

# Markov Models for Biogeography-Based Optimization and Genetic Algorithms with Global Uniform Recombination

Dan Simon,\* Mehmet Ergezer, and Dawei Du  
Cleveland State University  
Department of Electrical and Computer Engineering  
Stilwell Hall Room 332  
2121 Euclid Avenue  
Cleveland, Ohio 44115

June 14, 2009

## Abstract

Biogeography-based optimization (BBO) is a population-based evolutionary algorithm (EA) that is based on the mathematics of biogeography. Biogeography is the study of the geographical distribution of biological organisms. In BBO, problem solutions are represented as islands, and the sharing of features between solutions is represented as migration. BBO is similar to a genetic algorithm with global uniform recombination (GAGUR). This paper derives Markov models for BBO and GAGUR. Our models provide the limiting (stationary) probability of each population distribution for a given problem. We compare the Markov models for BBO, GAGUR, and GA with single-point crossover (GASP), which was derived by previous researchers. Comparisons on various types of simple problems (unimodal, multimodal, deceptive, and hierarchical) show that with high mutation rates the performance of BBO, GAGUR, and GASP is similar. With low mutation rates BBO outperforms GASP and GAGUR.

*Key Words* – biogeography-based optimization, genetic algorithms, uniform crossover, global recombination, Markov models

---

\*Corresponding author, email address [d.j.simon@csuohio.edu](mailto:d.j.simon@csuohio.edu). This work was supported by NSF Grant 0826124 in the CMMI Division of the Engineering Directorate.

## 1 Introduction

Mathematical models of biogeography describe the migration, speciation, and extinction of species. Biogeography-based optimization (BBO) was first presented in [1] and is an example of how a natural process can be modeled to solve general optimization problems. This is similar to what has occurred in the past few decades with GAs, neural networks, ant colony optimization, particle swarm optimization, and other types of computer intelligence. In this introduction we present a basic BBO algorithm.

BBO is based on the idea of probabilistically sharing features between solutions based on the solutions' fitness values. This is similar to many other evolutionary algorithms (EAs). However, the details are sufficiently different that BBO cannot be easily classified as a special case of another EA, as will be explained in Section 2.1.

In BBO, if solution feature  $s$  is shared by solution  $x$  with solution  $y$ , we say that  $s$  has *emigrated* from  $x$  and *immigrated* to  $y$ . Note that emigration of solution feature  $s$  from solution  $x$  to solution  $y$  does not mean that  $x$  loses that feature; it means that  $y$  gains that feature while losing some other feature. That is,  $s$  replaces a feature in  $y$ .

The probability that a given solution shares its features is proportional to its fitness, and the probability that a given solution receives features from the rest of the population is inversely proportional to its fitness. We base migration probabilities on a curve like that shown in Figure 1. For the sake of simplicity we assume that all solutions have identical migration curves. Figure 1 illustrates two solutions in BBO.  $S_1$  represents a poor solution and  $S_2$  represents a good 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, although mutation is not an essential feature of BBO. In this paper, mutation is implemented in a standard way for GA and BBO. We deal with discrete optimization problems in this paper, so each solution feature is either a 0 or a 1. The probability of mutation for the GA or 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

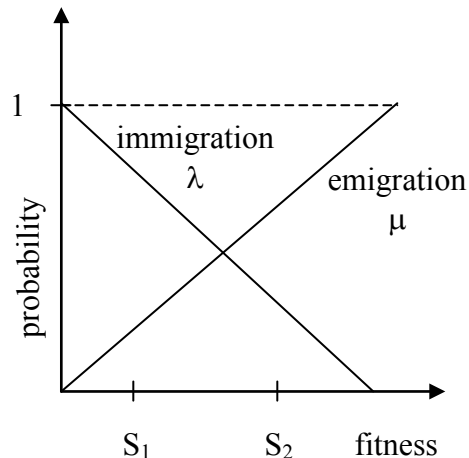


Figure 1: Illustration of two candidate solutions to some problem using symmetric immigration and emigration curves.  $S_1$  is a relatively poor solution and  $S_2$  is a relatively good solution.  $S_1$  has high immigration and low emigration, which means that it is likely to receive features from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

complement the bit under consideration.

Also similar to other population-based algorithms, we often incorporate some sort of elitism in BBO in order to retain the best solutions in the population. This prevents the best solutions from being corrupted by immigration. Elitism can be implemented by setting the immigration rate  $\lambda$  equal to zero for the  $p$  best solutions, where  $p$  is a user-selected elitism parameter. Elitism is not used in this paper but is modeled in [2].

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 [3]. In this approach, for each feature in each solution we probabilistically decide whether or not 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 Figure 2 as a conceptual description of one generation. 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 Figure 2.

Next, in Section 2, we show the similarities and differences between BBO and GA with global uniform recombination (GAGUR), and we justify the view of natural biogeography as an optimization process. In Section 3 we derive Markov models for BBO and GAGUR

---

```

 $z \leftarrow y$ 
Define immigration probability  $\lambda_k \propto$  fitness of the  $k$ -th solution
Define emigration probability  $\mu_k \propto 1/\lambda_k$ 
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
 $y \leftarrow z$ 

```

---

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

to obtain the limiting distribution of their populations. In Section 4 we confirm the models with simulations and compare the Markov models for BBO, GAGUR, and GA with single point crossover (GASP). We provide some concluding remarks and directions for future work in Section 5.

## 2 Biogeography-Based Optimization

In Section 2.1 we discuss the relationship between BBO and GAs. Although BBO could be viewed as a special type of GA, it is worth retaining it as a separate paradigm for at least two reasons. First, the differences in philosophy and implementation are sufficient that it does not easily fit into the GA framework. Second, as shown in Section 2.2, retaining BBO as a separate paradigm stimulates the exploitation of the extensive biogeography literature in order to explore and develop BBO from a biological perspective.

### 2.1 Biogeography-Based Optimization and Genetic Algorithms

The BBO migration strategy is conceptually similar to a combination of two ideas from GAs: global recombination, and uniform crossover. The first idea, global recombination, originated with evolutionary strategies (ES), and means that many parents can contribute to

a single offspring [4, 5]. This idea has also been applied in GAs with the names multi-parent recombination [6] and scanning crossover [7]. This strays from the biological foundation of GAs because individuals in nature cannot have more than two parents. There are several choices to be made when implementing global recombination in GAs. For example, how many parents should be in the pool of potential contributors? How should the parents be chosen for the pool? Once the pool has been selected, how should the individuals be selected from the pool to contribute to the new individual?

The second idea, uniform crossover, was first proposed in [8]. Uniform crossover is a generalization of single-point crossover, and means that each solution feature in an offspring is selected independently from a set of parents. If we combine global recombination and uniform crossover, we get global uniform recombination. If in addition we use the entire population as potential contributors to the next generation, and we use fitness-based selection for each solution feature in each offspring, we get the algorithm shown in Figure 3.

---

```

 $z \leftarrow y$ 
For each solution  $z_k$ 
  For each solution feature  $s$ 
    Use the fitness values to probabilistically select the parent feature  $y_j$ 
     $z_k(s) \leftarrow y_j(s)$ 
  end if
  Probabilistically decide whether to mutate  $z_k(s)$ 
next solution feature
next solution
 $y \leftarrow z$ 

```

---

Figure 3: One generation of a GA with global uniform recombination (GAGUR).  $y$  is the entire population of solutions,  $y_k$  is the  $k$ th solution, and  $y_k(s)$  is the  $s$ th feature of  $y_k$ . Compare with Figure 2.

Comparing Figures 2 and 3, it can be seen that BBO is a GA with global uniform recombination if  $\lambda_k = 1$  for all  $k$  in the BBO algorithm of Figure 2. The analogy between BBO and GAs can be summarized in Table 1. A solution is represented as a chromosome in a GA and as an island in BBO. A solution feature is represented as an allele in a GA and as a species in BBO. Recombination is represented as crossover in a GA and as migration in BBO.

Note that in BBO, islands are representative of problem solutions. This is much different

than island GAs [9], in which islands are representative of populations of solutions. A BBO algorithm consists of a single population with  $N$  islands/solutions, each having  $q$  solution features.

GA	$\longleftrightarrow$	BBO
chromosome	$\longleftrightarrow$	island
allele	$\longleftrightarrow$	species
crossover	$\longleftrightarrow$	migration

Table 1: Analogies between GA and BBO terminology.

It is not too surprising that BBO is similar to GAGUR. After all, many EAs can be expressed in terms of each other. For example, consider differential evolution (DE). Using the notation in [10], DE involves the selection of three random individuals from the population denoted as  $r_1$ ,  $r_2$ , and  $r_3$ , and the generation of a random parameter  $n$  between 1 and the population size. If, however,  $r_1$  is selected on the basis of fitness,  $r_2$  is replaced with  $r_1$ ,  $r_3$  is replaced with  $r_2$  and is selected on the basis of fitness, and  $n = 1$ , then DE is equivalent to a continuous GA with intermediate global recombination where the first parent is chosen deterministically and the second parent is chosen based on fitness. The converse can also be claimed: a GA can be expressed as a modified version of DE.

As another example, consider particle swarm optimization (PSO). Using the notation in [11], if a particle’s velocity at each generation is independent of its previous velocity,  $\phi_1 = 0$ , and  $p_g$  is probabilistically selected based on fitness, then PSO is equivalent to a continuous GA with intermediate fitness-based global recombination. The converse can also be claimed: a GA can be expressed as a modified version of PSO.

A final example is an evolution strategy (ES) [12]. A GA and an ES are “identical with respect to their major working scheme” [13]. If a  $(\mu, \lambda)$  ES is implemented with  $\lambda = \mu$ , parents selected based on fitness, uniform recombination, and a constant (nonadaptive) mutation parameter, then it is equivalent to a GA with fitness-based global uniform recombination. The converse can also be claimed: a GA can be expressed a special type of ES.

Since a GA can be viewed as either DE, PSO, or ES, it follows that all four of these EAs are equivalent to each other under certain special conditions. However, it is useful to retain the distinction between these EAs because of their differences and because of their

biological motivations. For example, retaining the biological foundation of GAs stimulates the incorporation of behaviors from biology in GAs, which makes the study of GAs richer and more flexible. Some of these behaviors include gender, niching, crowding, aging, diploidy, co-evolution, and ontogony [5]. Similarly, although BBO is equivalent to a special type of GA under certain circumstances, it is useful to retain it as a distinctive EA rather than simply viewing BBO as a GA variation. This idea is expanded in the following subsection.

## 2.2 Biogeography as an Optimizing Metaphor

In nature, a hospitable island has many species and thus is likely to share its species with other islands through migration. An inhospitable island has few species and thus is unlikely to share many species with other islands. Conversely, a hospitable island already supports many species and thus does not have room for many immigrating species. But an inhospitable island has few species and thus is likely to receive many immigrants. The issue of whether or not those immigrants can survive on the island is another question, but the immigrants may make the island more hospitable since diversity improves habitability [14].

Biogeography is nature's way of distributing species, which BBO views as an analogy to general problem solving. Suppose that we have some problem, and that we also have a certain number of candidate solutions. A good solution is analogous to an island that is hospitable to life, and a poor solution is analogous to an island that is not hospitable to life. In BBO each problem solution can share its features with other solutions. The probability that a given solution shares its features is proportional to its fitness, and the probability that a given solution receives features from other solutions is inversely proportional to its fitness.

Biogeography has often been studied as a process that enforces equilibrium in habitats. Equilibrium can be seen in Figure 1 where the immigration and emigration curves intersect. One reason that biogeography has been viewed from this perspective is that the equilibrium viewpoint was the first to place biogeography on a firm mathematical footing in the 1960s [15, 16]. However, since then the equilibrium perspective has been increasingly questioned by biogeographers.

In engineering, we often view stability and optimality as competing objectives; for example, a simple system is typically more stable than a complex system, while an optimal

system is typically complex and less stable than a simpler system. However, in biogeography, stability and optimality are two sides of the same coin. Biogeographical optimality involves diverse, complex communities which are highly adaptable to their environment. Biogeographical stability involves the persistence of existing populations. Field observations show that complex communities are more adaptable and stable than simple communities [17, p. 82], and this idea has also been supported by simulation [18, 19, 20].

Although the complementary nature of biogeographical optimality and stability has been challenged [21], adequate responses to these challenges have been put forth [22] and the idea is generally accepted today [20]. The equilibrium/optimality debate in biogeography thus becomes a matter of semantics, because equilibrium and optimality are simply two different perspectives on the same phenomenon in biogeography.

A dramatic example of the optimality of biogeography is Krakatoa, a volcanic island in the Indian Ocean which erupted in August 1883 [23]. The eruption was heard from thousands of miles away and resulted in the death of over 36,000 people, mostly from tidal waves whose remnants were recorded as far away as England. The eruption threw dust particles 30 miles high which remained aloft for months and were visible all around the world. Rogier Verbeek, a geologist and mining engineer, was the first visitor to Krakatoa, six weeks after the eruption, but the surface of the island was too hot to touch and showed no evidence of life. The island was completely sterilized [24]. The first animal life (a spider) was discovered on Krakatoa in May 1884, nine months after the eruption. By 1887, dense fields of grass were discovered on the island. By 1906, plant and animal life was abundant. Although volcanic activity continues today on Krakatoa, by 1983 (one century after its desolation) there were 88 species of trees and 53 species of shrubs [24], and the species count continues to increase linearly with time. Life immigrates to Krakatoa, and immigration makes the island more habitable, which in turn makes the island more friendly to additional immigration.

Biogeography is a positive feedback phenomenon, at least to a certain point. This is analogous to natural selection, also called survival of the fittest. As species become more fit, they are more likely to survive. As they continue to survive, they are better able to adapt to their environment and become more fit. Natural selection, like biogeography, is a positive feedback phenomenon, at least to a certain point. However, biogeography is an optimization

process that acts on a time scale that is much shorter than natural selection.

Another good example of biogeography as an optimization process is the Amazon rainforest, which is a typical case of a mutually optimizing life/environment system [17]. The rainforest has a large capacity to recycle moisture, which leads to less aridity and increased evaporation. This leads to cooler and wetter surfaces, which are more amenable to life. This suggests that a view of biogeography “based on *optimizing* environmental conditions for biotic activity seems more appropriate than a definition based on homeostasis” [25] (emphasis added). This view of the environment as a life-optimizing system was suggested as early as 1998 [26]. There are many other examples of the optimality of biogeographical processes, such as Earth’s temperature [17], Earth’s atmospheric composition [27], and the ocean’s mineral content [28].

This is not to say that biogeography is optimal for any particular species. For example, investigations of the Bikini Atoll shows that the high level of radioactivity resulting from nuclear tests had little effect on its natural ecology, but mammals were seriously affected [29, p. 37]. This and similar studies indicate that the Earth “will take care of itself [and] environmental excesses will be ameliorated, but its likely that such restoration of the environment will occur in a world devoid of people” [30]. Interestingly, amid all of the current warnings about ozone depletion, we forget that for the first two billion years of life Earth had no ozone at all [29, p. 109]. Life flourishes and evolves without ozone, but not in a human-centric way. Although global warming or an ice age would be disastrous for humans and many other mammals, it would be a minor event in the overall history of biogeography.

In summary, although BBO is equivalent to a special type of GA under specific conditions, it is useful to retain its distinction rather than simply viewing BBO as a GA variation. Unifying various EAs is instructive, but retaining BBO as a separate algorithm stimulates the incorporation of behaviors from biogeography in BBO, which opens up many areas of further research. Some of these behaviors include the effect of geographical proximity on migration rates, nonlinear migration curves to better match nature, species populations (including mortality and reproduction), predator/prey relationships, the effect of varying species mobilities on migration rates, directional momentum during migration, the effect of habitat area and isolation on migration rates, and many others.

### 3 Markov Models for BBO and GAGUR

Suppose that we have a problem with a binary search space. The search space consists of  $n$  possible bit strings  $x_i$ , each  $x_i$  containing  $q$  bits. 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. That is,  $v_i$  is the number of  $x_i$  individuals in the population. We see that

$$\sum_{i=1}^n v_i = N \quad (1)$$

The population of the search algorithm can be depicted as follows.

$$\text{Population} = \{y_1, \dots, y_N\} = \underbrace{\{x_1, x_1, \dots, x_1\}}_{v_1 \text{ copies}}, \underbrace{\{x_2, x_2, \dots, x_2\}}_{v_2 \text{ copies}}, \dots, \underbrace{\{x_n, x_n, \dots, x_n\}}_{v_n \text{ copies}} \quad (2)$$

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  $x_i$ . In BBO,  $\lambda_i$  is inversely proportional to the fitness of  $x_i$ , and in GAGUR,  $\lambda_i = 1$ . 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,

$$\mathcal{J}_i(s) = \{j : x_j(s) = x_i(s)\} \quad (3)$$

We use  $y_k$  to denote the  $k$ th individual in the population, and we order them in the same order as  $x_i$ . That is,

$$y_k = \begin{cases} x_1 & \text{for } k = 1, \dots, v_1 \\ x_2 & \text{for } k = v_1 + 1, \dots, v_1 + v_2 \\ x_3 & \text{for } k = v_1 + v_2 + 1, \dots, v_1 + v_2 + v_3 \\ \vdots & \vdots \\ x_n & \text{for } k = \sum_{i=1}^{n-1} v_i + 1, \dots, N \end{cases} \quad (4)$$

This is also illustrated in (2), and can be written more compactly as

$$\begin{aligned} y_k &= x_m \text{ for } k = 1, \dots, N \\ m &= \min r \text{ such that } \sum_{i=1}^r v_i \geq k \end{aligned} \quad (5)$$

If we need to denote the generation number of the BBO or GAGUR 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  $\#t$ .

### 3.1 Migration

We make some assumptions in the Markov model development in this section. First, all of the new solutions are created before any solutions are replaced in the population. This is clear from the use of the temporary population  $z$  in Figures 2 and 3. Second, a solution can emigrate a bit to itself. This is analogous to crossover in GAs when a solution is allowed to cross with itself. Third, the migration rates  $\lambda$  and  $\mu$  are independent of the population distribution. That is, absolute fitness values are used to obtain  $\lambda$  and  $\mu$ , as opposed to some sort of rank-based fitness.

For each solution feature (bit),  $y_k$  has  $v_m$  chances of being selected for immigration, each chance with probability  $\lambda_m$ . If the  $s$ th feature of  $y_k$  is not selected for immigration during generation  $t$ , then

$$y_k(s)_{t+1} = x_m(s) \text{ (immigration did not occur)} \quad (6)$$

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

$$\Pr(y_k(s)_{t+1} = x_i(s)) = \frac{\sum_{j \in \mathcal{J}_i(s)} v_j \mu_j}{\sum_j v_j \mu_j} \text{ (immigration occurred)} \quad (7)$$

We can combine (6) and (7), along with the fact that the probability of immigration to  $y_k(s)$  is equal to  $\lambda_m$ , to obtain the total probability

$$\Pr(y_k(s)_{t+1} = x_i(s)) = (1 - \lambda_m) \mathbf{1}_0(x_m(s) - x_i(s)) + \lambda_m \frac{\sum_{j \in \mathcal{J}_i(s)} v_j \mu_j}{\sum_j v_j \mu_j} \quad (8)$$

where  $\mathbf{1}_0$  is the indicator function on the set  $\{0\}$ . Given the fact that there are  $q$  bits in each solution, the probability that immigration results in  $y_k$  being equal to  $x_i$ , given that the population is described by the vector  $v$ , is denoted as  $P_{ki}(v)$  and can be written as

$$\begin{aligned} P_{ki}(v) &= \Pr(y_{k,t+1} = x_i) \\ &= \prod_{s=1}^q \left[ (1 - \lambda_m) \mathbf{1}_0(x_m(s) - x_i(s)) + \lambda_m \frac{\sum_{j \in \mathcal{J}_i(s)} v_j \mu_j}{\sum_j v_j \mu_j} \right] \end{aligned} \quad (9)$$

$P_{ki}(v)$  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

Figures 2 and 3. The  $i$ th column of  $P(v)$  corresponds to the probability of obtaining island  $x_i$  during each outer loop iteration.

The BBO and GAGUR algorithms entail  $N$  trials (i.e.,  $N$  iterations of the outer loop in Figures 2 and 3), 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 all  $N$  trials have been completed, and  $u = [u_1 \ \cdots \ u_n]^T$ . Then the probability that we start with a population vector  $v$  and obtain a population vector  $u$  after one generation is given by the generalized multinomial theorem [31] as

$$\Pr(u|v) = \sum_Y \prod_{k=1}^N \prod_{i=1}^n P_{ki}^{J_{ki}}(v)$$

$$Y = \left\{ J \in \mathbf{R}^{N \times n} : J_{ki} \in \{0, 1\}, \sum_{i=1}^n J_{ki} = 1 \text{ for all } k, \sum_{k=1}^N J_{ki} = u_i \text{ for all } i \right\} \quad (10)$$

In order to find the probability that the BBO or GAGUR population transitions from  $v$  to  $u$  after one generation, we find all of the  $J$  matrices that satisfy the conditions of (10). For each of these  $J$  matrices, we compute the product of products given in (10). We then add up all the product-of-products to obtain the desired probability.

### 3.2 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.  $U_{ij}$  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

$$P_{ki}^{(2)}(v) = \sum_{j=1}^n U_{ij} P_{kj}(v)$$

$$P^{(2)}(v) = P(v)U^T \quad (11)$$

where the elements of  $P(v)$  are given in (9).  $P(v)$  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 population vector  $u$  after one generation as

$$\Pr^{(2)}(u|v) = \sum_Y \prod_{k=1}^N \prod_{i=1}^n [P_{ki}^{(2)}(v)]^{J_{ki}} \quad (12)$$

where  $Y$  is given in (10). Equation 12 gives the transition matrix elements for BBO with migration and mutation. We can use standard Markov tools [32] with the transition matrix to find the limiting distribution of the BBO or GAGUR population.

The Markov transition matrix  $Q$  is obtained by computing (12) for each possible  $v$  vector and each possible  $u$  vector.  $Q$  is therefore 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 whose elements sum to  $N$  and each of whose elements is in  $[0, N]$ . This number can be calculated several different ways. In [33] it is shown that  $T$  can be calculated with the choose function

$$T = \binom{n + N - 1}{N} \quad (13)$$

Other methods for calculating  $T$  can be found in [34].

## 4 Results

This section first confirms the BBO Markov model with simulation. We then use Markov models in Section 4.2 and simulations in Section 4.3 to explore the relative performance of GASP, GAGUR, and BBO.

### 4.1 BBO Simulations

In this section we compare the BBO Markov model derived in the previous section with simulations. We use the three-bit 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. From (13) we see that the total number of possible populations is equal to 330. Equation (12) can be used to find the limiting population distribution of BBO. Table 2 shows the most probable populations, along with the combined probabilities of the populations that do not contain any optimal solutions. The Markov model and simulation results match well, which confirms the model. Table 2 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%. With this high mutation rate the probability that the population does not have any optimal individuals is 30%. However, as the mutation rate decreases to the more reasonable values of 1% and 0.1%, the probability that the population is comprised entirely of optimal individuals increases to 53% and 86% respectively, and the probability that the population has no optimal individuals decreases to 11% and 9% respectively.

Mutation Rate	Population Vector	Probability	
		Markov	Simulation
0.1	0 0 0 0 0 0 1 3	0.0290	0.0285 ± 0.0287
	0 0 0 0 0 1 0 3	0.0290	0.0284 ± 0.0286
	0 0 0 1 0 0 0 3	0.0290	0.0284 ± 0.0286
	* * * * * 0	0.2999	0.3026 ± 0.3042
0.01	0 0 0 0 0 0 0 4	0.5344	0.5322 ± 0.5350
	0 0 0 0 0 0 1 3	0.0718	0.0715 ± 0.0719
	0 0 0 0 0 1 0 3	0.0718	0.0716 ± 0.0721
	0 0 0 1 0 0 0 3	0.0718	0.0726 ± 0.0730
	* * * * * 0	0.1134	0.1138 ± 0.1158
0.001	0 0 0 0 0 0 0 4	0.8605	0.8437 ± 0.8499
	0 0 0 0 0 0 4 0	0.0288	0.0386 ± 0.0493
	0 0 0 0 0 4 0 0	0.0288	0.0408 ± 0.0507
	0 0 0 4 0 0 0 0	0.0288	0.0380 ± 0.0494
	* * * * * 0	0.0923	0.1092 ± 0.1246

Table 2: BBO Markov model and simulation results (mean ± standard deviation) of the most probable populations, and the combined probability of convergence to populations that contain no optimal solutions. Simulation results are based on 100 Monte Carlo runs.

Figure 4 shows typical simulation results of 20,000 generations of BBO for the three-bit 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 2.

## 4.2 Markov Model Comparisons

Consider a GA with fitness-proportional (roulette wheel) selection, followed by mutation, followed by single point crossover (GASP). We use  $v_i$  to represent the  $i$ th element of the population vector  $v$ , and  $f_i$  is the fitness of  $x_i$ .  $G_j^s(v)$  is the probability of obtaining individual

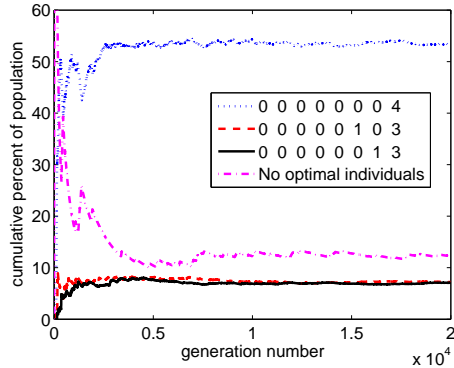


Figure 4: Typical BBO simulation results for a three-bit unimodal optimization problem (one-max) with a mutation rate of 1% per bit. The three most probable populations are shown, along with the cumulative probability of all populations that have no optimal individuals.

$x_j$  by selection alone from a population represented by vector  $v$ .  $U_{ij}$  is the probability of obtaining  $x_i$  from  $x_j$  by mutation,  $G_i^{sm}(v)$  is the probability of obtaining individual  $x_i$  by selection and mutation combined,  $r(i, j, k)$  is the probability that  $x_i$  and  $x_j$  cross to form  $x_k$ , and  $G_k^{smc}(v)$  is the probability of obtaining individual  $x_k$  by selection, mutation, and crossover. These probabilities are derived in [32, 33, 35, 36] as

$$\begin{aligned}
 G_j^s(v) &= \frac{v_j f_j}{\sum_j v_j f_j} \\
 G_i^{sm}(v) &= \sum_j U_{ij} G_j^s(v) \\
 G_k^{smc}(v) &= \sum_j r(i, j, k) G_i^{sm}(v) G_j^{sm}(v)
 \end{aligned} \tag{14}$$

Equation (14) can be used with the multinomial theorem [37] to obtain the probability that population vector  $v$  transitions to  $u$  after one generation.

$$\Pr_G(u|v) = N! \prod_i \frac{(G_i^{smc}(v))^{u_i}}{u_i!} \tag{15}$$

In this section we use (12) to obtain the limiting population distributions of BBO ( $\lambda_k \propto f_k$ ) and GAGUR ( $\lambda_k = 1$ ), and (15) to obtain the limiting population distributions of GASP. Due to the exponential increase of matrix sizes with problem size, investigation was limited to four-bit problems ( $n = 16$ ) with a population size of four ( $N = 4$ ), which results in 3,876 possible population vectors as shown in (13).

The problems we investigated are denoted as  $F_u$ ,  $F_m$ ,  $F_d$ , and  $F_h$ .  $F_u$  is the unimodal one-max problem in which the fitness of each bit string is equal to the number of ones in the

bit string.  $F_m$  is a multimodal problem; its fitness values are equal to those of the one-max problem, except that the bit string consisting of all zeros has the same fitness as the bit string consisting of all ones.  $F_d$  is a deceptive problem; its fitness values are equal to those of the one-max problem, except it is a unimodal problem in which the bit string consisting of all zeros has the highest fitness.  $F_h$  is a hierarchical-if-and-only-if (H-IFF) problem; its fitness values are defined in such a way that it is amenable to the use of building blocks for optimum seeking [38]. H-IFF problems are constructed to be compatible with the capabilities of GASP.

Tables 3-6 show comparisons between Markov model results for GASP, GAGUR, and BBO. The tables show the probability of obtaining a uniform optimal population and the probability of obtaining a population which does not have any optimal individuals. The crossover probability used in GASP was 0.9.

Mutation Rate	Population Vector	Probability		
		GASP	GAGUR	BBO
0.1	Uniform Optimal	<b>0.0084</b>	0.0079	0.0044
	No Optima	0.5826	0.5623	<b>0.5111</b>
0.01	Uniform Optimal	0.2492	0.2513	<b>0.3484</b>
	No Optima	0.5436	0.5372	<b>0.2128</b>
0.001	Uniform Optimal	0.4029	0.4034	<b>0.7616</b>
	No Optima	0.5696	0.5690	<b>0.1679</b>

Table 3: Unimodal problem optimization results for  $F_u$ . The results were obtained using Markov models. The best performance is in **bold font** in each row.

Mutation Rate	Population Vector	Probability		
		GASP	GAGUR	BBO
0.1	Uniform Optimal	<b>0.0119</b>	0.0106	0.0066
	No Optima	0.5006	0.4939	<b>0.4370</b>
0.01	Uniform Optimal	0.3675	0.3701	<b>0.4715</b>
	No Optima	0.4139	0.4079	<b>0.1450</b>
0.001	Uniform Optimal	0.5655	0.5670	<b>0.8502</b>
	No Optima	0.4069	0.4053	<b>0.0968</b>

Table 4: Multimodal problem optimization results for  $F_m$ . The results were obtained using Markov models. The best performance is in **bold font** in each row.

Mutation Rate	Population Vector	Probability		
		GASP	GAGUR	BBO
0.1	Uniform Optimal	<b>0.01314</b>	0.0109	0.0120
	No Optima	0.8120	0.8325	<b>0.7954</b>
0.01	Uniform Optimal	0.4601	0.4760	<b>0.6506</b>
	No Optima	0.4308	0.4103	<b>0.1915</b>
0.001	Uniform Optimal	0.6230	0.6383	<b>0.9074</b>
	No Optima	0.3638	0.3482	<b>0.0730</b>

Table 5: Deceptive problem optimization results for  $F_d$ . The results were obtained using Markov models. The best performance is in **bold font** in each row.

Mutation Rate	Population Vector	Probability		
		GASP	GAGUR	BBO
0.1	Uniform Optimal	<b>0.0143</b>	0.0126	0.0091
	No Optima	0.4820	0.4887	<b>0.4233</b>
0.01	Uniform Optimal	0.4305	0.4317	<b>0.5546</b>
	No Optima	0.3494	0.3463	<b>0.1155</b>
0.001	Uniform Optimal	0.6535	0.6539	<b>0.8976</b>
	No Optima	0.3190	0.3185	<b>0.0604</b>

Table 6: H-IFF problem optimization results for  $F_h$ . The results were obtained using Markov models. The best performance is in **bold font** in each row.

Several things are notable about the results of Tables 3–6.

1. In each table as the mutation rate decreases, performance improves. The probability of a uniform optimal population increases, and the probability of no optima decreases. This is true for all three algorithms.
2. GASP is the best algorithm only when the mutation rate is high (10% per bit) and only insofar that the probability of a uniform optimal population is slightly higher in GASP than in GAGUR and BBO.
3. In every other performance comparison in the tables, GAGUR slightly outperforms GASP, and BBO far outperforms both GASP and GAGUR. This is especially true when the mutation rate is low (0.1% per bit). In this case, BBO outperforms GASP and GAGUR in its higher probability of a uniform optimal population by an average of 85% to 56%, and BBO outperforms GASP and GAGUR in its lower probability of a population with no optima by an average of 10% to 41%.

Similar comparisons between single-parent and multi-parent EAs have been noted in

previous papers. In particular, ES performance has been seen to generally improve as the number of parents increases [39]. However, we have shown that the difference between GASP and GAGUR is relatively small for the specific problems in this section, but the improved performance that comes with using BBO is significant. This is because a BBO individual uses its own fitness before deciding how likely it is to accept features from other solutions. This simple and intuitive idea does not have an analogy in natural selection, but is motivated by biogeography. The results in this section are also consistent with [40], which showed that BBO with a constant immigration rate ( $\lambda_k = 1$ ), which we have shown in this paper to be equivalent to GAGUR, gives much worse performance than standard BBO ( $\lambda_k \propto f_k$ ).

### 4.3 Time to First Optimum

In this section we investigate the performance of BBO, GASP, and GAGUR experimentally for the four problems discussed in the previous section. We ran 100 Monte Carlo simulations of 20,000 generations each and recorded the time at which the population found its first optimum. Time to first optimum is not given by the Markov models, so we need to use simulations to obtain this information. Tables 7–10 give the percent of runs in which at least one optimum was found, and the average generation number when an optimum was first found.

Mutation		GASP	GAGUR	BBO
0.1	Percent	100	100	100
	Generation	$5.8 \pm 6.4$	<b><math>4.9 \pm 5.2</math></b>	$6.0 \pm 5.5$
0.01	Percent	100	100	100
	Generation	$32.2 \pm 46.8$	$25.0 \pm 35.8$	<b><math>24.0 \pm 24.0</math></b>
0.001	Percent	100	100	100
	Generation	$248.7 \pm 336.6$	$212.8 \pm 334.7$	<b><math>147.1 \pm 238.2</math></b>

Table 7: Unimodal problem optimization results for  $F_u$  showing the percent of simulations that achieved at least one optimum, and the generation number (mean  $\pm$  standard deviation) when the optimum was first achieved. The results were obtained with 100 Monte Carlo simulations of 20,000 generations each. The best performance is in **bold font** in each row.

Mutation		GASP	GAGUR	BBO
0.1	Percent	100	100	100
	Generation	3.2 ± 3.4	<b>2.8 ± 2.6</b>	<b>2.8 ± 2.6</b>
0.01	Percent	100	100	100
	Generation	27.5 ± 44.1	<b>16.8 ± 29.7</b>	17.0 ± 33.4
0.001	Percent	100	100	100
	Generation	152.6 ± 288.0	144.1 ± 326.6	<b>126.0 ± 257.7</b>

Table 8: Multimodal problem optimization results for  $F_m$  showing the percent of simulations that achieved at least one optimum, and the generation number (mean ± standard deviation) when the optimum was first achieved. The results were obtained with 100 Monte Carlo simulations of 20,000 generations each. The best performance is in **bold font** in each row.

Mutation		GASP	GAGUR	BBO
0.1	Percent	100	100	100
	Generation	36.4 ± 41.9	34.1 ± 42.9	28.4 ± 43.2
0.01	Percent	<b>98</b>	96	85
	Generation	4030 ± 4741	<b>3639 ± 4461</b>	3870 ± 4734
0.001	Percent	26	24	<b>29</b>
	Generation	<b>660 ± 2091</b>	692 ± 2642	829 ± 3100

Table 9: Deceptive problem optimization results for  $F_d$  showing the percent of simulations that achieved at least one optimum, and the generation number (mean ± standard deviation) when the optimum was first achieved. The results were obtained with 100 Monte Carlo simulations of 20,000 generations each. The best performance is in **bold font** in each row.

Mutation		GASP	GAGUR	BBO
0.1	Percent	100	100	100
	Generation	3.6 ± 4.1	<b>3.0 ± 3.0</b>	3.3 ± 4.4
0.01	Percent	100	100	100
	Generation	28.4 ± 50.1	<b>19.1 ± 39.5</b>	20.4 ± 53.2
0.001	Percent	100	100	100
	Generation	402 ± 753	377 ± 802	<b>281 ± 848</b>

Table 10: H-IFF problem optimization results for  $F_h$  showing the percent of simulations that achieved at least one optimum, and the generation number (mean ± standard deviation) when the optimum was first achieved. The results were obtained with 100 Monte Carlo simulations of 20,000 generations each. The best performance is in **bold font** in each row.

Several things are notable about the results of Tables 7–10.

1. In each table, increasing the mutation rate decreases the time to first optimum, as expected for small problems.
2. The large standard deviations in each table show that there is a large variation in performance between simulations. This shows that EA results in general can vary significantly from run to run.
3. In the unimodal problem results shown in Table 7, BBO is generally the best EA.
4. In the multimodal problem results shown in Table 8, GAGUR is generally the best EA.
5. The deceptive problem results in Table 9 show that this is the only problem in which the EAs achieved the optimum less than 100% of the time. For a moderate mutation rate of 1%, GASP found the optimum most often. For a low mutation rate of 0.1%, BBO found the optimum most often, which is the most important metric, but of those times when the optimum was found, GASP found it more quickly.
6. In the H-IFF problem results shown in Table 10, GAGUR performs the best for moderate mutation rates, while BBO performs the best for low mutation rates.
7. When the mutation rate is low, BBO clearly performs the best in all four problems.

The conclusions drawn from the simulation results in this section are somewhat different than those drawn from the Markov models in Section 4.2. This is because the Markov model results in Section 4.2 are based on limiting population distributions as time approaches infinity. The Markov model is therefore only an approximation to finite-time simulations. The advantage of the Markov model is that it is exact, while simulation results vary from run to run because of random effects.

## 5 Conclusion

We have derived Markov models for genetic algorithms with global uniform recombination (GAGUR) and biogeography-based optimization (BBO). The models show the theoretical probability as the generation count goes to infinity of each possible population for a given problem. Some small representative problems were investigated. The results showed that

BBO generally outperforms both GAs with single-point crossover (GASP) and GAGUR, especially if the mutation rate is small.

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 [32], 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, and total emigration-based BBO [3], among others. 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.

We have shown that the improved performance that comes with using BBO compared to GASP and GAGUR can be significant. This is apparently because a BBO individual uses its own fitness before deciding how likely it is to accept features from other solutions. This simple and intuitive idea does not have an analogy in natural selection, but is motivated by biogeography. An interesting direction for future work is to use this idea in other EAs. For example, before using neighbors to modify a given particle swarm optimization solution, the algorithm could first take the given particle's performance into account. The same idea could be used with ant colony optimization, differential evolution, and other EAs.

Finally, the Markov model developed here forms a foundation that can be used to develop a dynamic systems analysis of BBO. Dynamic systems 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 systems analysis [32].

## **Acknowledgments**

The authors are grateful to Rick Rarick for reviewing a preliminary version of this paper and helping to clarify it. Also, the comments of the Associate Editor and the anonymous reviewers were instrumental in significantly strengthening this paper after its original submission.

## References

- [1] D. Simon, Biogeography-based optimization, *IEEE Transactions on Evolutionary Computation* (12) pp. 702-713, December 2008.
- [2] D. Simon, M. Ergezer, and D. Du, Population distributions in biogeography-based optimization algorithms with elitism, accepted for publication, *IEEE Conference on Systems, Man, and Cybernetics*, October 2009, <http://embeddedlab.csuohio.edu/BBO>.
- [3] D. Simon, A probabilistic analysis of a simplified biogeography-based optimization algorithm, submitted for publication, <http://academic.csuohio.edu/simond/bbo/simplified>, February 2009.
- [4] T. Back, *Evolutionary Algorithms in Theory and Practice* (Oxford University Press, 1996).
- [5] T. Back, U. Hammel, and H. Schwefel, Evolutionary computation: Comments on the history and current state, *IEEE Transactions on Evolutionary Computation* (1) pp. 3-17, January 1997.
- [6] A. Eiben, Multiparent recombination in evolutionary computing, in: A. Ghosh and S. Tsutsui (eds.), *Advances in Evolutionary Computing* (Springer-Verlag, 2003), pp. 175-192.
- [7] A. Eiben and C Schippers, Multi-parent's niche: n-ary crossovers on NK-landscapes, *Proceedings of the 4th Conference on Parallel Problem Solving from Nature*, pp. 319-328, 1996.
- [8] D. Ackley, *A Connectionist Machine for Genetic Hillclimbing*, Kluwer Academic Publishers, 1987.
- [9] Z. Skolicki, An analysis of island models in evolutionary computation, *Genetic And Evolutionary Computation Conference*, Washington, DC, pp. 386-389, 2005.
- [10] K. Price and R. Storn, Differential evolution, *Dr. Dobb's Journal* (22) pp. 18-20, 22, 24, 78, April 1997.

- [11] M. Clerc and J. Kennedy, The particle swarm – Explosion, stability, and convergence in a multidimensional complex space, *IEEE Transactions on Evolutionary Computation* (6) pp. 58-73, February 2002.
- [12] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, 1998.
- [13] F. Hoffmeister and T. Back, Genetic algorithms and evolution strategies: Similarities and differences, *Proceedings of the 1st Workshop on Parallel Problem Solving from Nature*, pp. 455-469, 1991.
- [14] T. Wesche, G. Goertler, and W. Hubert, Modified habitat suitability index model for brown trout in southeastern Wyoming, *North American Journal of Fisheries Management* (7) pp. 232-237, 1987.
- [15] R. MacArthur and E. Wilson, An equilibrium theory of insular zoogeography, *Evolution* (17) pp. 373-387, 1963.
- [16] R. MacArthur and E. Wilson, *The Theory of Biogeography* (Princeton University Press, 1967).
- [17] S. Harding, *Animate Earth*, Chelsea Green Publishing Company, 2006.
- [18] C. Elton, *Ecology of Invasions by Animals and Plants*, Chapman & Hall, 1958.
- [19] R. MacArthur, Fluctuations of animal populations and a measure of community stability, *Ecology* (36) pp. 533-536, 1955.
- [20] K. McCann, The diversity-stability debate, *Nature* (405) pp. 228-233, 2000.
- [21] R. May, *Stability and Complexity in Model Ecosystems*, Princeton University Press, 1973.
- [22] M. Kondoh, Does foraging adaptation create the positive complexity-stability relationship in realistic food-web structure? *Journal of Theoretical Biology* (238) pp. 646-651, 2006.
- [23] S. Winchester, *The Day the World Exploded*, Collins, 2008.

- [24] R. Whittaker and M. Bush, Dispersal and establishment of tropical forest assemblages, Krakatoa, Indonesia, in: J. Miles and D. Walton (eds.) *Primary Succession on Land*, Blackwell Science, pp. 147-160, 1993.
- [25] A. Kleidon, Amazonian biogeography as a test for Gaia, in: S. Schneider et. al, *Scientists Debate Gaia*, pp. 291-296, MIT Press, 2004.
- [26] T. Volk, *Gaia's Body: Toward a Physiology of Earth*, MIT Press, 2003.
- [27] T. Lenton, Gaia and natural selection, *Nature* (394) pp. 439-447, 1998.
- [28] J. Lovelock, Hands up for the Gaia hypothesis, *Nature* (344) pp. 100-102, 1990.
- [29] J. Lovelock, *Gaia*, Oxford University Press, 1995.
- [30] L. Margulis, Gaia is a tough bitch, in: J. Brockman, *The Third Culture*, Chapter 7, Touchstone, 1991.
- [31] N. Beaulieu, On the generalized multinomial distribution, optimal multinomial detectors, and generalized weighted partial decision detectors, *IEEE Transactions on Communications* (39) pp. 193-194, February 1991.
- [32] C. Reeves and J. Rowe, *Genetic Algorithms: Principles and Perspectives* (Kluwer, 2003).
- [33] A. Nix and M. Vose, Modeling genetic algorithms with Markov chains, *Annals of Mathematics and Artificial Intelligence* (5) pp. 79-88, 1992.
- [34] D. Simon, The dimension of an evolutionary algorithm transition matrix, <http://academic.csuohio.edu/simond/bbo/markov>, May 2009.
- [35] T. Davis and J. Principe, A simulated annealing like convergence theory for the simple genetic algorithms, in *Proceedings of the Fourth International Conference on Genetic Algorithms* (R. Belew and L. Booker, eds.), Morgan Kaufman, 1991, pp. 174-181.
- [36] T. Davis and J. Principe, A Markov chain framework for the simple genetic algorithm, *Evolutionary Computation* (1) pp. 269-288, 1993.
- [37] C. Chuan-Chong and K. Khee-Meng, *Principles and Techniques in Combinatorics*, World Scientific, 1992.
- [38] R. Watson and J. Pollack, Hierarchically consistent test problems for genetic algorithms, *Congress on Evolutionary Computation*, pp. 292-297, 1999.

- [39] A. Eiben and T. Back, Empirical investigation of multiparent recombination operators in evolution strategies, *Evolutionary Computation* (5) pp. 347-365, 1998.
- [40] H. Ma and X. Chen, Equilibrium species counts and migration model tradeoffs for biogeography-based optimization, submitted for publication, <http://embeddedlab.csuohio.edu/BBO>, 2009.