Complex Systems **7** (1993) 155–170

Finite Markov Chain Models of an Alternative Selection Strategy for the Genetic Algorithm

Samir W. Mahfoud*
Department of Computer Science,
University of Illinois at Urbana-Champaign,
1304 West Springfield Avenue
Urbana, IL 61801, USA

Abstract. This paper presents finite Markov chain models of the selection strategy known as Boltzmann tournament selection. Unlike previous research at the string level, this study represents populations at the more general, equivalence-class level. The changing distribution of classes is analyzed using Markov chains, and a Markov chain model is used to predict expected drift time for the selection procedure. The accuracy of the final model is verified through direct simulation.

1. Introduction

In a genetic algorithm (GA), the composition of the population during a particular generation depends, probabilistically, only upon the population of the previous generation. Therefore, a GA can be effectively modeled by a Markov chain. Properties of finite populations (such as genetic drift) can be investigated using finite Markov chain models.

Markov chains are capable of representing a GA to any level of detail and, unlike many other methods of simulation, are exact for a particular level. The subjects of previous studies have ranged from fitness-proportionate selection acting alone on the one-bit strings of a finite population [5], to the fixed convergence points of the genotypes of an infinite-population-size GA with fitness-proportionate selection, single-point crossover, and mutation [10].

In order to set up a Markov chain for a particular GA, one must first partition the search space into *equivalence classes*, such that each point in the search space is a member of exactly one class. A class thus represents a set of potential solutions which are similar in some meaningful way. Assuming that there are c classes, numbered from 0 to $c-1$, if I_i represents the number of elements of the GA population which are members of $class_i$, $\sum_{i=0}^{c-1} I_i = n$,

*Electronic mail address: mahfoud@gal4.ge.uiuc.edu

where n is the population size. One partitioning method is to let each unique genotype represent a class, yielding 2^l classes, where l is the binary string length. Another method, appropriate for multimodal function optimization, is to define classes based upon peak membership—that is, each population member is an element of the class defined by the local optimum to which it is attracted. In that case, there are as many classes as there are peaks in the fitness landscape. Other class partitions (such as bit-based, objective-function-based, phenotypic, and schema partitions) are also possible.

This paper is a case study of Markov chain modeling of alternative genetic selection procedures, using Boltzmann tournament selection (BTS). The property of the selection procedure that we emphasize is its drift time, or expected time to absorption. This quantity, which is defined later in more detail, can serve as a measure of stability for a particular selection scheme. Because one purpose of BTS is to maintain stable subpopulations (like the method of sharing [4] does), it is essential to indicate the points at which entire classes can be expected to disappear from the population. From such a drift analysis, one can then hope to determine the degree to which the algorithm is capable of maintaining a stable distribution of solutions.

2. Boltzmann tournament selection

Boltzmann tournament selection [2] is a selection procedure for genetic algorithms that is motivated by simulated annealing. BTS maintains diversity in a population by sampling in such a way that the samples, over time, become Boltzmann distributed. However, the factor of genetic drift [5] works against BTS by limiting the amount of diversity it can maintain. A sketch of the BTS procedure follows (for full details, see [2]).

n individuals make up a population. In each generation, a three-way competition or "tournament" is held for each population slot. The first individual—call it ind_1—is chosen at random. ind_2 is also chosen at random, but must differ in fitness from ind_1 by an amount Θ. ind_3 must also differ from ind_1 by Θ; in half of the trials ind_3 must also differ from ind_2 by Θ (this is called *strict choice*, as opposed to *relaxed choice*, which occurs in the other half of the trials). ind_2 and ind_3 compete in a preliminary competition, with ind_2 probabilistically winning according to the logistic function

$$\frac{1}{1 + e^{[E(ind_3) - E(ind_2)]/T}}$$

The winner then competes against ind_1, with ind_1 advancing to the next generation with probability

$$\frac{1}{1 + e^{[E(ind_1) - E(winner)]/T}}$$

where $E(winner)$ is the energy or negated fitness of the winner of the previous competition, and T is the current temperature of BTS.

BTS forms classes that are nearly Boltzmann distributed, based on objective function values (energies). Goldberg's implementation, in searching

for an individual whose objective function value differs by Θ, cycles through a fixed percentage of the population, before giving up and arbitrarily choosing an individual. In this study, the entire population is checked before the algorithm is allowed to pick an identical or nearly-identical individual. Θ is consistently set to a value small enough to allow all classes to be distinguished. At high temperatures, the model is independent of any particular objective function. At lower temperatures, representative objective function values must be assigned to each class.

Goldberg [2] shows that the expected allocation of population elements to classes in BTS approaches the Boltzmann distribution at high and low temperatures. He gives results of a simulation, using five classes, that approaches the Boltzmann distribution at all temperatures, though with a good bit of variance. It is precisely this variance with which we are concerned.

3. Genetic drift

GAs that use selection alone are incapable of generating solutions that are not currently in the population. Aggravating this situation is the fact that some good solutions in the population eventually disappear due to the variance of the selection process. In fact, if the typical selection scheme is run long enough, all but one solution will disappear, even without selection pressure (high temperatures in BTS result in little to no selection pressure). This phenomenon is known as *genetic drift*. Although we may expect a quasi-stable Boltzmann distribution at a particular temperature in BTS, in actuality, genetic drift can lead to a rapid loss of diversity across classes.

As a selection scheme runs, the number of alternative solutions declines. Of course, one does not run a GA using selection alone. Crossover and mutation (along with many other operators) are used to explore solutions not currently present. However, these operators are limited in their abilities to explore the solution space. When operating on identical or nearly identical strings, crossover will yield results similar or identical to the parent strings; thus, it cannot be relied upon to reintroduce diversity. Mutation, though it can potentially explore the full solution space, is also of dubious value when it comes to reintroducing diversity: too high a mutation rate will lead to a more random search procedure, and random search is of very little use on sufficiently hard problems.

4. Markov chain analysis of genetic algorithms

Because Markov chains are concerned with the transition of a system from state to state, they can be used in the analysis of genetic algorithms; as a GA proceeds from generation to generation, each current or potential population can be considered a state.

A *finite Markov chain* [6, 7] is a sequence of trials in which there are a finite number of possible outcomes (states) for each trial, and each state depends on the previous state only. A Markov chain is specified by a matrix

of probabilities of moving from state to state. This *transition matrix* stays constant from trial to trial. Raising the transition matrix to the power k will yield a matrix containing probabilities of each outcome after k trials, where the row headings of the resulting matrix specify starting states, and the column headings specify ending states. Each entry in this matrix will thus contain the probability of going in k iterations from a particular starting state to a particular final state.

A Markov chain is *regular* if some power of its transition matrix contains all positive entries. A Markov chain is *absorbing* if it has at least one absorbing state, and if, for each nonabsorbing state, there exists a possibility of transition (perhaps using many steps) to an absorbing state. (An absorbing state is one in which transition to any other state is impossible, but probability of transition to itself is 1.) Long-term transition matrices can be calculated for both regular and absorbing Markov chains; they predict where a system will end up in the long run. Absorbing Markov chains, regardless of initial state, will eventually end up in an absorbing state. For absorbing chains, we can calculate the expected number of iterations to absorption, and the expected time spent in each transient (nonabsorbing) state. This study utilizes only absorbing Markov chains.

Traditional selection algorithms, when run long enough, will eventually lose every representative of every class but one. This is proved by noting that for any population, there exists some positive probability of proceeding (perhaps over many generations) to a population in which only one class remains. Once a class is lost, it can not be regained through selection alone. If we consider all possible class distributions as states of an absorbing Markov chain, it is possible to eventually move from any state to one in which only one class remains (an absorbing state).

Boltzmann tournament selection is no exception to this rule. Since BTS employs a three-way tournament, we consider an absorbing state to be one in which two or fewer classes remain. (A degenerate BTS will still work with just two classes; it holds tournaments using identical individuals and eventually gets rid of all but one class.) We are interested in calculating the expected number of generations to absorption for different population sizes, number and distribution of initial classes, objective functions, and temperatures. We are also interested in noting the points at which classes are lost. This is accomplished by redefining an absorbing state using $ABS > 2$, where ABS is a lower threshold on the number of distinct classes represented (remaining) in a population, at or below which we consider the population to correspond to an absorbing state.

5. Previous research

Goldberg and Segrest [5] use absorbing finite Markov chains to simulate a simple GA, which runs selection alone, on a population containing two equally fit classes (represented by the two one-bit strings, '0' and '1'). They also extend the model to include mutation. They calculate the expected

number of generations to "first passage" (initial arrival to a state in which one class has taken over a specified percentage of the population), for different population sizes and percent-convergence levels. As anticipated, the GA without mutation is expected to converge fairly rapidly (as a linear function of the population size) to a population containing the specified percentage of one or the other solution. The GA with mutation is more successful at delaying convergence, depending on how high the mutation rate is. Nevertheless, it is still expected to converge; in this case, convergence time is an exponential function of the population size. A second result deals with two unequally fit classes: the higher the fitness differential, the faster the expected convergence, and the less likely that the weaker class wins in the long run.

Two more recent papers present more exhaustive Markov chain models of the simple GA [1, 8], incorporating fitness-proportionate selection, crossover, and mutation into the calculation of transition matrix probabilities. Classes are defined at the genotype level, with each unique string representing a class. The analysis is complete, in that the number of classes (2^l, where l is the length of the binary string) is not restricted. For a population of n strings of length l, there are $(n+x)!/(n!\,x!)$ possible populations, where $x = 2^l - 1$. For example, given ten strings of length ten, we need approximately 3.65×10^{23} states in our transition matrix. Clearly, for any nontrivial problem, one could not reasonably expect to calculate all transition probabilities, nor to process the resulting matrix.

Rather than use the transition matrix for calculations, Nix and Vose employ the general formula for a transition probability in describing the asymptotic behavior of the GA as population size increases. Davis and Principe attempt to use the Markov chain model to develop a theory of convergence for GAs, analogous to the simulated annealing convergence proof. A variable probability-of-mutation parameter is assumed as an analog to the temperature parameter of simulated annealing.

6. The model

The model of Boltzmann tournament selection examined in this paper differs in several ways from the models described in the previous section. It simulates a nontraditional selection procedure without the incorporation of other genetic operators. More importantly, rather than representing the bit or genotypic level, the model represents a more general, equivalence-class level. Since crossover and mutation are not included, the composition of a particular class need not be specified. Also, during high-temperature simulation, the fitness assignment to a particular class is irrelevant; one need only assign fitness values to classes so that they are distinguishable by fitness. The model is flexible enough, however, to simulate lower temperatures and different class fitnesses, as demonstrated in subsequent calculations. This BTS model is a multiclass model, with more representational power than specialized, two-class models; however, this power is gained at the expense

of increased complexity.

Three assumptions were made in modeling BTS. The first, which Goldberg made also in his 1990 debut of BTS, is that classes can be distinguished solely by fitness. Hence, if two individuals differ in fitness by Θ or more, BTS assumes that they belong to different classes. If they differ by less than Θ, BTS assumes that they belong to the same class. The second is that Θ always performs perfect discrimination. In real-world problems, class boundaries may sometimes be blurry; or, given points near a boundary, simple thresholding of the difference in fitness (using Θ) may not always be sufficient to discriminate properly. (Accounting for various sources of noise is a topic unto itself [3]). The assumption in this paper is that Θ is small enough so that elements of two distinct classes will never have fitness differentials less than Θ, and large enough so that elements of the same class will never have fitness differentials of Θ or greater. (In fact, due to the third assumption, intraclass fitness differences are always zero.)

The third assumption is that all elements of a class have identical fitness. For GAs that are capable of maintaining separate niches, this assumption is often valid after preliminary convergence has weeded out the less-fit individuals of each class. It is common practice to approximate the fitnesses of all individuals in a particular class with a single "representative" value, perhaps an average or maximum fitness. The model will be exact if all individuals in a class do indeed have the same fitness; otherwise it should still be a good approximation.

This model is capable of representing any class definitions and energy assignments consistent with the preceding three assumptions. This is done by transforming the desired partition into an equivalent one-genotype-per-class representation, and assigning each genotype the energy of the class it represents. It does not matter which genotype is assigned to represent a particular class. For example, suppose we partition the space of all possible 3-bit strings into the following four classes,

$$class_0 = \{000, 111\},\ class_1 = \{001, 110\},$$
$$class_2 = \{010, 101\},\ class_3 = \{011, 100\}$$

$$E(000) = E(111) = 9,\ E(001) = E(110) = .3,$$
$$E(010) = E(101) = 2,\ E(011) = E(100) = 5$$

and we wish to simulate a uniform starting distribution. We would substitute the following one-genotype-per-class definitions into our model, under a uniform starting distribution, this time using 2-bit strings.

$$class_0 = \{00\},\ class_1 = \{01\},\ class_2 = \{10\},\ class_3 = \{11\}$$

$$E(00) = 9,\ E(01) = .3,\ E(10) = 2,\ E(11) = 5$$

The two representations are equivalent, in the sense that class energies and the number of representatives of each class in the initial population are the same in both. Expected absorption time remains the same in the second representation, as do other properties of the Markov chain (such as the amount of time spent in each transient state prior to absorption).

Using basic properties of finite absorbing Markov chains, we can compute the states to which BTS is expected to converge, and the expected time required to reach one of them. However, before we can compute such long-term results, we must set up the transition matrix, and fill it with the required transition probabilities. Algorithms have been designed for enumerating all possible distributions of classes, and for computing the transition probabilities. The corresponding programs work for any population size, number of starting classes, initial class distribution, temperature, objective function, and level of absorption (loss of all but *ABS* classes). These algorithms and their derivations are outlined in the remainder of this section.

6.1 Automatic enumeration of all possible states

Given a population of size n, and number of classes c, there are

$$\frac{(n+c-1)!}{n!\,(c-1)!}$$

possible states [8]. Classes will be numbered from 0 to $c-1$, and will be called $class_0$, $class_1$, and so on. States will be written as $I_0/I_1/I_2/\ldots/I_{c-1}$, where I_i represents the number of population members in $class_i$. For example, if $n = 3$ and $c = 4$, 0/0/0/3 represents a population where all three individuals belong to $class_3$.

An algorithm which enumerates all states is now given. The enumeration problem is precisely the problem of outputting all elements of a base c number system which meet the constraint that the digits of each number must sum to n.

ENUMERATE $(lhs, n, digits)$
    if $digits = 1$
      output(**concatenate**(lhs, n))
    else
      for i from 0 to n
        ENUMERATE(**concatenate**$(lhs, i), n - i, digits - 1$)

Initial Procedure Call: **ENUMERATE**([ ],n,c)

"[ ]" represents the empty string or list. **concatenate** represents a string concatenation procedure which joins two strings using a "/". **output** represents an appropriate print routine.

6.2 Probabilities of competitors winning tournament

Recall that ind_1, ind_2, and ind_3 are the three competitors specified in Section 2. T is temperature. $E(i)$ is the energy or antifitness of individual i. The probability of keeping ind_2 over ind_3 in the first competition is

$$p' = \frac{1}{1 + e^{[E(ind_3) - E(ind_2)]/T}}$$

The probability of ind_1 winning the second competition assuming it competes against ind_2 is

$$p'' = \frac{1}{1 + e^{[E(ind_1) - E(ind_2)]/T}}$$

The probability of ind_1 winning the second competition assuming it competes against ind_3 is

$$p''' = \frac{1}{1 + e^{[E(ind_1) - E(ind_3)]/T}}$$

p_1, p_2, and p_3 are the probabilities that ind_1, ind_2, and ind_3, respectively, will win the overall (3-way) competition. Note that $p_1 + p_2 + p_3 = 1$.

$p_1 = p'p'' + (1 - p')p'''$ 2 cases: competition against ind_2, and against ind_3
$p_2 = p'(1 - p'')$ ind_2 must win both competitions to advance
$p_3 = (1 - p')(1 - p''')$ ind_3 must win both competitions to advance

At high temperatures, as $T \to \infty$, $p_1 \to 1/2$, $p_2 \to 1/4$, and $p_3 \to 1/4$.

6.3 Probability of being selected as a competitor, then winning

Full transition probabilities can now be computed via two methods, the choice of method depending on whether we sample the first competitor with or without replacement. Sampling ind_1 with replacement is a much higher variance procedure than sampling without replacement, so it tends to lose classes faster. (This observation was verified through Markov chain calculations using this method, as well as by running BTS with replacement on ind_1.) Since sampling with replacement is inferior, results using replacement are not presented. The calculations with replacement are included here for completeness, and because they motivated the eventual calculations without replacement. All variables or symbols are either defined as they appear, or are consistent with those used earlier in this paper.

6.3.1 Calculation I: ind_1 chosen randomly with replacement

Suppose ind_1 is chosen from some class that we will call $class_a$, and ind_2 is chosen from some other class that we will call $class_b$ ($ind_1 \neq ind_2, a \neq b$). The likelihood of these choices occurring is $(I_a/n)\,(I_b/(n-I_a))$, where I_i, as defined earlier, is the number of population elements in $class_i$. Note that $\sum_{i=0}^{c-1} I_i = n$, and $\sum_{a=0}^{c-1} \sum_{b=0(b\neq a)}^{c-1} (I_a/n)\,(I_b/(n-I_a)) = 1$. For each $class_k$ ($k = 0$ to $c-1$), we wish to calculate $P_{k,a,b}$, the probability that some element in $class_k$ wins a particular trial (advances to the next generation), given the above choices for ind_1 and ind_2. A trial winner must first be selected as one of the three competing individuals, then win the tournament. There are n trials in all.

IF $a = k$, $class_k$ can only win the trial using ind_1

$$P_{k,a,b} = p_1$$
$$P_{k,a,b} \to \frac{1}{2} \text{ at high temperatures}$$

ELSE IF $b = k$, $class_k$ can win the trial using ind_2 under strict choice, or using ind_2 or ind_3 under relaxed choice[1]

$$P_{k,a,b} = p_2 + \left(\frac{p_3}{2}\right)\left(\frac{I_k}{n-I_a}\right)$$
$$P_{k,a,b} \to \frac{1}{4} + \left(\frac{1}{8}\right)\left(\frac{I_k}{n-I_a}\right) \text{ at high temperatures}$$

ELSE ($a \neq k$, $b \neq k$), $class_k$ must win the trial using ind_3[2]

$$P_{k,a,b} = \left(\frac{p_3 I_k}{2}\right)\left[\frac{1}{n-I_a-I_b} + \frac{1}{n-I_a}\right]$$
$$P_{k,a,b} \to \left(\frac{I_k}{8}\right)\left[\frac{1}{n-I_a-I_b} + \frac{1}{n-I_a}\right] \text{ at high temperatures}$$

As partial verification of the above derivations, we can suppose that a and b are fixed, and then calculate

$$\sum_{k=0}^{c-1} P_{k,a,b} = P_{a,a,b} + P_{b,a,b} + \sum_{k=0(k\neq a,k\neq b)}^{c-1} P_{k,a,b} = 1$$

Note that the above calculations of the $P_{k,a,b}$ assume that we are given a and b. Dispensing with this assumption, we can calculate for each k the probability of some $class_k$ element winning a trial from scratch, $P(k)$, as the

[1]Derivation: A combination of the probabilities from the three ways of winning: (1) as ind_2 under strict choice, $p_2/2$; (2) as ind_2 under relaxed choice, $p_2/2$; (3) as ind_3 under relaxed choice, $(p_3/2)/(I_k/(n-I_a))$; probability of strict choice = probability of relaxed choice = 1/2.

[2]Derivation: A combination of the probabilities of winning as ind_3 under strict and under relaxed choice.

sum over all legal (a, b) pairs of "$P_{k,a,b}$ times the probability of the (a, b) pair occurring," as follows.

$$P(k) = \sum_{a=0}^{c-1} \sum_{b=0(b\neq a)}^{c-1} \left(\frac{I_a}{n}\right)\left(\frac{I_b}{n - I_a}\right) P_{k,a,b}$$

The $P(k)$ represent independent events, in that only one class can have a winner per trial ($\sum_{k=0}^{c-1} P(k) = 1$), and successive trials (of which there are n total) are independent; that is, they do not depend on the outcome of any previous trials. Through the multinomial distribution of $P(0) \ldots P(c-1)$, we can calculate the probability of going from any initial class distribution $I_0/I_1/\ldots/I_{c-1}$ to any target distribution $I'_0/I'_1/\ldots/I'_{c-1}$. We fill our transition matrix with all such transition probabilities.

6.3.2 Calculation II: ind_1 chosen without replacement

Each individual in the population will get exactly one trial as ind_1. As previously noted, this is a more stable method than that of Calculation I, and is the method used in further analysis. The calculations are actually simpler than those with replacement, as they are now divided into many more cases. The algorithm operates on a starting distribution $I_0/I_1/\ldots/I_{c-1}$, and fills the corresponding row of the transition matrix as follows.

I. For each $class_a$, $a = 0$ to $c - 1$, assume that ind_1 is chosen from that class.

1. For each $class_k$, $k = 0$ to $c - 1$, calculate the probability of some element of that class advancing, for all legal choices of ind_2 (call its class $class_b$) and of ind_3 (call its class $class_d$). Neither ind_2 nor ind_3 can equal ind_1 ($a \neq b$,$a \neq d$). The computation goes as follows.

   CASE $a = k$ ($b \neq k$, $d \neq k$):
   $P(\textit{some } k \textit{ advances given case}) = p_1$ (.5 as $T \to \infty$)

   CASE $b = k$, $d = k$ ($a \neq k$):
   $P(\textit{some } k \textit{ advances given case}) = p_2 + p_3$ (.5 as $T \to \infty$)

   CASE $b = k$, $d \neq k$ ($a \neq k$):
   $P(\textit{some } k \textit{ advances given case}) = p_2$ (.25 as $T \to \infty$)

   CASE $b \neq k$, $d = k$ ($a \neq k$):
   $P(\textit{some } k \textit{ advances given case}) = p_3$ (.25 as $T \to \infty$)

   CASE $b \neq k$, $d \neq k$ ($a \neq k$):
   $P(\textit{some } k \textit{ advances given case}) = 0$ (0 as $T \to \infty$)

   Also calculate the probability of the particular case (of having chosen ind_2 from $class_b$ and ind_3 from $class_d$), still under the

assumption that ind_1 is from $class_a$:

$$\text{IF } b \neq d,\ P(case) = \left(\frac{1}{2}\right)\left(\frac{I_b}{n - I_a}\right)\left[\frac{I_d}{n - I_a - I_b} + \frac{I_d}{n - I_a}\right]$$

$$\text{IF } b = d,\ P(case) = \left(\frac{1}{2}\right)\left(\frac{I_b}{n - I_a}\right)^2$$

($b = d$ occurs only under relaxed choice)

2. For each k, compute the probability that some element of $class_k$ advances (still given that ind_1 comes from $class_a$) as the sum (over all legal choices of ind_2 and ind_3) of products, as follows.

$$padvance(k) = \sum_{b=0(b\neq a)}^{c-1} \sum_{d=0(d\neq a)}^{c-1} P(case)P(some\ k\ advances\ given\ case)$$

3. Now enumerate all possible resulting distributions from those trials in which $ind_1 \in class_a$. For example, if the starting distribution contains two elements from $class_a$, and $c = 3$, the enumerations are 2/0/0, 1/1/0, 1/0/1, 0/2/0, 0/1/1, and 0/0/2.

4. Calculate the probability of occurrence of each such resulting distribution using the multinomial distribution of the *padvance* probabilities.

5. In the final step, "add" the results into cumulative probabilities. These probabilities start at 0, and accumulate the results from each of the outermost loop's assignments to a. This works like a cross product. For example, if the accumulator currently holds probabilities for 5/5/5 and 5/4/6 (among other things), and the new calculation holds probabilities for 1/1/1, 1/2/0, and so on, the new accumulator state $p(6/6/6)$ is assigned as $p(5/5/5) \times p(1/1/1) + p(5/4/6) \times p(1/2/0) + \cdots$. This is correct because, for 6/6/6 to occur, either (5/5/5 and 1/1/1) or (5/4/6 and 1/2/0) (or other combinations that yield 6/6/6) must occur.

II. Assign a row of the transition matrix using the corresponding probabilities in the accumulator.

7. Results

This section compares the expected long-term behavior of BTS under various starting conditions (from the Markov chain model without replacement) with results from actual runs. Tables 1 through 3 and Figure 1 show the expected number of generations until all representatives of all but *ABS* classes are lost, assuming BTS starts with the stated number of classes c, and runs at high temperature T. Results are given for the cases where population elements are initially uniformly distributed among classes. In cases where n is not evenly divisible by c (such as $n = 4$, $c = 3$), an initial distribution as close as possible to uniform is used (such as 1/1/2). Note that at high

n	3 start classes	4 classes	5 classes	6 classes
4	1.864	2.418		
6	3.873	4.844	5.443	5.859
8	7.698	9.599	10.275	
10	15.175	17.946	19.233	
12	29.678	33.768		
18	226.567			
24	1,828.879			
36	129,118.006			

Table 1: The expected number of generations until all but 2 classes are lost, for various population sizes at $T = \infty$.

n	4 start classes	5 classes	6 classes
4	1.149		
6	1.659	2.188	2.596
8	2.641	3.360	
10	3.787		
12	5.411		

Table 2: The expected number of generations until all but 3 classes are lost, for various population sizes at $T = \infty$.

temperatures, permutations (such as 1/1/2, 1/2/1, 2/1/1) all have the same expected arrival time to an absorbing state. Such uniformly distributed initial states simulate the random starting conditions of a GA and, at least at high temperatures, take somewhat longer to lose classes. Values not given in Tables 1 through 3 involve prohibitive calculations. (One of the longest calculations performed was for $n = 10$, $c = 5$, $ABS = 2$. It required computing transition probabilities for, and inverting, a 906×906 matrix.)

To check the results of the Markov chain calculations, BTS was run at high temperature ($T = 5000.0$), under various population sizes, using three starting classes ($c = 3$). Each run was stopped at the point it lost one class, and the number of generations to this point was recorded. 5000 runs were performed for each n, and the mean and standard deviation of the time to absorption were recorded. The results are summarized in Table 4.

Table 4 shows that the Markov chain exactly models the BTS process at high temperatures. This can be seen by comparing the mean values, or by examining the tight confidence interval in which the true mean falls. The high variance in the number of generations to absorption should also be noted; the distribution of the sample number of generations to absorption

n	5 start classes	6 classes
6	1.179	1.435
8	1.509	
10	2.141	
12	2.722	

Table 3: The expected number of generations until all but 4 classes are lost, for various population sizes at $T = \infty$.

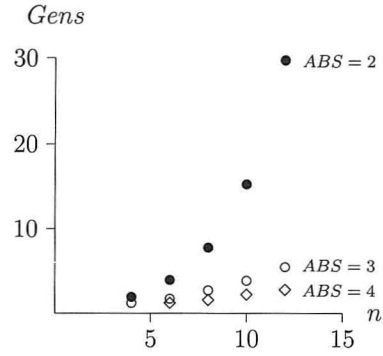

Figure 1: The expected number of generations to absorption, *Gens*, as a function of population size, n. Absorption occurs when representatives of only ABS classes remain. Initially, representatives of $ABS + 1$ classes are present in the population.

highly resembles an exponential distribution. BTS was also run with $n = 36$, although fewer than 5000 runs were performed. Again, results were similar to those obtained from the corresponding Markov chains.

The results of low-temperature Markov simulations ($T = 2.0$), each verified by 5000 runs of BTS, appear in Table 5. The low-temperature model simulates three starting classes, having energies of 0.0, 1.0, and 2.0. At lower temperatures, as at higher temperatures, the Markov chain was found to exactly model the BTS procedure. Initial distributions are again uniform, or as close as possible to uniform. Nonuniform distributions have one extra element apportioned to classes, in order of increasing energy.

As expected, time-to-absorption decreases as temperature is lowered, with lower-energy classes favored according to the Boltzmann distribution. Due to this bias, distributions that start or become biased in favor of lower-energy individuals drift faster than distributions that are uniform or are skewed towards higher-energy individuals. However, as the population size grows, the starting distribution has a decreasing effect on absorption time; BTS increasingly tends to restore the population towards the Boltzmann distribution, though with high variance.

n	Sample Mean	Expected Mean	95% Confidence Interval	Sample StDev
4	1.8304	1.8641	$1.80 < \mu < 1.86$	1.0429
6	3.8290	3.8727	$3.74 < \mu < 3.91$	3.0595
8	7.6674	7.6981	$7.48 < \mu < 7.86$	6.9249
10	14.8960	15.1752	$14.50 < \mu < 15.29$	14.2277
12	30.1126	29.6777	$29.31 < \mu < 30.91$	28.8529
18	231.2022	226.5667	$224.71 < \mu < 237.69$	234.2339
24	1810.4540	1828.8790	$1759.29 < \mu < 1861.62$	1845.8005

Table 4: The sample mean time to absorption from BTS runs at high temperature ($T = 5000$), compared with the expected mean (μ) from the Markov chain model ($c = 3$, $ABS = 2$, $T = \infty$).

n	Sample Mean	Expected Mean	95% Confidence Interval	Sample StDev
4	1.6704	1.6997	$1.64 < \mu < 1.70$	1.0878
6	3.3388	3.3140	$3.26 < \mu < 3.41$	2.7033
8	5.3716	5.3563	$5.24 < \mu < 5.50$	4.6513
10	8.5318	8.4654	$8.32 < \mu < 8.74$	7.6720
12	13.5386	13.3931	$13.20 < \mu < 13.88$	12.2385
18	46.0360	45.9392	$44.79 < \mu < 47.28$	44.8253
24	159.0438	157.8056	$154.74 < \mu < 163.35$	155.2366
36	1938.4614	1930.9141	$1885.43 < \mu < 1991.49$	1913.0984

Table 5: The sample mean time to absorption from BTS runs at low temperature ($T = 2.0$), compared with the expected mean (μ) from the Markov chain model ($c = 3$, $ABS = 2$, $T = 2.0$, $x \in class_i \Rightarrow E(x) = i$).

8. Conclusion

This paper has presented the results of a study of finite Markov chain models of the Boltzmann tournament selection strategy. One of the models was used to predict the effects of genetic drift on the selection strategy. Predictions were verified through direct simulation, and corresponded almost perfectly to actual results. The general concept of a class (i.e., equivalence class) was used in the modeling. This represents a significant departure from the string-level modeling of previous studies. Methods for enumerating and computing transition probabilities were developed, and outlined in detail.

The Markov chain models of BTS suggest that the loss of classes is a problem that must be taken into consideration. The high variance in the sample number of generations to absorption indicates that our expectation for a particular run may vary significantly from the mean. The Markov chain models also suggest that the initial class distribution does not greatly affect

the time to absorption, as long as the distribution is not greatly skewed. It would be useful to explore further the hypothesis that the effect of the initial class distribution becomes negligible as population size increases. The most practical cases are those in which there are far more classes than population members. The limiting value on expected time to absorption as c approaches n is a crucial value that could be either calculated or estimated.

One suggestion often made for traditional genetic algorithms is to raise the population size to increase initial diversity. Likewise, BTS benefits from larger population sizes, as demonstrated by the increase in time-to-absorption for larger populations.

Although this analysis has been primarily devoted to behavior at high temperatures, it also reveals much about expected low-temperature behavior. If BTS loses classes too rapidly at high temperatures, there is little hope for maintaining them at low temperatures, where selection pressure is great and the population has already partially converged to a distribution biased in favor of lower-energy individuals. Conversely, if BTS can achieve a stable Boltzmann distribution at a high temperature, it stands a better chance of achieving this distribution at low temperatures. Hence, the high-temperature model gives us the best-case expected drift behavior of BTS. High-temperature models have the additional benefit of being able to ignore the objective function. To construct low-temperature models, one can employ simple objective functions that assign all elements in a particular class the same function value.

One prominent observation resulting from this study is that incorporating high population sizes or large numbers of classes into the model leads to prohibitive growth in the size of the transition matrix, and in the amount of computation required to determine properties of the finite Markov chain (such as time to absorption). One possible improvement would be to model *steady-state* rather than generational GAs [9]. Because steady-state GAs process only one individual at a time, the resulting transition matrix will be sparse. Another possibility would be to define fewer classes, perhaps by combining similar classes or broadening the definition of a class. The two-class case is easily modeled by both Markov chains and diffusion equations; we need track only one class, because the other must contain all other population elements. A third possibility, if classes are in some sense equivalent (so that we would not be interested in the number of population members in any particular class, but in the degree of nonuniformity), would be to reduce the size of the transition matrix by combining permutations.

A different approach would be to let the states of the transition matrix represent proportions of classes, using some predetermined resolution. The advantage of this approach is that the size of the transition matrix does not depend upon the population size, but on the chosen resolution. The disadvantage is that finite algorithmic properties (such as variance) are no longer modeled. Finally, another approach would be to not construct the transition matrix, but to derive a general formula for a transition probability, and to use this formula in subsequent derivations.

In conclusion, we observe that the analysis of BTS opens up many possibilities. Not only does it tell us what to expect when running BTS, but it can also help in modeling other genetic algorithms (such as multimodal function optimizers), or in designing improved, related algorithms.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant ECS-9022007. I would like to thank David Goldberg for much valuable discussion related to this work.

References

[1] T. E. Davis and J. C. Principe, "A Simulated Annealing Like Convergence Theory for the Simple Genetic Algorithm," pages 174–181 in *Proceedings of the Fourth International Conference on Genetic Algorithms* (1991).

[2] D. E. Goldberg, "A Note on Boltzmann Tournament Selection for Genetic Algorithms and Population-oriented Simulated Annealing," *Complex Systems*, **4** (1990) 445–460.

[3] D. E. Goldberg, K. Deb, and J. H. Clark, "Accounting for Noise in the Sizing of Populations," pages 127–140 in *Foundations of Genetic Algorithms, Volume 2*, edited by D. Whitley (San Mateo: Morgan Kaufmann, 1993).

[4] D. E. Goldberg and J. J. Richardson, "Genetic Algorithms with Sharing for Multimodal Function Optimization," pages 41–49 in *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms* (1987).

[5] D. E. Goldberg and P. Segrest, "Finite Markov Chain Analysis of Genetic Algorithms," pages 1–8 in *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms* (1987).

[6] J. G. Kemeny and J. L. Snell, *Finite Markov Chains* (Princeton: Van Nostrand, 1960).

[7] M. L. Lial and C. D. Miller, *Finite Mathematics* (Glenview, IL: Scott, Foresman, 1989).

[8] A. E. Nix and M. D. Vose, "Modeling Genetic Algorithms with Markov Chains," *Annals of Mathematics and Artificial Intelligence*, **5** (1992) 79–88.

[9] G. Syswerda, "A Study of Reproduction in Generational and Steady-state Genetic Algorithms," pages 94–101 in *Foundations of Genetic Algorithms*, edited by G. Rawlins (San Mateo: Morgan Kaufmann, 1991).

[10] M. D. Vose, "Modeling Simple Genetic Algorithms," pages 63–73 in *Foundations of Genetic Algorithms, Volume 2*, edited by D. Whitley (San Mateo: Morgan Kaufmann, 1993).