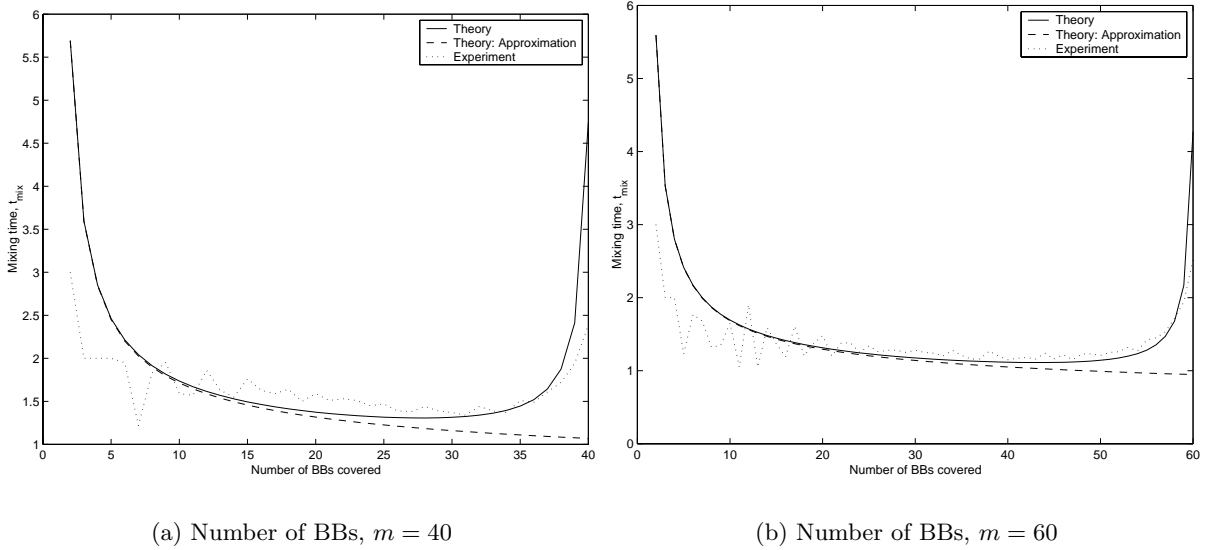


Figure 8: Comparison of mixing time, t_{mix} predicted by equations 19 and 20 with empirical results. The mixing time is plotted as a function of number of covered BBs (mixing level).

Recognizing that in a single generations, we have $\frac{n}{2}p_c$ recombinations, and using equations 15 and 18, we can calculate the mixing time, t_x :

$$t_x = \frac{n_x}{\frac{n}{2}p_cp_{\text{mix}}} = c_{x,m} \frac{m}{np_c} \left[\frac{(m-1)((m-2)k+3)}{m(mk-1)} \right], \quad (19)$$

where $c_{x,m} = \frac{2}{3}c_{mx}$ is a constant. The above equation can be approximately written as

$$t_x \approx c_{x,m} \frac{m}{np_c}. \quad (20)$$

The mixing times predicted by equations 19 and 20 are validated by empirical results in figure 8. The empirical results are for a 4-bit deceptive trap function with 40 and 60 BBs and are averaged over 500 independent runs. Figure 8 plots the mixing time, t_x as a function of mixing level. The results show that the agreement between the model and experiments agree better with the number of BBs.

Similar to the approach used in section 5, we want to compare the mixing time with convergence time. Here, we have to recognize that the convergence time refers to the total duration of a GA run. Therefore, unlike in section 5, we have to compute the total mixing time, $t_{x,\text{tot}}$, that is, the time required to go from mixing level 1 to mixing level m . Note that for the two-BB case the total mixing time an the mixing time for climbing one step of the ladder are the same. There are $m-1$ mixing levels between mixing level 1 and m . Approximating, $m-1 \approx m$, the total mixing time is

$$t_{x,\text{tot}} = mt_x = c_{x,m} \frac{m^2}{np_c}. \quad (21)$$

The model suggests that the total mixing time grows quadratically with the number of building blocks and is inversely proportional to the population size and crossover probability.

It should be noted here that if we want to compare the mixing time with takeover time, then we have to use mixing time to climb one step of the mixing ladder (Equation 20) rather than the

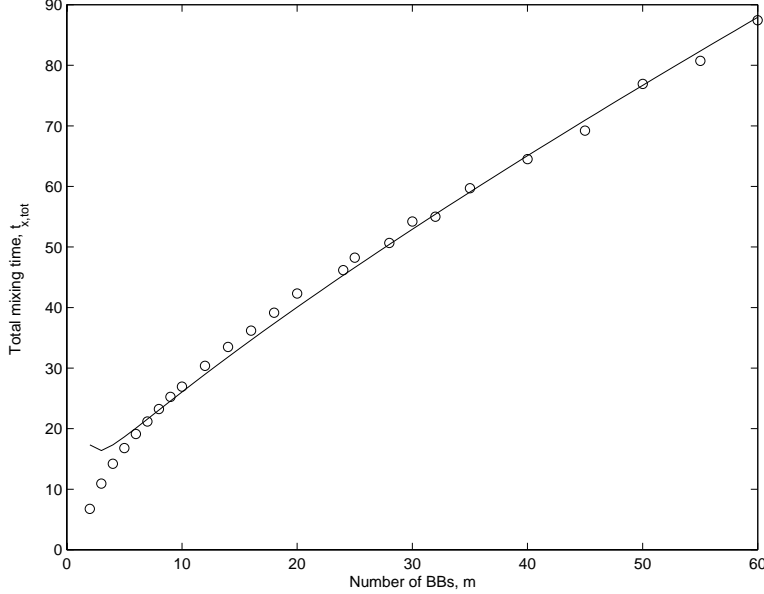


Figure 9: Comparison of total mixing time, $t_{x,tot}$ predicted by equations 21 with empirical results. The total mixing time is plotted as a function of number of BBs. The empirical results are for 4-bit deceptive trap function and are averaged over 500 independent runs.

total mixing time. This is because takeover time refers the time required for the best individual to takeover the whole population. Therefore, for complete innovative success occurs when the mixing time at every step in the ladder is less than the takeover time.

The mixing-time model (Equation 21) is compared to empirical results in figure 9, where the total mixing time is plotted as function of number of BBs, m . The results indicate that the agreement between theoretical and experimental results gets better with the number of BBs.

We are now ready to compare the mixing- and convergence-time models with one small caveat. In all the analysis presented for mixing in m -BB case, we have assumed that every individual in the population contains either a BB (all ones) or a deceptive attractor (all zeros). However, in the actual case, the probability of having a BB in a string is 2^{-k} . Therefore, the proportion of recombinations that can result in a mixing event is given by $2^{-mk} (2^{(m-1)k} - 1) \approx 2^k$. This multiplication factor is identical to the one used in section 5. Using this multiplication factor with the total mixing time and comparing it to the convergence-time model (equation 10) we get

$$c_{x,m} 2^k \frac{m^2}{np_c} < \frac{c_c}{I} \sqrt{mk} \quad (22)$$

Rearranging terms in the above equation, we obtain

$$n > c_x \frac{I}{p_c} 2^k m \sqrt{\frac{m}{k}} \quad (23)$$

where $c_x = c_{x,m}/c_c$ is a constant.

The above equation suggest that the population size grows as $O(2^k m \sqrt{m})$ as opposed to the population sizing predicted by the gambler's ruin model (Harik, Cantú-Paz, Goldberg, & Miller, 1997):

$$n = - \left(\frac{\sqrt{\pi \sigma_{bb}}}{2d} \right) \ln(\alpha_1) 2^k \sqrt{m} \quad (24)$$

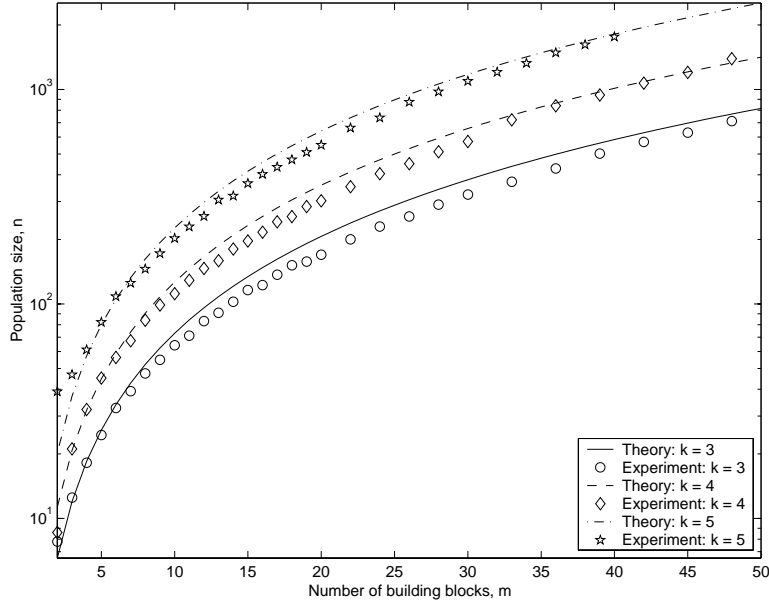


Figure 10: Empirical validation of the population-sizing model dictated by BB mixing (Equation 23). Minimum population size required for a GA success is plotted as a function of number of BBs m . The minimum population size is determined by a bisection method and the empirical results are averaged over 25 independent bisection runs.

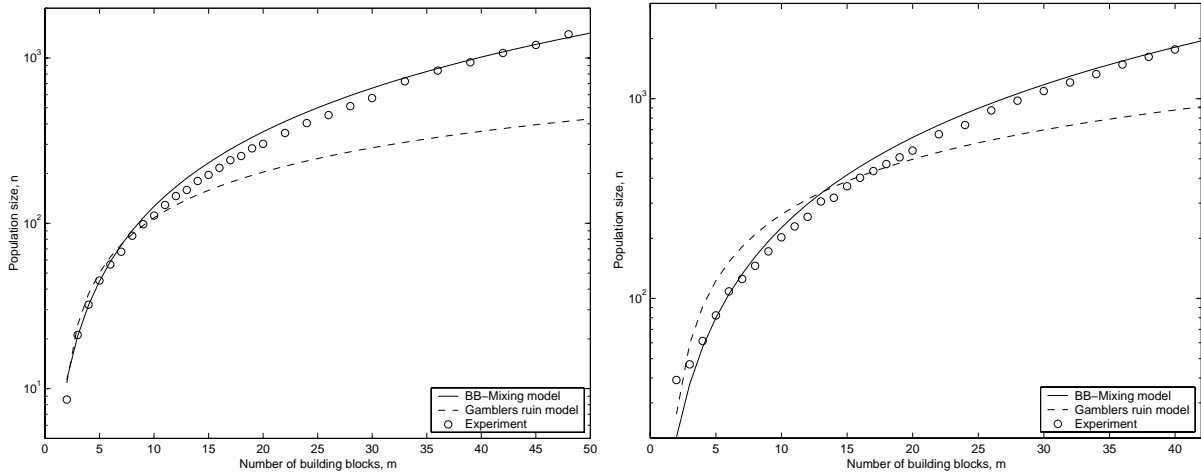
where α_1 (usually set to $1/m$) is the GA failure probability, d is the signal difference, quantified by the fitness difference between the best and the next best schemata, σ_{bb} is the fitness variance of schemata in a single partition.

The BB-mixing based population-sizing model (Equation 23) is verified with empirical results in figure 10. The figure plots the minimum population size required to obtain a solution of desired quality as a function of number of BBs. Population sizes for three different BB sizes, $k = 1, 2,$ and 3 are compared. The empirical results are obtained as follows: A binary tournament selection and single-point crossover with $p_c = 1$ is used. A GA run is terminated when all the individuals in the population converge to the same fitness value. The average number of BBs correctly converged are computed over 50 independent runs. The minimum population required such that the failure probability, α_1 , is at most $1/m$ is determined by a bisection method. The results shown in figure 3 are averaged over 25 such bisection runs.

The experimental results show good agreement with theoretical predictions, especially for moderate-to-large problems. It is interesting to compare the BB-mixing based and the gambler’s ruin based population-sizing models, which is shown in figure 11. Two key observations can be made from the figure:

1. Decision making bounds population sizing for small problems and the BB mixing bounds population sizing for moderate-to-large problems.
2. The range—in terms of number of BBs—in which decision-making bounds population sizing is dependent on the BB size. The range increases with the BB size.

Therefore, there is a boundary which segregates the problem space into two regions. One where BB decision-making governs the population sizing, and the other where BB-mixing governs the population sizing. Such a boundary, predicted by the respective population-sizing



(a) BB size, $k = 4$

(b) BB size, $k = 5$

Figure 11: Comparison of the mixing-based population-sizing model (Equation 23) and Gambler’s-ruin-based population-sizing model (Equation 24) with empirical results. Results indicate that BB supply and BB decision-making bound the population sizing for small problem sizes and BB mixing bounds the population sizing for moderate-to-large problem sizes.

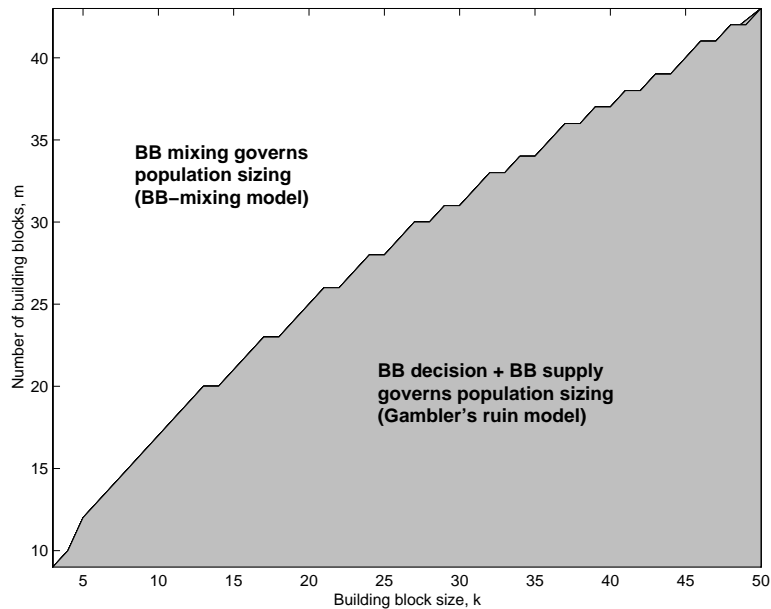


Figure 12: Areas in problem space where BB-mixing and BB decision-making models govern the population sizing required for a GA success. Equations 23 and 24 are used to compute the transition boundary. Given a problem with BB-size, k , and m , the gambler’s ruin model governs the population sizing if the point (k, m) lies below the boundary line. Otherwise—if the point (k, m) lies above the boundary line—BB-mixing governs population sizing.

models (Equations 24, and 23) is shown in figure 12.

7 Summary and Conclusions

This study analyzed the mixing behavior of single-point crossover on boundedly decomposable additive fitness functions with tight linkage. Models for predicting the time required for achieving innovative mixing—first for a two building block problem, and then for a general m building block case—are developed. Two key features of single-point crossover, namely the *ladder-climbing* phenomenon, and the *length-reduction* phenomenon are observed empirically and successfully used in developing the mixing time model for m -BB case.

The resulting mixing time models are compared with existing convergence-time model to investigate scenarios which result in *innovative success*. Using such a comparison, population-sizing model dictated by BB mixing is derived. All the models derived in this study are compared with empirical results. Empirical results show good agreement with theory, thus validating the facetwise models.

Results indicate that the mixing time grows exponentially with building block size, linearly with the number of building blocks. Furthermore, the mixing time is inversely proportional to the population size and crossover probability. The key result of this study is that it suggests that the population-sizing required for a GA success is dictated by building-block mixing rather than building-block-wise decision-making or building-block supply for moderate-to-large problems. Furthermore, the minimum population size scales as $O(2^k m \sqrt{m})$, as opposed to $O(2^k \sqrt{m})$ which is the case for advanced operators that adapt linkage.

Acknowledgments

This work was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant F49620-00-0163, and the National Science Foundation under grant DMI-9908252. 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

- Altenberg, L. (1995). The schema theorem and Price's theorem. *Foundations of Genetic Algorithms*, 3, 23–50.
- Bäck, T. (1994). Selective pressure in evolutionary algorithms: A characterization of selection mechanisms. *Proceedings of the First IEEE Conference on Evolutionary Computation*, 57–62.
- Bagley, J. D. (1967). *The behavior of adaptive systems which employ genetic and correlation algorithms*. Doctoral dissertation, University of Michigan, Ann Arbor, MI. (University Microfilms No. 68-7556).
- Bailey, N. T. J. (1961). *Introduction to mathematical theory of genetic linkage*. London, UK: Oxford University Press.

- Booker, L. B. (1993). Recombination distributions for genetic algorithms. *Foundations of Genetic Algorithms, 2*, 29–44.
- Bridges, C. L., & Goldberg, D. E. (1987). An analysis of reproduction and crossover in binary-coded genetic algorithm. *Proceedings of the Second International Conference on Genetic Algorithms*, 9–13.
- Bulmer, M. G. (1985). *The mathematical theory of quantitative genetics*. Oxford: Oxford University Press.
- Christiansen, F. B. (1989). The effect of population subdivision on multiple loci without selection. In Feldman, M. W. (Ed.), *Mathematical Evolutionary Theory* (pp. 71–85). Princeton, NJ: Princeton University Press.
- De Jong, K. A. (1975). *An analysis of the behavior of a class of genetic adaptive systems*. Doctoral dissertation, University of Michigan, Ann-Arbor, MI. (University Microfilms No. 76-9381).
- De Jong, K. A., & Spears, W. M. (1990). An analysis of the interacting roles of population size and crossover in genetic algorithms. *Parallel Problem Solving from Nature*, 38–47. (Also AIC Report No. AIC-90-003).
- De Jong, K. A., & Spears, W. M. (1992). A formal analysis of the role of multi-point crossover in genetic algorithms. *Annals of Mathematics and Artificial Intelligence*, 5(1), 1–26.
- Deb, K., & Goldberg, D. E. (1993). Analyzing deception in trap functions. *Foundations of Genetic Algorithms, 2*, 93–108. (Also IlliGAL Report No. 91009).
- Eshelman, L. J. (1991). The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. *Foundations of Genetic Algorithms, 1*, 265–283.
- Eshelman, L. J., Caruana, R. A., & Schaffer, J. D. (1989). Biases in the crossover landscape. *Proceedings of the Third International Conference on Genetic Algorithms*, 10–19.
- Eshelman, L. J., & Schaffer, J. D. (1995). Productive recombination and propagating and preserving schemata. *Foundations of Genetic Algorithms, 3*, 299–313.
- Falconer, D. S. (1989). *Introduction to quantitative genetics*. London: Longman Scientific & Technical.
- Frantz, D. R. (1972). *Non-linearities in genetic adaptive search*. Doctoral dissertation, University of Michigan, Ann Arbor, MI. (University Microfilms No. 73-11,116).
- Geiringer, H. (1944). On the probability theory of linkage in mendelian heredity. *Annals of Mathematical Statistics, 15*, 25–57.
- Goldberg, D. E. (1987). Simple genetic algorithms and the minimal, deceptive problem. In Davis, L. (Ed.), *Genetic algorithms and simulated annealing* (Chapter 6, pp. 74–88). Los Altos, CA: Morgan Kaufmann.
- Goldberg, D. E. (1989a). Genetic algorithms and Walsh functions: Part II, deception and its analysis. *Complex Systems, 3*, 153–171. (Also IlliGAL Report No. 89001).
- Goldberg, D. E. (1989b). *Genetic algorithms in search optimization and machine learning*. Reading, MA: Addison-Wesley.
- Goldberg, D. E. (1991). *Six steps to GA happiness*. Paper presented at Oregon Graduate Institute, Beaverton, OR.

- Goldberg, D. E. (1999). The race, the hurdle, and the sweet spot: Lessons from genetic algorithms for the automation of design innovation and creativity. In Bentley, P. (Ed.), *Evolutionary Design by Computers* (Chapter 4, pp. 105–118). San Mateo, CA: Morgan Kaufmann.
- Goldberg, D. E. (in press). *Design of innovation: Lessons from and for competent genetic algorithms*. Boston, MA: Kluwer Academic Publishers.
- Goldberg, D. E., & Bridges, C. L. (1990). An analysis of a reordering operator on a ga-hard problem. *Biological Cybernetics*, 62, 397–405. (Also TCGA Report No. 88005).
- Goldberg, D. E., & Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. *Foundations of Genetic Algorithms*, 1, 69–93.
- Goldberg, D. E., Deb, K., & Clark, J. H. (1992). Genetic algorithms, noise, and the sizing of populations. *Complex Systems*, 6, 333–362. (Also IlliGAL Report No. 91010).
- 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).
- Goldberg, D. E., & Liepens, G. (1991). Theory tutorial. (Tutorial presented at the 1991 International Conference on Genetic Algorithms, La Jolla, CA).
- Goldberg, D. E., & Sastry, K. (2001). A practical schema theorem for genetic algorithm design and tuning. *Proceedings of the Genetic and Evolutionary Computation Conference*, 328–335. (Also IlliGAL Report No. 2001017).
- Goldberg, D. E., & Segrest, P. (1987). Finite Markov chain analysis of genetic algorithms. *Proceedings of the Second International Conference on Genetic Algorithms*, 1–8.
- Goldberg, D. E., Thierens, D., & Deb, K. (1993). Toward a better understanding of mixing in genetic algorithms. *Journal of the Society of Instrument and Control Engineers*, 32(1), 10–16. (Also IlliGAL Report No. 92009).
- Harik, G., Cantú-Paz, E., Goldberg, D. E., & Miller, B. L. (1997). The gambler’s ruin problem, genetic algorithms, and the sizing of populations. *Proceedings of the IEEE International Conference on Evolutionary Computation*, 7–12. (Also IlliGAL Report No. 96004).
- Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor, MI: University of Michigan Press.
- McPeck, M. S. (1996). Introduction to recombination and linkage analysis. In Speed, T., & Waterman, M. S. (Eds.), *Genetic Mapping and DNA Sequencing*, Volume 81 of *IMA Volumes in Mathematics and its Applications* (pp. 1–15). Springer-Verlag.
- Mühlenbein, H., & Schlierkamp-Voosen, D. (1993). Predictive models for the breeder genetic algorithm: I. continuous parameter optimization. *Evolutionary Computation*, 1(1), 25–49.
- Nix, A. E., & Vose, M. D. (1992). Modeling genetic algorithms with Markov chains. *Annals of Mathematics and Artificial Intelligence*, 5, 79–88.
- Prügel-Bennett, A. (2001). The mixing rate of different crossover operators. *Foundations of Genetic Algorithms*, 6, 261–274.
- Prügel-Bennett, A., & Shapiro, J. L. (1994). An analysis of genetic algorithms using statistical mechanics. *Physical Review Letters*, 72(9), 1305–1309.
- Rabani, Y., Rabinovich, Y., & Sinclair, A. (1998). A computational view of population genetics. *Random Structures & Algorithms*, 12(4), 313–334.

- Rana, S. (1999). The distributional biases of crossover operators. *Proceedings of the Genetic and Evolutionary Computation Conference*, 549–556.
- Ratray, L. M. (1996). *Modelling the dynamics of genetic algorithms using statistical mechanics*. Doctoral dissertation, University of Manchester, Manchester, UK.
- Robbins, R. (1918). Some applications of mathematics to breeding problems, III. *Genetics*, 3, 375–389.
- Spears, W. M., & De Jong, K. A. (1991). An analysis of multi-point crossover. *Foundations of Genetic Algorithms*, 1, 301–315. (Also AIC Report No. AIC-90-014).
- Stephens, C., & Waelbroeck, H. (1999). Schemata evolution and building blocks. *Evolutionary computation*, 7(2), 109–124.
- Syswerda, G. (1989). Uniform crossover in genetic algorithms. *Proceedings of the Third International Conference on Genetic Algorithms*, 2–9.
- Thierens, D. (1995). *Analysis and design of genetic algorithms*. Doctoral dissertation, Katholieke Universiteit Leuven, Leuven, Belgium.
- Thierens, D. (1996). Dimensional analysis of allele-wise mixing revisited. *Parallel Problem Solving from Nature*, 255–265.
- Thierens, D. (1999). Scalability problems of simple genetic algorithms. *Evolutionary Computation*, 7(4), 331–352.
- Thierens, D., & Goldberg, D. E. (1993). Mixing in genetic algorithms. *Proceedings of the Fifth International Conference On Genetic Algorithms*, 38–45.
- Vose, M. D., & Liepens, G. E. (1991). Punctuated equilibria in genetic search. *Complex Systems*, 5(1), 31–44.