# Markov Models for Biogeography-Based Optimization

Dan Simon, *Senior Member, IEEE*, Mehmet Ergezer, *Member, IEEE*, Dawei Du, and Rick Rarick

***Abstract*—Biogeography-based optimization (BBO) is a population-based evolutionary algorithm that is based on the mathematics of biogeography. Biogeography is the science and study of the geographical distribution of biological organisms. In BBO, problem solutions are analogous to islands, and the sharing of features between solutions is analogous to the migration of species. This paper derives Markov models for BBO with selection, migration, and mutation operators. Our models give the theoretically exact limiting probabilities for each possible population distribution for a given problem. We provide simulation results to confirm the Markov models.**

***Index Terms*—Biogeography-based optimization (BBO), evolutionary algorithms (EAs), Markov models.**

## I. INTRODUCTION

EVOLUTIONARY algorithms (EAs) are a growing field, commonly used for global optimization. Biogeography-based optimization (BBO) is a new EA and was first presented in [1] as an application of the mathematics of biogeography [2], [3] to evolutionary computation. BBO is an example of how a natural process can be modeled to solve general optimization problems. Ongoing research provides empirical evidence of the potential of BBO compared to other evolutionary computing algorithms [4]–[7]; however, as with most other EAs, there are limited theoretical results for BBO [8]. This paper derives a Markov chain model for BBO that can help in understanding its convergence and performance properties.

Markov models have already been developed for other EAs, such as simple genetic algorithms [9], [10] and simulated annealing [11]. Due to the unique migration mechanism in BBO (discussed in Section II), we need to use the generalized multinomial theorem [12] in this paper to derive a Markov model for BBO's selection, migration, and mutation operators.

A Markov chain is a random process that has a discrete set of possible state values $s_i$ $(i = 1, \dots, T)$ [13, Ch. 11]. The probability that the system transitions from state $s_i$ to $s_j$ is given by the probability $p_{ij}$, which is called a transition probability. The $T \times T$ matrix $P = [p_{ij}]$ is called the transition matrix. A Markov chain is called regular if it is possible to go from any state to any other state (not necessarily in one step). The fundamental limit theorem for regular Markov chains states that if $P$ is regular, then

$$\lim_{n\to\infty} P^n = P_{ss} \tag{1}$$

where each row $p_{ss}$ of $P_{ss}$ is the same. The $i$th element of $p_{ss}$ denotes the probability that the Markov chain is in state $s_i$ after an infinite number of transitions. $p_{ss}$ is independent of the initial state.

As applied to BBO, a Markov state represents a BBO population distribution. The probability $p_{ij}$ is the probability that the population transitions from the distribution $s_i$ to the distribution $s_j$ after one generation. If the mutation rate is nonzero, this probability is greater than zero, which means that the transition matrix is regular. This means that there is a unique nonzero limiting probability for each possible population distribution as the number of generations approaches infinity.

If BBO does not incorporate mutation, then it may converge to a uniform population, i.e., a population in which each individual is identical. This type of Markov chain is called absorbing [13, Ch. 11]. In this case, we can calculate the probability that the population will converge to each state, and the expected time to convergence. We do not consider BBO with zero mutation in this paper, but the mathematical foundation that we lay allows this variation to be explored in future research.

Section II gives an introduction to BBO. Section III derives Markov models for BBO, which allows us to obtain the limiting probability (as the generation count approaches infinity) of all possible populations. Section IV gives a simple simulation to confirm the Markov model. We provide some concluding remarks and directions for future work in Section V. The appendices give a review of generalized multinomial probability, and three different expressions for the dimension of the BBO population transition matrix.

## II. BBO

Suppose that we have a set of candidate solutions to some problem. Each candidate solution is defined by specific features. BBO is based on the idea of probabilistically sharing features between solutions based on the solutions' fitness values. In BBO, if a copy of feature $s$ from solution $x$ replaces one of the features in solution $y$, we say that $s$ has emigrated from $x$ and immigrated to $y$.

The probability that solution $x$ shares its features with some other individual in the population is proportional to the fitness of $x$. The probability that solution $y$ receives a feature from some other individual in the population decreases with the

Manuscript received November 27, 2009; revised March 15, 2010; accepted May 8, 2010. This paper was recommended by Associate Editor J. Wang.

The authors are with the Department of Electrical and Computer Engineering, Cleveland State University, Cleveland, OH 44115 USA (e-mail: d.j.simon@csuohio.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCB.2010.2051149

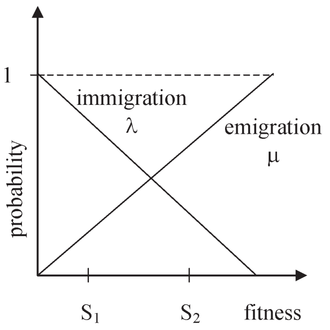

Fig. 1. Illustration of two candidate solutions to some problem using symmetric migration curves. $S_1$ is a relatively poor solution, and $S_2$ is a relatively good solution. $S_1$ has high immigration and low emigration, which means that it is likely to receive features from other solutions but unlikely to share features with other solutions. $S_2$ has low immigration and high emigration, which means that it is unlikely to receive features from other solutions but likely to share features with other solutions.

fitness of $y$. We base these migration probabilities on curves, such as those shown in Fig. 1. For the sake of simplicity, we assume that all solutions have identical migration curves. Fig. 1 shows two solutions in BBO. $S_1$ represents a poor solution, and $S_2$ represents a more fit solution. The immigration probability for $S_1$ will therefore be higher than the immigration probability for $S_2$. The emigration probability for $S_1$ will be lower than the emigration probability for $S_2$.

As with every other EA, each solution might also have some probability of mutation. In this paper, mutation is implemented in a standard way. We deal with discrete optimization problems, so each solution feature is either a 0 or a 1. The probability of mutation for BBO is defined as a constant $p_m \in [0, 1]$. At each generation and for each feature in each solution, we generate a uniformly distributed random number $r \in [0, 1]$. If $r < p_m$, then we mutate (i.e., complement) the bit under consideration.

Also, similar to other population-based algorithms, we often incorporate elitism in BBO in order to retain the best solutions in the population from one generation to the next. This prevents the best solutions from being corrupted by immigration or mutation. Elitism can be implemented by setting the immigration rate $\lambda$ equal to zero for the $\alpha$ best solutions, where $\alpha$ is a user-selected elitism parameter. Elitism is not used in this paper but was modeled in [14].

There are several different ways to implement the details of BBO, but in this paper, we use the original BBO formulation [1], which is called partial immigration-based BBO in [8]. In this approach, for each feature in each solution, we probabilistically decide whether to immigrate. If immigration is selected for a given feature, then the emigrating solution is probabilistically selected based on fitness (e.g., using roulette wheel selection). This gives the algorithm shown in Fig. 2 as a description of one generation of BBO. Migration and mutation of the entire population take place before any of the solutions are replaced in the population, which requires the use of the temporary population $z$ in the algorithm.

## III. MARKOV MODELS FOR BBO

A Markov chain model provides us with the probability $p_{ij}$ of transitioning from state $s_i$ to $s_j$. This probability is, by definition, independent of how the system reached state $s_i$. All of the transition probabilities can be used to form the transition matrix $P = [p_{ij}]$. In this section, we derive a Markov model of BBO based on its selection, migration, and mutation operators.

Suppose that we have a problem whose solutions are in a binary search space. The possible solutions are represented by all bit strings $x_i$ consisting of $q$ bits each. Therefore, the cardinality of the search space is $n = 2^q$. We use $N$ to denote the population size, and we use $v$ to denote the population vector, where $v_i$ is the number of $x_i$ individuals in the population. We see that

$$\sum_{i=1}^{n} v_i = N. \tag{2}$$

We use $y_k$ to denote the $k$th individual in the population. The population of the search algorithm can be depicted as

$$\begin{aligned}\text{Population}\\ &= \{y_1, \ldots, y_N\}\\ &= \{\underbrace{x_1, x_1, \ldots, x_1}_{v_1 \text{ copies}}, \underbrace{x_2, x_2, \ldots, x_2}_{v_2 \text{ copies}}, \ldots, \underbrace{x_n, x_n, \ldots, x_n}_{v_n \text{ copies}}\}\end{aligned} \tag{3}$$

where the $y_i$'s have been ordered to group identical individuals. We use $\lambda_i$ to denote the immigration probability of $x_i$, and $\mu_i$ to denote the emigration probability of $x_i$. Note that $\mu_i$ is proportional to the fitness of $x_i$, and $\lambda_i$ decreases with the fitness of $x_i$. We use the notation $x_i(s)$ to denote the $s$th bit of solution $x_i$. We use the notation $\mathcal{J}_i(s)$ to denote the set of population indices $j$ such that the $s$th bit of $x_j$ is equal to the $s$th bit of $x_i$. That is

$$\mathcal{J}_i(s) = \{j : x_j(s) = x_i(s)\}. \tag{4}$$

We order $y_k$ in the same order as $x_i$. That is

$$y_k = \begin{cases} x_1, & \text{for } k = 1, \ldots, v_1\\ x_2, & \text{for } k = v_1 + 1, \ldots, v_1 + v_2\\ x_3, & \text{for } k = v_1 + v_2 + 1, \ldots, v_1 + v_2 + v_3\\ \vdots & \vdots\\ x_n, & \text{for } k = \sum_{i=1}^{n-1} v_i + 1, \ldots, N. \end{cases} \tag{5}$$

This is also shown in (3) and can be written more compactly as

$$y_k = x_{m(k)}, \qquad \text{for} \quad k = 1, \ldots, N \tag{6}$$

where $m(k)$ is defined as

$$m(k) = \min r, \qquad \text{such that} \quad \sum_{i=1}^{r} v_i \geq k. \tag{7}$$

If we need to denote the generation number of the algorithm, we use an additional subscript. For example, $y_k(s)_t$ is the value of the $s$th bit of the $k$th individual at generation $t$.

```
$z \leftarrow y$
Define emigration probability $\mu_k \propto$ fitness of the $k$-th solution
Define immigration probability $\lambda_k = 1 - \mu_k$
For each solution $z_k$
    For each solution feature $s$
        Use $\lambda_k$ to probabilistically decide whether to immigrate to $z_k$
        If immigrating then
            Use the $\mu$ values to probabilistically select the emigrating solution $y_j$
            $z_k(s) \leftarrow y_j(s)$
        end if
        Probabilistically decide whether to mutate $z_k(s)$
    next solution feature
next solution
$y \leftarrow z$
```

Fig. 2. One generation of the BBO algorithm. $y$ is the entire population of candidate solutions, $y_k$ is the $k$th candidate solution, and $y_k(s)$ is the $s$th feature of $y_k$.

*Example:* Suppose that we have a two-bit problem ($q = 2, n = 4$) with a population size $N = 3$. The search space consists of the bit strings $x = \{x_1, x_2, x_3, x_4\} = \{00, 01, 10, 11\}$. Suppose that the three individuals in the current population are $y = \{x_2, x_2, x_4\} = \{01, 01, 11\}$. Then, we have $v_1 = 0$, $v_2 = 2$, $v_3 = 0$, and $v_4 = 1$.

Let us consider the derivation of $\mathcal{J}_1(1)$. We arbitrarily number bits from left to right, i.e., in any given bit string, bit 1 is the leftmost bit, and bit 2 is the rightmost bit. From (4), we see that

$$\mathcal{J}_1(1) = \{j : x_j(1) = x_1(1)\}\,. \tag{8}$$

Since $x_1 = 00$, we see that $x_1(1) = 0$ (i.e., the leftmost bit). Then, (8) can be written as

$$\mathcal{J}_1(1) = \{j : x_j(1) = 0\}\,.$$

However, $x_j(1) = 0$ for $x_j \in \{00, 01\}$, which, in turn, indicates that $j \in [1, 2]$; therefore, $\mathcal{J}_1(1) = \{1, 2\}$. Continuing this process, we see that

$$\begin{aligned} \mathcal{J}_1(1) &= \{1, 2\}, \quad \mathcal{J}_1(2) = \{1, 3\} \\ \mathcal{J}_2(1) &= \{1, 2\}, \quad \mathcal{J}_2(2) = \{2, 4\} \\ \mathcal{J}_3(1) &= \{3, 4\}, \quad \mathcal{J}_3(2) = \{1, 3\} \\ \mathcal{J}_4(1) &= \{3, 4\}, \quad \mathcal{J}_4(2) = \{2, 4\}. \end{aligned}$$

### A. *Migration*

We make some assumptions in the Markov model development in this section. First, all of the new BBO solutions are created before any solutions are replaced in the population, i.e., we use a generational BBO algorithm rather than a steady-state BBO algorithm. This is clear from the use of the temporary population $z$ in Fig. 2.

Second, a solution can emigrate a bit to itself. This means that, in the statement "use the $\mu$ values to probabilistically select the emigrating solution $y_j$" in Fig. 2, $j$ might be chosen to be equal to $k$. That is, when a bit is replaced via migration in a given solution $z_k$, the new bit might be chosen to come from $z_k$ itself. In this case, the bit is not actually replaced in $z_k$. However, the probabilistic choice of the emigrating solution allows this to happen on occasion.

Third, the migration rates $\lambda$ and $\mu$ are independent of the population distribution, i.e., absolute fitness values are used to obtain $\lambda$ and $\mu$, as opposed to a rank-based fitness. Alternatives to these assumptions will change the Markov model development of this section, but this is left for future work.

If the $s$th feature of $y_k$ is not selected for immigration during generation $t$, then

$$y_k(s)_{t+1} = x_{m(k)}(s) \quad \text{(immigration did not occur)}. \tag{9}$$

That is, $y_k(s)$ does not change from generation $t$ to generation $t + 1$. However, if the $s$th feature of $y_k$ is selected for immigration during generation $t$, then the probability that $y_k(s)_{t+1}$ is equal to $x_i(s)$ is proportional to the combined emigration rates of all individuals whose $s$th feature is equal to $x_i(s)$. This probability can be written as

$$\begin{aligned} &\Pr\nolimits_{\text{imm}} \left(y_k(s)_{t+1} = x_i(s)\right) \\ &\qquad = \frac{\sum_{j \in \mathcal{J}_i(s)} v_j \mu_j}{\sum_{j=1}^{n} v_j \mu_j} \quad \text{(immigration occurred)}. \end{aligned} \tag{10}$$

We can combine (9) and (10), along with the fact that the probability of immigration to $y_k(s)$ is equal to $\lambda_{m(k)}$, to obtain the total probability

$$\begin{aligned} &\Pr\left(y_k(s)_{t+1} = x_i(s)\right) \\ &\quad = \Pr(\text{no immigration}) \\ &\qquad \times \Pr\left(y_k(s)_{t+1} = x_i(s) | \text{no immigration}\right) \\ &\qquad + \Pr(\text{immigration}) \\ &\qquad \times \Pr\left(y_k(s)_{t+1} = x_i(s) | \text{immigration}\right) \\ &\quad = \left(1 - \lambda_{m(k)}\right) \mathbf{1}_0 \left(x_{m(k)}(s) - x_i(s)\right) \\ &\qquad + \lambda_{m(k)} \frac{\sum_{j \in \mathcal{J}_i(s)} v_j \mu_j}{\sum_{j=1}^{n} v_j \mu_j} \end{aligned} \tag{11}$$

where $\mathbf{1}_0$ is the indicator function on the set $\{0\}$.

Now, recall that there are $q$ bits in each solution. Use $P_{ki}(v)$ to denote the probability that immigration results in $y_k = x_i$, given that the population is described by the vector $v$. This probability can be written as

$$\begin{aligned} P_{ki}(v) &= \Pr\left(y_{k,t+1} = x_i\right) \\ &= \prod_{s=1}^{q} \left[ \left(1 - \lambda_{m(k)}\right) \mathbf{1}_0 \left(x_{m(k)}(s) - x_i(s)\right) \right. \\ &\qquad \left. + \lambda_{m(k)} \frac{\sum_{j \in \mathcal{J}_i(s)} v_j \mu_j}{\sum_{j=1}^{n} v_j \mu_j} \right]. \end{aligned} \tag{12}$$

$P_{ki}(v)$ can be computed for each $k \in [1, N]$ and each $i \in [1, n]$ in order to form the $N \times n$ matrix $P(v)$. The $k$th row of $P(v)$ corresponds to the $k$th iteration of the outer loop in Fig. 2. The $i$th column of $P(v)$ corresponds to the probability of obtaining island $x_i$ during each outer loop iteration.

The BBO algorithm entails $N$ trials (i.e., $N$ iterations of the outer loop in Fig. 2), where the probability of the $i$th outcome on the $k$th trial is given as $P_{ki}(v)$. We use $u_i$ to denote the total number of times that outcome $i$ occurs after $N$ trials have been completed, and define $u = [u_1 \quad \cdots \quad u_n]^{\mathrm{T}}$. Then, the probability $\Pr(u|v)$ that we obtain a population vector $u$ after one generation, given that we start with a population vector $v$, can be derived from the generalized multinomial theorem [12].

The generalized multinomial theorem gives the probability of obtaining a certain set of experimental outcomes when the probability of each trial is dependent on the trial number. See Appendix A for an overview. The reason that the generalized multinomial theorem applies to BBO is that the probability of obtaining a specific individual $x_i$ in the population depends on the migration trial number $k$, as shown in (12). We can therefore use the generalized multinomial theorem to find $\Pr(u|v)$ as

$$\begin{aligned} \Pr(u|v) &= \sum_{J \in Y} \prod_{k=1}^{N} \prod_{i=1}^{n} \left[P_{ki}(v)\right]^{J_{ki}} \\ Y &= \left\{ J \in \mathbf{R}^{N \times n} : J_{ki} \in \{0,1\}, \sum_{i=1}^{n} J_{ki} = 1 \text{ for all } k, \right. \\ &\qquad \left. \sum_{k=1}^{N} J_{ki} = u_i \text{ for all i} \right\}. \end{aligned} \tag{13}$$

In order to find the probability that the BBO population transitions from $v$ to $u$ after one generation, we find all of the $J$ matrices that satisfy the conditions of (13). For each of these $J$ matrices, we compute the product of products given in (13). We then add up all the product of products to obtain the desired probability.

### B. Mutation

The previous section considered only migration. In this section, we add the possibility of mutation. We use $U$ to denote the $n \times n$ mutation matrix, where $U_{ij}$ is the probability that $x_j$ mutates to $x_i$. The probability that the $k$th immigration trial followed by mutation results in $x_i$ is denoted as $P_{ki}^{(2)}(v)$. This can be written as

$$\begin{aligned} P_{ki}^{(2)}(v) &= \sum_{j=1}^{n} U_{ij} P_{kj}(v) \\ P^{(2)}(v) &= P(v) U^T \end{aligned} \tag{14}$$

where the elements of $P(v)$ are given in (12). $P(v)$ is the $N \times n$ matrix containing the probabilities of obtaining each of $n$ possible individuals at each of $N$ trials, where only migration is considered. $P^{(2)}(v)$ contains those probabilities when both migration and mutation are considered. In this case, we can write the probability of transitioning from population vector $v$ to $u$ after one generation as

$$\Pr{}^{(2)}(u|v) = \sum_{Y} \prod_{k=1}^{N} \prod_{i=1}^{n} \left[P_{ki}^{(2)}(v)\right]^{J_{ki}} \tag{15}$$

where $Y$ is given in (13). Equation (15) can be used to find the transition matrix for BBO with migration and mutation.

The Markov transition matrix $Q$ is obtained by computing (15) for each possible $v$ and each possible $u$. The element $Q_{ij}$ will give the probability of transitioning from population vector $v$ to $u$ after one generation. The matrix $Q$ is therefore a $T \times T$ matrix, where $T$ is the total number of possible populations. That is, $T$ is the number of possible $n \times 1$ integer vectors $v$ whose elements sum to $N$ and each of whose elements $v_i \in [0, N]$. The number $T$ can be calculated in several different ways, as discussed in Appendix B. After we calculate the transition matrix, we can apply a wealth of Markov tools [15] to the transition matrix to find the statistical properties of BBO populations, including the limiting probability of each possible BBO population.

## IV. SIMULATION RESULTS

This section confirms the BBO Markov model with simulation. We use the 3-b one-max problem with a search space cardinality of eight and a population size of four. The one-max problem has a fitness function that is proportional to the number of ones in the population member, and is a popular test function in EA research [16]. From (22) in Appendix B, we calculate the total number of possible populations as

$$T = \binom{n+N-1}{N} = \binom{8+4-1}{4} = 330.$$

Equation (15) can be used to find the limiting population distribution of BBO. This is the probability, in the limit as the generation count approaches infinity, that the BBO population consists of any particular set of individuals.

The fitness values of the 3-b one-max problem are given as

$$\begin{aligned} f(000) &= 1, & f(001) &= 2 \\ f(010) &= 2, & f(011) &= 3 \\ f(100) &= 2, & f(101) &= 3 \\ f(110) &= 3, & f(111) &= 4. \end{aligned} \tag{16}$$

TABLE I

BBO MARKOV MODEL AND SIMULATION RESULTS FOR THE 3-B ONE-MAX PROBLEM. THE TABLE SHOWS THE MOST PROBABLE POPULATIONS, AND THE COMBINED PROBABILITY OF CONVERGENCE TO POPULATIONS THAT CONTAIN NO OPTIMAL SOLUTIONS (*= "don't care" bit). SIMULATION RESULTS ARE THE AVERAGE OF 100 MONTE CARLO RUNS

| Mutation Rate | Population Vector | Probability Markov | Probability Simulation |
|---|---|---|---|
| 0.1 | 0 0 0 0 0 0 1 3 | 0.0290 | 0.0285 |
| | 0 0 0 0 0 1 0 3 | 0.0290 | 0.0284 |
| | 0 0 0 1 0 0 0 3 | 0.0290 | 0.0284 |
| | * * * * * * * 0 | 0.2999 | 0.3026 |
| 0.01 | 0 0 0 0 0 0 0 4 | 0.5344 | 0.5322 |
| | 0 0 0 0 0 0 1 3 | 0.0718 | 0.0715 |
| | 0 0 0 0 0 1 0 3 | 0.0718 | 0.0716 |
| | 0 0 0 1 0 0 0 3 | 0.0718 | 0.0726 |
| | * * * * * * * 0 | 0.1134 | 0.1138 |
| 0.001 | 0 0 0 0 0 0 0 4 | 0.8605 | 0.8437 |
| | 0 0 0 0 0 0 4 0 | 0.0288 | 0.0386 |
| | 0 0 0 0 0 4 0 0 | 0.0288 | 0.0408 |
| | 0 0 0 4 0 0 0 0 | 0.0288 | 0.0380 |
| | * * * * * * * 0 | 0.0923 | 0.1092 |

Table I shows the most probable populations, along with the combined probabilities of the populations that do not contain any optimal solutions. The population vector $\{v_1, v_2, \ldots, v_8\}$ in Table I indicates the numbers of individuals that are equal to $\{000, 001, \ldots, 111\}$, respectively. The Markov model and simulation results match well, which confirms the model. Table I shows that a high mutation rate of 10% per bit results in too much exploration, so the uniform optimal population is not one of the most probable populations—in fact, it is only the seventh most probable population with a probability of 2.5% (not shown in the table). With this high 10% mutation rate, the probability that the population does not have any optimal individuals is 30%, as shown in the table. However, as the mutation rate decreases to the more reasonable values of 1% and 0.1%, the probabilities that the population is composed entirely of optimal individuals increase to 53% and 86%, respectively, and the probabilities that the population has no optimal individuals decrease to 11% and 9%, respectively.

Fig. 3 shows typical simulation results of 20 000 generations of BBO for the 3-b one-max problem with a mutation rate of 1% per bit. It is seen that the uniform optimal population occurs just over 50% of the time, in agreement with Table I.

Our second benchmark is a 3-b deceptive problem, again with a search space cardinality of eight and a population size of four. The fitness values were the same as that of the one-max problem shown in (16), except that the bit string of all zeros had the highest fitness, i.e., $f(000) = 5$. Table II shows the most probable populations, along with the combined probabilities of the populations that do not contain any optimal solutions. Once again, the Markov model and simulation results match well. Table II shows that a high mutation rate of 10% per bit results in too much exploration, resulting in a probability of no optima in the population of over 50%. However, as the mutation rate decreases to the more reasonable values of 1% and 0.1%, the probabilities that the population is composed entirely of nonoptimal individuals decrease to 12% and 6%, respectively.

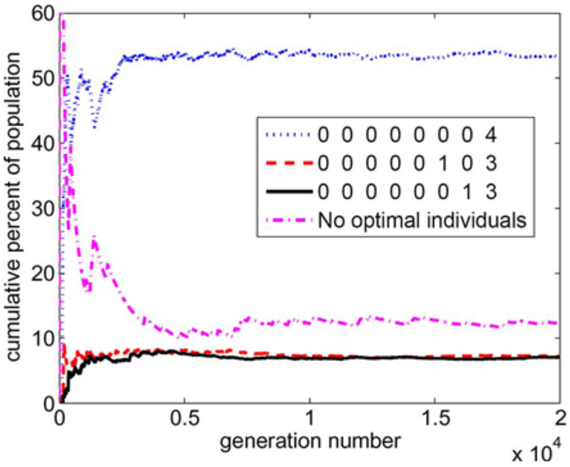

Fig. 3. Typical BBO simulation results for a 3-b one-max optimization problem with a mutation rate of 1% per bit. The three most probable populations are shown, along with the cumulative probability of all populations that have no optimal individuals.

TABLE II

BBO MARKOV MODEL AND SIMULATION RESULTS FOR A 3-B DECEPTIVE PROBLEM. THE TABLE SHOWS THE MOST PROBABLE POPULATIONS, AND THE COMBINED PROBABILITY OF CONVERGENCE TO POPULATIONS THAT CONTAIN NO OPTIMAL SOLUTIONS. SIMULATION RESULTS ARE THE AVERAGE OF 100 MONTE CARLO RUNS

| Mutation Rate | Population Vector | Probability Markov | Probability Simulation |
|---|---|---|---|
| 0.1 | 4 0 0 0 0 0 0 0 | 0.0335 | 0.0336 |
| | 3 0 1 0 0 0 0 0 | 0.0269 | 0.0265 |
| | 3 1 0 0 0 0 0 0 | 0.0269 | 0.0272 |
| | 0 * * * * * * * | 0.5048 | 0.5045 |
| 0.01 | 4 0 0 0 0 0 0 0 | 0.7009 | 0.7029 |
| | 0 0 0 0 0 0 0 4 | 0.0598 | 0.0584 |
| | 3 1 0 0 0 0 0 0 | 0.0506 | 0.0505 |
| | 0 * * * * * * * | 0.1193 | 0.1169 |
| 0.001 | 4 0 0 0 0 0 0 0 | 0.9200 | 0.9227 |
| | 0 0 0 0 0 0 0 4 | 0.0462 | 0.0486 |
| | 3 0 1 0 0 0 0 0 | 0.0066 | 0.0067 |
| | 0 * * * * * * * | 0.0597 | 0.0562 |

Note that the migration curves that we used to derive these results were linear, as shown in Fig. 1. An optimization problem with a search space size of eight and linear migration curves, like the problems explored in this section, could have the following migration values, listed in order from least fit to most fit:

$$\lambda = \{0.9 \quad 0.8 \quad 0.7 \quad 0.6 \quad 0.5 \quad 0.4 \quad 0.3 \quad 0.2\}$$
$$\mu = \{0.1 \quad 0.2 \quad 0.3 \quad 0.4 \quad 0.5 \quad 0.6 \quad 0.7 \quad 0.8\}.$$

If nonlinear migration curves are used in BBO, as suggested in [17], the migration values would change, but the Markov model derived in this paper would remain the same.

## V. CONCLUSION

We have derived a Markov model for BBO. The model gives the theoretical probability of the occurrence of each possible population as the generation count goes to infinity. The theory was confirmed with simulation results.

The Markov model development in this paper is computationally expensive because the size of the Markov transition matrix is $(n+N-1)$-choose-$N$, where $n$ is the cardinality of the search space and $N$ is the population size. Computational savings can be obtained by grouping Markov states together and then computing the probability that the population transitions from one group of populations to another group, as discussed in [15], but this is left for further research. Computational savings could also be obtained by not allowing duplicate individuals in the population. This would require an adjustment to the Markov model and would reduce the size of the transition matrix to $n$-choose-$N$.

Other future work includes extending the Markov model to variations of BBO. This paper focused on the original BBO algorithm, which is called partial immigration-based BBO. An extension of the Markov model in this paper to BBO variations would analytically show their advantages or disadvantages. Some of these variations include partial emigration-based BBO, total immigration-based BBO, total emigration-based BBO [8], and BBO with different migration curve shapes [17]. Also, the Markov model in this paper could be extended to other EAs so that comparisons could be made between EAs theoretically rather than based only on simulations.

The Markov model development in this paper has been restricted to binary problems, i.e., problems in which each solution feature is a bit. Future work could explore the extension of this paper to problems in which the solution features are integers, as in the original BBO paper [1], or to problems in which the solution features are real numbers.

Our current work involves the comparison of BBO and GA Markov models and the use of the Markov model developed here to develop a dynamic system model of BBO. Dynamic system analysis of EAs is used to find the proportion of each possible individual in a population as the population size tends to infinity. This is exemplified by the extension of GA Markov models to dynamic system analysis [15].

## APPENDIX A
## GENERALIZED MULTINOMIAL PROBABILITY

Suppose that an experiment has $n$ possible outcomes $\{a_1, \ldots, a_n\}$ and that the experiment is repeated $N$ times. Suppose that the probability of obtaining outcome $a_i$ on the $k$th trial is equal to $P_{ki}$. Let $C = [C_1, \ldots, C_n]$ be a vector of random variables, where $C_i$ denotes the total number of times that $a_i$ occurs in $N$ trials, and let $\gamma = [\gamma_1, \ldots, \gamma_n]$ be a realization of $C$. Define

$$Y(\gamma) = \left\{ J \in \mathbf{R}^{N\times n} : J_{ki} \in \{0,1\}, \sum_{i=1}^{n} J_{ki} = 1 \text{ for all } k, \right. \left. \sum_{k=1}^{N} J_{ki} = \gamma_i \text{ for all } i \right\}. \tag{17}$$

Note that the cardinality of $Y(\gamma)$ is

$$|Y(\gamma)| = \frac{N!}{\gamma_1! \cdots \gamma_n!}. \tag{18}$$

Then, the generalized multinomial theorem [12] gives the following probability that the repeated experiment results in the outcome vector $\gamma$:

$$\Pr(C=\gamma) = \sum_{J\in Y(\gamma)} \prod_{k=1}^{N} \prod_{i=1}^{n} P_{ki}^{J_{ki}}. \tag{19}$$

*Example:* Prof. Smith submits three papers to three different journals. Each journal has a probability $P_a$ of acceptance, $P_m$ of acceptance with major revisions, $P_n$ of acceptance with minor revisions, and $P_r$ of rejection. The probabilities are given as

$$\text{Journal 1}: \quad P_{1a}=0.1, P_{1m}=0.3, P_{1n}=0.5, P_{1r}=0.1$$

$$\text{Journal 2}: \quad P_{2a}=0.1, P_{2m}=0.1, P_{2n}=0.1, P_{2r}=0.7$$

$$\text{Journal 3}: \quad P_{3a}=0.1, P_{3m}=0.3, P_{3n}=0.1, P_{3r}=0.5.$$

Of Prof. Smith's three papers, we want to calculate the probability that one paper will be accepted, one paper will be accepted with major revisions, and one paper will be rejected. In order to calculate this probability, we use $\gamma_1 = 1$, $\gamma_2 = 1$, $\gamma_3 = 0$, and $\gamma_4 = 1$ in (19) to obtain

$$\Pr(C_1=1, C_2=1, C_3=0, C_4=1) = \sum_{J\in Y(\gamma)} \prod_{k=1}^{3} \prod_{i=1}^{4} P_{ki}^{J_{ki}} \tag{20}$$

where

$$Y(\gamma) = \left\{ J \in \mathbf{R}^{3\times 4} : J_{ki} \in \{0,1\}, \sum_{i=1}^{4} J_{ki} = 1 \text{ for all } k, \right. \left. \sum_{k=1}^{3} J_{ki} = \gamma_i \text{ for all } i \right\}. \tag{21}$$

$J$ belongs to $Y$ if it satisifies all of the following conditions.

1) $J$ is a $3 \times 4$ matrix.
2) Each element of $J$ is either 0 or 1.
3) The elements in each row of $J$ add up to 1.
4) The elements in the $i$th column of $J$ add up to $\gamma_i$.

There are a total of $N!/(\gamma_1! \cdots \gamma_n!) = 3!/(1!\ 1!\ 0!\ 1!) = 6$ matrices $J^{(t)}$ that satisfy these conditions, and they are found as

$$J^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad J^{(2)} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$J^{(3)} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad J^{(4)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$J^{(5)} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad J^{(6)} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

Substituting these matrices into (20) gives

$$\begin{aligned}
&\Pr(C_1 = 1, C_2 = 1, C_3 = 0, C_4 = 1) \\
&= \sum_{t=1}^{6} \left( P_{11}^{J_{11}^{(t)}} P_{12}^{J_{12}^{(t)}} P_{13}^{J_{13}^{(t)}} P_{14}^{J_{14}^{(t)}} \right) \\
&\quad \times \left( P_{21}^{J_{21}^{(t)}} P_{22}^{J_{22}^{(t)}} P_{23}^{J_{23}^{(t)}} P_{24}^{J_{24}^{(t)}} \right) \left( P_{31}^{J_{31}^{(t)}} P_{32}^{J_{32}^{(t)}} P_{33}^{J_{33}^{(t)}} P_{34}^{J_{34}^{(t)}} \right) \\
&= P_{11}P_{22}P_{34} + P_{12}P_{21}P_{34} + P_{14}P_{21}P_{32} \\
&\quad + P_{11}P_{24}P_{32} + P_{12}P_{24}P_{31} + P_{14}P_{22}P_{31} \\
&= 0.066.
\end{aligned}$$

## APPENDIX B
## TRANSITION MATRIX DIMENSION

The elements of $Q$ are the probabilities of transitioning from one BBO population to another. $Q$ is a $T \times T$ matrix, where $T$ is the total number of possible population distributions. That is, $T$ is the number of possible $n \times 1$ integer vectors $v$ whose elements sum to $N$ and each of whose elements $v_i \in [0, N]$. This number can be calculated in several different ways. In [18], it is shown that

$$T = \binom{n + N - 1}{N}. \tag{22}$$

We can also use the multinomial theorem [19] to find $T$. The multinomial theorem can be stated in several ways, including the following. Given $K$ classes of objects, the number of different ways that $N$ objects can be selected (independent of order) while choosing from each class no more than $M$ times is the coefficient $q_N$ in the polynomial

$$\begin{aligned}
q(x) &= (1 + x + x^2 + \cdots + x^M)^K \\
&= 1 + q_1 x + q_2 x^2 + \cdots + q_N x^N + \cdots + x^{MK}.
\end{aligned} \tag{23}$$

Recall that the population vector $v$ is an $n$-element vector such that each element is an integer between 0 and $N$ (inclusive), and the sum of its elements is $N$. $T$ is the number of unique population vectors $v$. Thus, $T$ is the number of ways that $N$ objects can be selected (independent of order) from $n$ classes of objects while choosing from each class no more than $N$ times. Applying the multinomial theorem to this problem gives

$$\begin{aligned}
T &= q_N \\
q(x) &= (1 + x + x^2 + \cdots + x^N)^n \\
&= 1 + q_1 x + q_2 x^2 + \cdots + x^{Nn}.
\end{aligned} \tag{24}$$

A different form of the multinomial theorem can also be used to find $T$. The multinomial theorem can be stated as

$$\begin{aligned}
(x_1 + x_2 + \cdots + x_N)^n &= \sum_{S(k)} \frac{n!}{\prod_{j=0}^{N} k_j!} \prod_{j=0}^{N} x_j^{k_j} \\
&= \sum_{S(k)} \prod_{i=0}^{N} \binom{\sum_{j=0}^{i} k_j}{k_i} \prod_{j=0}^{N} x_j^{k_j} \\
S(k) &= \Bigg\{ k \in \mathbf{R}^N : k_j \in \{0, 1, \ldots, n\}, \\
&\qquad \sum_{j=0}^{N} k_j = n \Bigg\}.
\end{aligned} \tag{25}$$

Now, consider the polynomial $(x^0 + x^1 + x^2 + \cdots + x^N)^n$. From the multinomial theorem (25), we see that the coefficient of $[(x^0)^{k_0}(x^1)^{k_1}(x^2)^{k_2} \cdots (x^N)^{k_N}]$ is given by

$$\prod_{i=0}^{N} \binom{\sum_{j=0}^{i} k_j}{k_i}. \tag{26}$$

If we sum up these terms for all $k_j$ such that

$$\sum_{j=0}^{N} j k_j = N \tag{27}$$

then we obtain the coefficient of $x^N$. However, (24) shows that $T$ is equal to the coefficient of $x^N$. Therefore

$$\begin{aligned}
T &= \sum_{S'(k)} \prod_{i=0}^{N} \binom{\sum_{j=0}^{i} k_j}{k_i} \\
S'(k) &= \Bigg\{ k \in \mathbf{R}^{N+1} : k_j \in \{0, 1, \ldots, n\}, \\
&\qquad \sum_{j=0}^{N} k_j = n, \sum_{j=0}^{N} j k_j = n \Bigg\}.
\end{aligned} \tag{28}$$

Equations (22), (24), and (28) give three different expressions for the dimension of the Markov transition matrix $Q$.

*Example:* Suppose that our population consists of 2-b individuals ($q = 2, n = 4$) and a population size $N = 4$. Equation (22) gives

$$T = \binom{7}{4} = 35.$$

Equation (24) gives

$$\begin{aligned}
q(x) &= (1 + x + x^2 + x^3 + x^4)^4 \\
&= 1 + \cdots + 35x^4 + \cdots + x^{16} \\
T &= q_4 = 35.
\end{aligned}$$

Equation (28) gives

$$\begin{aligned}
T &= \sum_{S'(k)} \prod_{i=0}^{4} \binom{\sum_{j=0}^{i} k_j}{k_i} \\
S'(k) &= \Bigg\{ k \in \mathbf{R}^5 : k_j \in \{0, 1, \ldots, 4\}, \\
&\qquad \sum_{j=0}^{4} k_j = 4, \sum_{j=0}^{4} j k_j = 4 \Bigg\} \\
&= \{(3 \quad 0 \quad 0 \quad 0 \quad 1), (2 \quad 1 \quad 0 \quad 1 \quad 0), \\
&\quad (2 \quad 0 \quad 2 \quad 0 \quad 0), (1 \quad 2 \quad 1 \quad 0 \quad 0), \\
&\quad (0 \quad 4 \quad 0 \quad 0 \quad 0)\} \\
T &= 4 + 12 + 6 + 12 + 1 = 35.
\end{aligned}$$

## ACKNOWLEDGMENT

This paper is based upon work supported by the National Science Foundation under Grant 0826124.

## REFERENCES

[1] D. Simon, "Biogeography-based optimization," *IEEE Trans. Evol. Comput.*, vol. 12, no. 6, pp. 702–713, Dec. 2008.

[2] R. MacArthur and E. Wilson, "An equilibrium theory of insular zoogeography," *Evolution*, vol. 17, no. 4, pp. 373–387, Dec. 1963.

[3] R. MacArthur and E. Wilson, *The Theory of Biogeography*. Princeton, NJ: Princeton Univ. Press, 1967.

[4] M. Ergezer, D. Simon, and D. Du, "Oppositional biogeography-based optimization," in *Proc. IEEE Conf. Syst., Man, Cybern.*, San Antonio, TX, Oct. 2009, pp. 1009–1014.

[5] D. Du, D. Simon, and M. Ergezer, "Biogeography-based optimization combined with evolutionary strategy and immigration refusal," in *Proc. IEEE Conf. Syst., Man, Cybern.*, San Antonio, TX, Oct. 2009, pp. 997–1002.

[6] R. Rarick, D. Simon, F. Villaseca, and B. Vyakaranam, "Biogeography-based optimization and the solution of the power flow problem," in *Proc. IEEE Conf. Syst., Man, Cybern.*, San Antonio, TX, Oct. 2009, pp. 1003–1008.

[7] V. Savsani, R. Rao, and D. Vakharia, "Discrete optimisation of a gear train using biogeography based optimisation technique," *Int. J. Des. Eng.*, vol. 2, no. 2, pp. 205–223, 2009.

[8] D. Simon, "A probabilistic analysis of a simplified biogeography-based optimization algorithm," *Evol. Comput.*, 2010. [Online]. Available: http://embeddedlab.csuohio.edu/BBO

[9] J. Suzuki, "A Markov chain analysis on simple genetic algorithms," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 25, no. 4, pp. 655–659, Apr. 1995.

[10] J. Suzuki, "A further result on the Markov chain model of genetic algorithms and its application to a simulated annealing-like strategy," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 28, no. 1, pp. 95–102, Feb. 1998.

[11] M. Lundy and A. Mees, "Convergence of an annealing algorithm," *Math. Program.*, vol. 34, no. 1, pp. 111–124, Jan. 1986.

[12] N. Beaulieu, "On the generalized multinomial distribution, optimal multinomial detectors, and generalized weighted partial decision detectors," *IEEE Trans. Commun.*, vol. 39, no. 2, pp. 193–194, Feb. 1991.

[13] C. Grinstead and J. Snell, *Introduction to Probability*. Providence, RI: Amer. Math. Soc., 1997.

[14] D. Simon, M. Ergezer, and D. Du, "Population distributions in biogeography-based optimization algorithms with elitism," in *Proc. IEEE Conf. Syst., Man, Cyber.*, San Antonio, TX, Oct. 2009, pp. 1017–1022.

[15] C. Reeves and J. Rowe, *Genetic Algorithms: Principles and Perspectives*. Norwell, MA: Kluwer, 2003.

[16] C. Ahn, *Advances in Evolutionary Algorithms: Theory, Design and Practice*. New York: Springer-Verlag, 2006.

[17] H. Ma, S. Ni, and M. Sun, "Equilibrium species counts and migration model tradeoffs for biogeography-based optimization," in *Proc. IEEE Conf. Decis. Control*, Shanghai, China, Dec. 2009, pp. 3306–3310.

[18] A. Nix and M. Vose, "Modeling genetic algorithms with Markov chains," *Ann. Math. Artif. Intell.*, vol. 5, no. 1, pp. 79–88, Mar. 1992.

[19] C. Chuan-Chong and K. Khee-Meng, *Principles and Techniques in Combinatorics*. Hackensack, NJ: World Scientific, 1992.

**Dan Simon** (S'89–M'90–SM'01) received the B.S. degree in electrical engineering from Arizona State University, Tempe, in 1982, the M.S. degree in electrical engineering from the University of Washington, Seattle, in 1987, and the Ph.D. degree in electrical engineering from Syracuse University, Syracuse, NY, in 1991.

He worked in industry for 14 years at Boeing, TRW, and several small companies. His industrial experience includes work in the aerospace, automotive, agricultural, biomedical, process control, and software fields. In 1999, he moved from industry to academia, where he is currently a Professor with the Department of Electrical and Computer Engineering, Cleveland State University, Cleveland, OH. He has published over 70 refereed conference and journal papers and is the author of the book entitled "Optimal State Estimation" (Wiley, 2006). His teaching and research involve embedded systems, control systems, and computer intelligence.

**Mehmet Ergezer** (M'05) was born in Istanbul, Turkey, in 1981. He received the B.S. and M.S. degrees in electrical and computer engineering from Youngstown State University, Youngstown, OH, in 2003 and 2006, respectively. He is currently working toward the Ph.D. degree in evolutionary algorithms in the Department of Electrical and Computer Engineering, Cleveland State University, Cleveland, OH.

He was an Intern with U.S. Steel in 2002. From 2003 to 2005, he was a Graduate Assistant with Youngstown State University and an Intern with the Digital Engineering Team, Philips Medical Systems. He is currently a Research Assistant with the Embedded Control Systems Research Laboratory, Cleveland State University, where he works on heuristic numerical optimization techniques.

Mr. Ergezer is a member of Eta Kappa Nu.

**Dawei Du** was born in Jinan, China, on February 20, 1984. He received the B.S. degree from South-Central University for Nationalities, Wuhan, China, and the M.S. degree from Cleveland State University, Cleveland, OH, where he is currently working toward the Ph.D. degree in electrical engineering in the Department of Electrical and Computer Engineering.

He is with the Embedded Control Systems Research Laboratory, Cleveland State University. His current research interests include evolutionary computation, optimal control, and Kalman filtering.

**Rick Rarick** received the B.S. degree in electrical engineering, the M.S. degree in mathematics, and the M.S. degree in electrical engineering, specializing in control theory, from Cleveland State University, Cleveland, OH, in 1988, 1989, and 2007, respectively, where he is currently working toward the Ph.D. degree in the Department of Electrical and Computer Engineering, with his research being in the areas of control systems, power systems, and computer intelligence.

After continuing his studies in mathematics at Case Western Reserve University, Cleveland, he was an Assistant Professor with the Department of Mathematics, Notre Dame College, South Euclid, OH. In 2000, he accepted a position as an Engineering Consultant with Multi-Dimensional Imaging, Inc., where his duties included the development of image reconstruction software for a new computed tomography scanner.