# The Gambler's Ruin Problem, Genetic Algorithms, and the Sizing of Populations

Georges Harik, Erick Cantú-Paz, David E. Goldberg, and Brad L. Miller

*Abstract*— **This paper presents a model for predicting the convergence quality of genetic algorithms. The model incorporates previous knowledge about decision making in genetic algorithms and the initial supply of building blocks in a novel way. The result is an equation that accurately predicts the quality of the solution found by a GA using a given population size. Adjustments for different selection intensities are considered and computational experiments demonstrate the effectiveness of the model.**

## I. Introduction

The size of the population in a genetic algorithm (GA) is a major factor in determining the quality of convergence. The question of how to choose an adequate population size for a particular domain is difficult and has puzzled GA practitioners for a long time. Hard questions are better approached using a divide-and-conquer strategy and the population sizing issue is no exception. In this case, we can identify two factors that influence convergence quality: the initial supply of building blocks (BBs), and the selection of the best BBs over their competitors.

In this paper, our goal is to provide an integrated answer to the population sizing question. We present a model that unifies previous knowledge on the initial supply of BBs and the decision process between competing BBs on convergence quality. The unified model is based on the solution to the classic gambler's ruin problem and is shown to predict the convergence quality of genetic algorithms accurately.

We begin by presenting the background to this work in the next section. There we review both previous studies on population sizing and the facetwise decomposition of GA design that guides this work. Section 3 explains the characteristics of decision making between two competing BBs, which we use in section 4 to derive the population sizing model. Section 5 presents computational experiments on several problems to verify that the model correctly predicts the quality of convergence of GAs. We recognize the role of different selection intensities in the convergence of a GA and adjust the model in section 6 to account for them. The paper concludes with a summary of results and numerous suggestions on how to extend this work.

G. Harik is with the Department of Electrical Engineering and Computer Science at the University of Michigan and with the Illinois Genetic Algorithms Laboratory. E-mail: gharik@illigal.ge.uiuc.edu.

E. Cantú-Paz is with the Department of Computer Science at the University of Illinois at Urbana-Champaign (UIUC) and with the Illinois Genetic Algorithms Laboratory. E-mail: cantupaz@illigal.ge.uiuc.edu.

D. Goldberg is a Professor in the Department of General Engineering at UIUC and is the Director of the Illinois Genetic Algorithms Laboratory. E-mail: deg@illigal.ge.uiuc.edu.

B. Miller is with the Department of Computer Science at UIUC and with the Illinois Genetic Algorithms Laboratory. E-mail: bmiller@illigal.ge.uiuc.edu.

## II. Background

Over the years researchers have noticed that the population size is a major factor in determining the convergence quality of GAs. Large populations result in better solutions, but the question of exactly how big the population has to be to insure a certain quality has been largely ignored. However, there are a handful of studies that guide the user to choose adequate population sizes [1], [2].

Previous estimates of adequate population sizes fall into two categories: models concerned with the initial supply of BBs, and models that involve decision-making between competing BBs. In this paper we present a model that integrates these two facets to obtain a better estimate for the quality of convergence.

Following Goldberg, Deb, and Thierens [3], we restrict the notion of BB to the minimal-order schemata that contribute to the global optimum. For fully deceptive functions with deception length $k$, the BBs are the optimal schemata of length $k$. In this view the juxtaposition of two BBs at a particular string does not lead to a BB of length $2k$, but instead we regard them as two separate BBs of length $k$.

### A. Supply models

Ignoring mutation the only source for diversity is the random initialization of the population. If the BBs are not present in the initial population, it is unlikely that mutation will create them. For this reason, we ignore mutation in this study and consider only the initial availability of correct BBs.

The first attempt to develop a model for the population size required to insure an adequate supply of BBs in the initial population is due to Holland [4]. This area remained untouched for many years until Goldberg computed bounding cases for serial and parallel GAs [1].

### B. Decision models

The second aspect of population sizing involves selecting better partial solutions. Holland [4], [5] recognized that the issue of choosing between BBs (and not between complete strings) can be recast as the two-armed bandit problem, a well-known problem in statistical decision theory. This classic problem is a concrete example of the tradeoff between exploration of a sample space and exploitation of the information already gathered. Holland's work assumes an idealization of the GA as a cluster of interconnected 2-armed bandits, so that his result relating the expected loss and the number of trials can be directly applied to schema processing. Even though Holland's calculations are based on an idealization, his results give an optimistic bound on the allocation of trials on a real GA.

Around the same time, De Jong recognized the importance of noise in the decision process and proposed an estimate of the population size based on the signal and noise characteristics of the problem [6]. Unfortunately, he did not use his estimate in the rest of his study and the result remained unverified and ignored by many.

Goldberg and Rudnick gave the first population sizing estimate based on the variance of fitness [7]. And more recently, Goldberg, Deb, and Clark developed a decision-based model that gives a conservative bound on the convergence quality of GAs [2]. Their model is based on deciding between the best BB in a partition and its closest competitor. The decision process is affected by collateral noise from other BBs present in the string.

The result of that investigation is a population sizing equation that for binary alphabets and ignoring external sources of noise is:

$$n = 2c(\alpha)2^k m' \frac{\sigma_M^2}{d^2} \qquad (1)$$

where $c(\alpha)$ is the square of the ordinate of a unit normal distribution where the probability equals $\alpha$; $\alpha$ is the probability of failure; $k$ is the order of the BB; $m'$ is one less than the number of BBs in a string ($m$); $\sigma_M^2$ is the average fitness variance of the partition; and $d$ is the signal difference between the best and second best BBs.

The major limitation of this model is that it only considers the first generation of the run. The model assumes that if the decision-making process favors the wrong BBs in the first generation the GA is unable to recover from the error. Also, if the decisions are correct in the first generation the model assumes that the GA converges to the right BB. Considering only the result of the first generation results in a conservative estimate of the convergence quality. We compare this estimate with the prediction from the gambler's ruin model in the experimental section of this paper.

Mühlenbein derived an expression for the expected quality of convergence for the counting-bits (or one-max) problem [8]. Calibrating their theoretical results with empirical experimentation resulted in a very accurate model for the particular problem they studied.

### C. Decomposing the problem

Despite their operational simplicity, GAs are complex algorithms. To have any hope of understanding and designing GAs we need to approach them as we do with other difficult engineering tasks: decompose the problem into tractable sub-problems, solve the sub-problems, and integrate the partial solutions into an integrated whole.

For some time now, we have used a rational, facetwise decomposition as a guide in our study of GAs [7], [2]. The decomposition consists of six points:

1. Know what the GA is processing: building blocks (BBs)
2. Solve problems tractable by BBs
3. Supply enough BBs in the initial population
4. Ensure the growth of necessary BBs
5. Mix the BBs properly
6. Decide well among competing BBs

This study involves two of these issues: the initial supply of BBs and the decision process between competing BBs. The population-sizing model is based mainly on the cumulative effects of good decision making over time, but it also depends on the number of correct BBs present in the initial population. This model is what has elsewhere [9] been called a little model, but to understand it we first need to review the decision process between two BBs.

## III. Deciding between two BBs

The procedure to decide between the best BB and the second best BB in a partition was discussed by Goldberg, Deb, and Clark (1992). Here we review that discussion to obtain a domain-dependent decision probability. We use the result of this section later as a piece of the gambler's ruin model.

Consider two BBs $H_1$ (with mean static fitness $f_{H_1}$ and fitness variance $\sigma_{H_1}^2$) and $H_2$ (with mean static fitness $f_{H_2}$ and fitness variance $\sigma_{H_2}^2$). We can assume that the fitness distributions are normal by the central limit theorem. Assuming that $H_1$ has a higher fitness, we hope to select more representatives of $H_1$ than of $H_2$. However, in a one-on-one competition there is a chance of erroneously choosing a representative of $H_2$. To compute this probability we have to accumulate all possible values of the area of the region where the fitness distributions of the two schemas overlap. This computation is called a *convolution* and, in the case of normal variables, the convolution itself is normally distributed and has known properties. The mean of the convolution is the difference of the means of the two individual distributions and the variance of the convolution is the sum of the individual variances.

Define the signal difference $d = f_{H_1} - f_{H_2}$. The probability of making a error on a single trial may be calculated as:

$$p = N\left(\frac{d}{\sqrt{\sigma_{H_1}^2 + \sigma_{H_2}^2}}\right)$$

where $N$ is the cumulative distribution function (CDF) for a normal distribution with zero mean and unit standard distribution.

As the problem consists on many BB competitions being held in parallel, we need to specialize this result to account for added noise introduced by other BBs. Following Goldberg, Deb, and Clark [2], we assume that the fitness function is the sum of $m$ independent and equally-scaled subfunctions $f_i$, each of the same size $k$ of the most deceptive partition. The overall variance of the function may be calculated as:

$$\sigma_f^2 = \sum_{i=1}^{m} \sigma_{f_i}^2$$

The root-mean-squared subfunction variance (average BB variance) is computed as: $\sigma_{BB}^2 = \sigma_f^2/m$. For each BB competition, there is noise resulting from the signal of the other $m' = m - 1$ BBs: $\sigma^2 = m'\sigma_{BB}^2$. Therefore the probability of making the right choice between a single sample of each BB becomes:

$$p = N\left(\frac{d}{\sigma_{BB}\sqrt{2m'}}\right) \quad (2)$$

In the next section we use this value to find the probability that the GA ultimately converges to the correct solution.

## IV. Gambler's ruin model

Random walks are mathematical tools that can be used to predict the outcome of certain stochastic processes. The most basic random walk deals with a particle that moves left and right in a one-dimensional space with certain probabilities. The size of the step is fixed and sometimes the movement of the particle is restricted by placing barriers at some points in the space. For our purposes, we consider a one-dimensional walk with two absorbing barriers that capture the particle once it reaches them.

The gambler's ruin problem is a classic example of a random walk with absorbing barriers and it has been studied extensively. In this problem the capital of a gambler is represented by the position, $x$, of the particle on one-dimensional space. Initially, the particle is positioned at $x = a$ where $a$ represents the gambler's starting capital. The gambler plays against an opponent that has an initial capital of $n - a$, and there are absorbing boundaries at $x = 0$ (representing bankruptcy) and at $x = n$ (representing winning all the opponent's money). At each step in the game, the gambler has a probability $p$ of increasing his capital by one unit and a probability $q = 1 - p$ of loosing one unit to his opponent.

To model the decision process in GAs as a random walk we concentrate on only one building block. The position of the particle on the one-dimensional space, $x$, represents the number of correct BBs. The space is bounded by absorbing barriers at $x = 0$ and $x = n$ that represent convergence to the wrong and right solutions, respectively. The initial position of the particle, $x_0$, is the expected number of copies of the best BB in a randomly initialized population, that is equal to $x_0 = n/2^k$, where $k$ is the order of the BB.

To apply the random walk model we have to abandon the notion of generations in a GA and we consider that decisions occur one at a time until all the $n$ BBs have converged to the same value. Also, we are implicitly assuming that the GA uses tournament selection, but the model can be modified for other selection schemes as we show later.

To compute the probability that the particle is captured at $x = n$, we use a well-known result from the random walk literature [10]:

$$P_n = \frac{1 - (q/p)^{x_0}}{1 - (q/p)^n} \quad (3)$$

where $q = 1 - p$ is the probability of making the wrong decision between two BBs.

It is straightforward to find an expression for the population size from this equation. Note that in our case $p > 1-p$ (because $f_{H_1} > f_{H_2}$ the convolution results in a $p > 0.5$) and that $x_0$ is usually small compared to the population size. For increasing values of $n$ the denominator in equation 3 approaches 1 faster than the numerator. Under these conditions and substituting $x_0 = n/2^k$ and $q = 1 - p$ we can simplify $P_n$ to:

$$P_n \approx 1 - \left(\frac{1-p}{p}\right)^{n/2^k} \quad (4)$$

Let $\alpha = 1 - P_n$ be the probability of failure. Solving the equation above for $n$ we get:

$$n = 2^k \ln(\alpha) / \ln\left(\frac{1-p}{p}\right) \quad (5)$$

where $p$, the probability of making the right decision between two BBs is given by equation 2.

To better appreciate the relationships between all the variables involved, we can approximate the equation above. First, $p$ can be approximated using the first two terms of the power series expansion for the normal distribution as [11]:

$$p = \frac{1}{2} + \frac{1}{2}x$$

where $x = d/(\sigma_{BB}\sqrt{2m'})$. Substituting the approximation for $p$ into equation 5 we get:

$$n = 2^k \ln(\alpha) / \ln\left(\frac{1-x}{1+x}\right)$$

Since $x$ is a small number, we can approximate $\ln(1-x)$ with $-x$ and $\ln(1+x)$ with $x$. Using these approximations and substituting the value of $x$ we get:

$$n = -2^{k-1}\ln(\alpha)\frac{\sigma_{BB}\sqrt{m'}}{d} \quad (6)$$

Using this rough approximation we can appreciate that the population size gets larger as the average variance of the BBs increases and also as the problem size grows. Also, a larger signal difference $d$ makes the problem easier and consequently the required population size gets smaller.

## V. Experiments

This section presents experimental evidence that validates our model. In all the experiments of this section we use a simple GA with binary tournament selection and a crossover operator that provides good mixing of BBs without excessive disruption. As such, the choice of crossover operator is problem-dependent and we specify it in each subsection. The probability of crossover is 1.0 and the mutation rate is set to zero, because the only source of diversity considered by the model comes from the initial population. The termination criteria was to reach a uniform population, and in all cases the results presented are the average of 50 runs.

We use equation 3 to plot the theoretical predictions along with the experimental results. The results from equation 4 are indistinguishable except where the population size is very small.

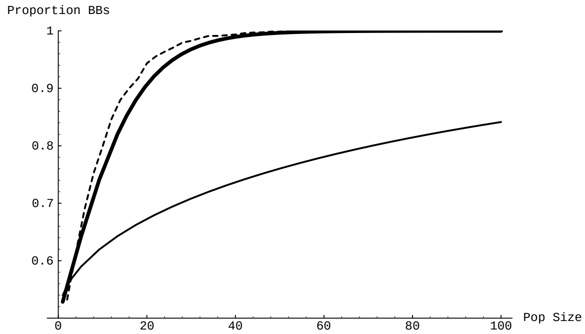


Fig. 1. Experimental and theoretical results of the proportion of correct BBs on a 100-bit one-max function. The prediction of the random walk model is in bold, the experimental results are the dotted line, and the previous decision-based model is the thin line.

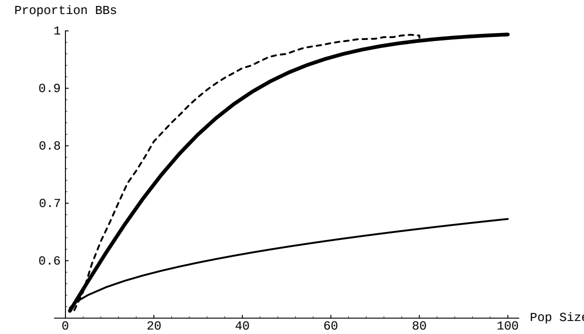


Fig. 2. Experimental and theoretical results of the proportion of correct BBs on a 500-bit one-max function.

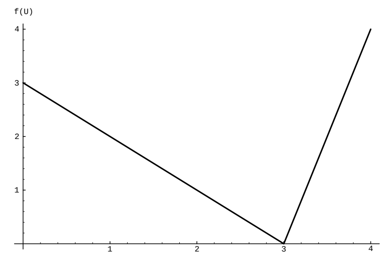


Fig. 3. A 4-bit deceptive function.

### A. *One-max*

The one-max function counts the number of bits set to 1 in the string and uses that value as the fitness of the individual. For the one-max experiments we use uniform crossover as it results in a good exchange of BBs and we are unconcerned with disruption, since the BBs have length equal to one.

For the one-max function, the signal difference $d$ equals 1 and the variance $\sigma_{BB}^2 = 0.25$. This is a very easy problem for GAs since there is no isolation or deception and the BB are short. The supply of BBs is no problem either since in a randomly initialized population we expect to find 50% of correct BBs.

In figure 1 the bold line is the theoretical prediction of the gambler's ruin model and the dotted line is the experimental results for the proportion of BBs correct at the end of a run for a 100-bit function. The thin line is the theoretical prediction of the population sizing model of Goldberg, Deb, and Clark [2]. Figure 2 presents the same results for a 500-bit problem. From the figures it is evident that the previous decision making model gives a very conservative prediction of quality. As we discussed earlier, the main reason for this is that the model only considers decisions in the first generation of the run.

The gambler's ruin model predicts the outcome of the experiments for the 100 and 500-bit functions quite accurately. However, in the 500-bit function the match is not as close as in the 100-bit case. We believe that the reason for this small discrepancy is that the theory only considers one BB at a time and that decisions for one BB are independent of all the others. In order to achieve this independence, the crossover operation must distribute BBs completely at random across all the individuals in the population. The goal of crossover in this case is to smooth the distribution of BB in the different alleles to avoid hitchhiking. To achieve this perfect distribution is not practical because it would involve many rounds of crossover each generation.

### B. *Trap functions*

The next set of experiments uses deceptive trap functions. The first example is the 4-bit trap function depicted in figure 3. This function was also used by Goldberg, Deb, and Clark [2]. The horizontal axis is the number of bits equal to 1 and the vertical axis is the fitness value. The signal difference $d$ (the distance between the two peaks) is 1 and the fitness variance ($\sigma_{BB}^2$) equals 1.215. Our test function is formed by concatenating 20 copies of the trap function for a total string length of 80 bits. To solve the trap functions, we used tight linkage, although there are algorithms like the messy GA [12] that are able to find tight linkages autonomously. This function is hard for traditional optimizers, but it has been shown that GAs can solve it satisfactorily when using properly sized populations [2].

Since the BBs in this function are longer than in the one-max problem, we chose a two-point crossover operator to avoid the excessive BB disruption of uniform crossover. The crossover probability was set to 1.0.

Figure 4 presents the prediction of the number of BBs correct at the end of the run along with the results from the experiments. Again, the bold line is the prediction of the random walk model and the thin line is the prediction of the previous decision-making model. Note that the convergence quality for small population sizes is dominated by the initial supply of BBs, and therefore, the previous decision-making model is not accurate at all in that region.

The second example of a trap function is a concatenation of 8 copies of an 8-bit trap function. The 8-bit trap is similar to the 4-bit above with a signal difference $d = 1$ but the fitness variance is $\sigma_{BB}^2 = 2.1804$. The higher variance

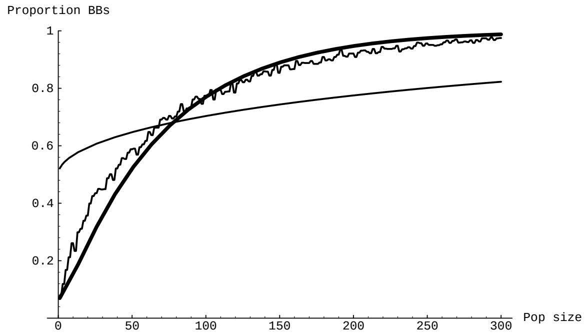


Fig. 4. Results comparing theory and experimental results for a 4-bit deceptive function with 20 BBs.

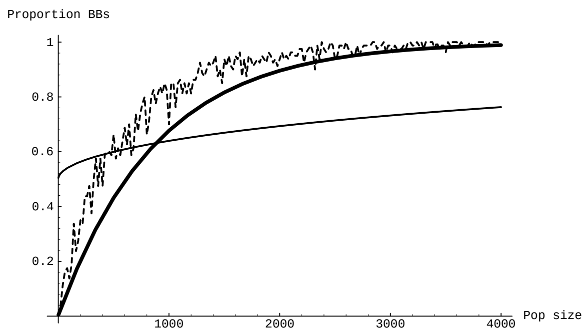


Fig. 5. Results comparing theory and experimental results for an 8-bit deceptive function with 8 BBs.

makes this function more difficult than the 4-bit example and we expect to use much larger populations to reach the desired quality. For this problem we used 1-point crossover since the BBs are longer and more likely to be disrupted by crossover. The crossover probability was again set to 1.0.

Figure 5 shows the results for this problem. As in previous cases, the random walk model (in bold) approximates the experimental results well. And, as expected, the previous population sizing model (thin line) gives a very conservative estimate of convergence quality.

The gambler's ruin model approximates very accurately the experimental results for the functions considered here. In the next section, we will extend the model to account for different selection pressures.

## VI. Effect of selection type

Besides the population size, an important factor in the convergence quality of GAs is the selection scheme. After all, the selection operator is the part of the GA making the decisions we have discussed in the previous sections. The goal of any selection method is to favor the proliferation of good individuals in the population. However, selection methods differ in how they allocate copies of good individuals. More formally, the *selection intensity* can be viewed as the expected average fitness after selection of a population whose fitness distribution is normally distributed [8].

The selection pressure of tournament selection varies with the size of the tournament size $s$. As the tournament

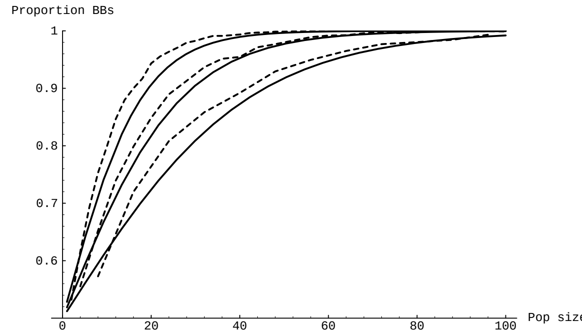


Fig. 6. Results comparing theory and experimental results for a 100-bit one-max function varying the selection intensity. From left to right: $s = 2, 4, 8$

size gets larger, the selection intensity increases. Here we only show results for tournament selection, but the results can be extrapolated to other selection methods with known selection intensities [13].

We conservatively assume that the correct BB is competing against $s-1$ copies of the second best BB. As the tournament size gets larger, the probability of making the wrong decision increases proportionately to $s$; thus, we approximate the probability of making the right decision as $1/s$. In reality this probability is higher than $1/s$ since a tournament might involve more than one copy of the best BB, especially as the run progresses and the proportion of correct BBs increases, but $1/s$ is a good initial approximation as we demonstrate.

The increasing difficulty of decision-making as the tournament size increases can be accounted for as a contraction in the signal that the GA is trying to detect. Setting the new signal to:

$$d' = d + z(1/s)\sigma_{BB}$$

where $z(1/s)$ is the ordinate of a unit normal distribution where the CDF equals $1/s$, we can compute a new probability of deciding well using $d'$ instead of $d$ in equation 2.

We experimented using a 100-bit one-max function and tournament sizes of 2, 4, and 8. The experimental results are plotted in figure 6 with dotted lines along with the theoretical predictions. The leftmost plot corresponds to a tournament size $s = 2$, the next is $s = 4$ and the last is $s = 8$. Once again, the model is a good predictor for the proportion of BBs correct at the end of the run, even as we consider a wide range of tournament sizes.

## VII. Extensions

The gambler's ruin model integrates two facets that influence the convergence of GAs: the initial supply of BBs ($x_0$) and decision-making between competing BBs ($p$). The fact that the integration of two facetwise models works so well is a proof that our principle of divide-and-conquer can have good results. Following this notion of integrating small models, the result of this paper can be extended to include other facets of GA efficiency. In previous sections we hinted at some possible extensions and here we elaborate them and present other possibilities.

Probably the most important extension to the model is to consider the effect of crossover. In its current state, the model assumes that all the BBs in a string are independent of each other and that the correct BBs are distributed evenly along the whole population. Implicit in this is the assertion that BBs are mixed properly and that strings have on average the same number of correct BBs. In reality, some strings may contain more copies of the correct BB and reproduce faster making the crossover operator unable to mix the BBs evenly.

Furthermore, crossover presents a tradeoff: we want to distribute BBs at random throughout the population, but we do not want to disrupt those BBs that were found already. The use of more aggressive crossover operators (e.g., 5-point or uniform) increases the chances that BBs mix correctly, but also more BBs are disrupted. We need to model both of these effects to find a balance between mixing and disruption.

In this study we considered only tournament selection and we showed a way to adjust for different tournament sizes. Changing the tournament size affects the selection pressure and the computations can be extrapolated to other schemes with known intensities. However, proportional selection does not have a constant selection pressure and a separate analysis is necessary for this case.

Another extension of the model deals with the sizing of populations for parallel GAs. Parallel implementations are important because they open many practical opportunities. Work is already underway to model and understand some important aspects of these algorithms. The gambler's ruin model can be easily extended to account for multiple isolated populations and it can also be modified to consider migration.

## VIII. Conclusions

In this paper we presented a model that predicts the probability that a GA converges to the correct BB. The model is based on a simple random walk where the position of a particle on a one-dimensional space represents the number of correct BBs in the population. Placing absorbing barriers in the one dimensional space results in an instance of the gambler's ruin problem. This problem has been studied extensively and an exact solution to the probability of absorption is well known. We used this solution to derive an expression for the population size required to reach a solution of any quality.

The gambler's ruin model was tested performing computational experiments with several functions. The test problems ranged from the very simple to the moderately hard and the results confirm that the model is a good predictor of performance.

In the gambler's ruin model, two key pieces of GA performance came together into a very accurate model for the quality of convergence. The first piece is the initial supply of building blocks that was used as the starting point for the random walk and the second piece models the competition between BBs.

This kind of integration of small models opens opportunities to develop more complete models of GA performance. As a first example, we extended the model to account for the effect of different selection types on the quality of convergence. The extension was tested experimentally and again the results matched the predictions very closely. More extensions are possible and work is already underway on some of them.


## Acknowledgments

This study was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grants number F49620-94-1-0103 and F49620-95-1-0338. The U.S. Government is authorized to reproduce and distribute reprints for Governmental 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 or the U.S. Government.

Erick Cantú-Paz was partially supported by a Fulbright-García Robles-CONACYT Fellowship.

Brad Miller was supported by NASA under grant number 9-4.

We also want to thank the anonymous reviewers for their useful comments.


## References


[1] D. E. Goldberg, "Sizing populations for serial and parallel genetic algorithms," in *Proceedings of the Third International Conference on Genetic Algorithms*, J. D. Schaffer, Ed., San Mateo, CA, 1989, pp. 70–79, Morgan Kaufmann, (Also TCGA Report 88004).

[2] D. E. Goldberg, K. Deb, and J. H. Clark, "Genetic algorithms, noise, and the sizing of populations," *Complex Systems*, vol. 6, pp. 333–362, 1992.

[3] D. E. Goldberg, K. Deb, and D. Thierens, "Toward a better understanding of mixing in genetic algorithms," *Journal of the Society of Instrument and Control Engineers*, vol. 32, no. 1, pp. 10–16, 1993.

[4] J. H. Holland, *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor, MI, 1975.

[5] J. H. Holland, "Genetic algorithms and the optimal allocations of trials," *SIAM Journal of Computing*, vol. 2, no. 2, pp. 88–105, 1973.

[6] K. A. De Jong, "An analysis of the behavior of a class of genetic adaptive systems," *Dissertation Abstracts International*, vol. 36, no. 10, pp. 5140B, 1975, (University Microfilms No. 76-9381).

[7] D. E. Goldberg and M. Rudnick, "Genetic algorithms and the variance of fitness," *Complex Systems*, vol. 5, no. 3, pp. 265–278, 1991, (Also IlliGAL Report No. 91001).

[8] H. Mühlenbein and D. Schlierkamp-Voosen, "Predictive models for the breeder genetic algorithm: I. Continuous parameter optimization," *Evolutionary Computation*, vol. 1, no. 1, pp. 25–49, 1993.

[9] D.E. Goldberg, "Toward a mechanic of conceptual machines," in *Proceedings of the Eighteenth Southeastern Conference on Theoretical and Applied Mechanics*, H.B. Wilson, R.C. Batra, and C.W. Bert, Eds. 1996, pp. 1–9, The University of Alabama.

[10] W. Feller, *An Introduction to probability theory and its applications*, vol. 1, Wiley, 2nd edition, 1966.

[11] M. Abramovitz and I. (eds) Stegun, *Handbook of mathematic functions with formulas, graphs, and mathematical tables*, Dover Publications, 1972.

[12] D. E. Goldberg, B. Korb, and K. Deb, "Messy genetic algorithms: Motivation, analysis, and first results," *Complex Systems*, vol. 3, no. 5, pp. 493–530, 1989, (Also TCGA Report 89003).

[13] B.L. Miller and D.E. Goldberg, "Genetic algorithms, selection schemes, and the varying effects of noise," *Evolutionary Computation*, vol. 4, no. 2, 1996.