

Summer 2011

A Probabilistic Analysis of a Simplified Biogeography-Based Optimization Algorithm

Daniel J. Simon

Cleveland State University, d.j.simon@csuohio.edu

Follow this and additional works at: https://engagedscholarship.csuohio.edu/enece_facpub

 Part of the [Electrical and Computer Engineering Commons](#)

How does access to this work benefit you? Let us know!

Original Citation

Simon, D. (2011). A probabilistic analysis of a simplified biogeography-based optimization algorithm. *Evolutionary Computation*, 19(2), 167-188.

Repository Citation

Simon, Daniel J., "A Probabilistic Analysis of a Simplified Biogeography-Based Optimization Algorithm" (2011). *Electrical Engineering & Computer Science Faculty Publications*. 4.
https://engagedscholarship.csuohio.edu/enece_facpub/4

This Article is brought to you for free and open access by the Electrical Engineering & Computer Science Department at EngagedScholarship@CSU. It has been accepted for inclusion in Electrical Engineering & Computer Science Faculty Publications by an authorized administrator of EngagedScholarship@CSU. For more information, please contact library.es@csuohio.edu.

A Probabilistic Analysis of a Simplified Biogeography-Based Optimization Algorithm

Dan Simon

d.j.simon@csuohio.edu

Department of Electrical and Computer Engineering, Cleveland State University,
Cleveland, Ohio, 44115

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. We present a simplified version of BBO and perform an approximate analysis of the BBO population using probability theory. Our analysis provides approximate values for the expected number of generations before the population's best solution improves, and the expected amount of improvement. These expected values are functions of the population size. We quantify three behaviors as the population size increases: first, we see that the best solution in the initial randomly generated population improves; second, we see that the expected number of generations before improvement increases; and third, we see that the expected amount of improvement decreases.

Keywords

Biogeography-based optimization, evolutionary algorithms, Markov analysis.

1 Introduction

1.1 Biogeography-Based Optimization

Mathematical models of biogeography describe the migration, speciation, and extinction of species (MacArthur and Wilson, 1967; Lomolino et al., 2009). Species migrate between islands. Islands that are well suited as residences for biological species are said to be highly habitable. Features that correlate with habitability include factors such as rainfall, diversity of vegetation, diversity of topographic features, land area, and temperature. Islands that are highly habitable tend to have many species, while those that are not very habitable have few species.

Highly habitable islands have a high emigration rate. Emigration occurs as animals ride flotsam to neighboring islands, or swim to neighboring islands. In the case of insects and birds, emigration can occur by flying or being carried by the wind. The reason that emigration occurs from habitable islands is due to the accumulation of random effects on their large populations.

Highly habitable islands have a low immigration rate because they are already nearly saturated with species, and therefore cannot easily support new species. Conversely, islands that are not habitable have a high immigration rate because of their sparse populations, which allows room for many additional species. The immigration of new species to islands might raise the habitability of those islands because habitability is proportional to biological diversity.

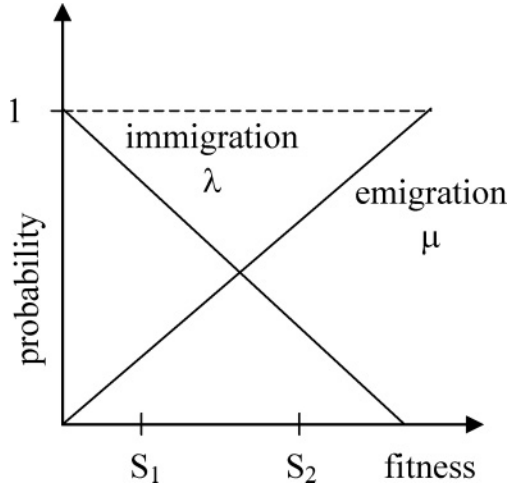


Figure 1: Illustration of two candidate problem solutions using symmetric immigration and emigration curves. S_1 is a relatively poor solution while S_2 is a relatively good solution. S_1 has a high immigration and a low emigration rate, and S_2 has a low immigration and a high emigration rate.

The application of biogeography to optimization was first presented in Simon (2008) and is an example of how a natural process can be modeled to solve general optimization problems. Biogeography is nature's way of distributing species, and is analogous 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 a highly habitable island, and a poor solution is analogous to a less habitable island. Good solutions are more likely to share their features with other solutions, and poor solutions are more likely to accept shared features from other solutions. This approach to problem solving is called biogeography-based optimization (BBO). As with every other evolutionary algorithm (EA), we might also incorporate mutation and elitism, although these are not essential features of BBO.

Figure 1 illustrates migration models. The immigration rate λ and the emigration rate μ of a solution are functions of its fitness. The immigration curve shows that the least fit solution has the largest immigration rate and smallest emigration rate. The most fit solution has the smallest immigration rate and the largest emigration rate. S_1 in Figure 1 represents a poor solution while S_2 represents a good solution.

We have shown the migration curves in Figure 1 as straight lines, but in general they might be more complicated curves. We also assume that each solution has identical migration curves, but in general the migration curves could be adjusted on a per-solution basis. Nevertheless, the simple model shown in Figure 1 gives a general description of migration.

There are several different ways to implement the details of BBO. Figure 2 outlines the original BBO algorithm (Simon, 2008), which is called partial immigration-based BBO. In this approach, for each feature in each solution, we probabilistically decide whether or not to immigrate (i.e., replace that solution feature). If immigration is selected for a given feature, then the emigrating solution is probabilistically selected based on fitness (e.g., using roulette wheel selection). Migration and mutation of the entire

```

 $z \leftarrow y$ 
Define immigration probability  $\lambda_k \propto \text{fitness of the } k\text{th solution}$ 
Define emigration probability  $\mu_k = 1 - \lambda_k$ 
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 a BBO algorithm, where 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 .

Table 1: Comparison between GA and BBO.

	GA	BBO
Set of solutions	Population	Archipelago
Solution	Chromosome	Island
Solution feature	Allele	Species
Recombination	Crossover	Migration

population take place before any of the solutions are replaced, which requires the use of the temporary population z in Figure 2.

The BBO migration strategy is similar to the global recombination approach of evolutionary strategies (ES; Bäck, 1996; Bäck et al., 1997), in which many parents can contribute to a single offspring. Global recombination has also been adapted to GAs (Eiben, 2000, 2003), but BBO differs from GAs in one important aspect. In GAs, recombination is used to create new solutions, while in BBO, migration is used to change existing solutions. Global recombination in ES is a reproductive process which creates new solutions, while BBO migration is an adaptive process that modifies existing solutions. A quantitative comparison between BBO and other EAs is included in Simon (2008), where 14 benchmark functions, each with 20 dimensions, were studied. It was shown that BBO and the stud GA (so named for its selection of the best individual in the population as one of the parents for every crossover operation) performed the best out of eight EAs.

The EA which is most like BBO is a GA with global uniform recombination. However, there are still differences between the two. A conceptual comparison and contrast between GAs and BBO is shown in Table 1 and is discussed in more detail in Simon et al. (2009, 2010). Note in Table 1 that we propose the term archipelago to refer to a set of candidate solutions in BBO. However, in this paper, in order to retain familiar EA terminology, we use the term population to refer to a set of candidate solutions in BBO.

1.2 Paper Goals and Overview

The goals of this paper are twofold. Our first goal is to present a simplified version of BBO, which we do in Section 2. Our second goal is to use probability theory to

analyze some of the mathematical properties of simplified BBO (SBBO), which we do in the following sections. Section 3 derives the probability that the best individual in the SBBO population improves from one generation to the next. Section 4 derives an approximate Markov transition matrix for SBBO. Section 5 derives the expected amount of improvement in the best individual over one SBBO generation. We give supporting simulation results in Section 6, and provide concluding remarks and directions for future work in Section 7.

A more traditional Markov analysis, along the lines of that discussed for GAs in Nix and Vose (1992), Davis and Principe (1993), and Reeves and Rowe (2003), has been presented for the standard BBO algorithm in Simon et al. (2009, 2010). In that approach, each Markov state is a population distribution. The limitations of traditional Markov analyses of EAs is that the transition matrix grows factorially with the problem size. Each dimension of the traditional Markov matrix is $(n + N - 1)\text{-choose-}N$, where n is the cardinality of the search space and N is the population size. As an example, a 10-bit optimization problem ($n = 1,024$) with a population size $N = 10$ results in a transition matrix that has on the order of 10^{24} elements! Traditional Markov analysis provides insights into very simple problems, but because of computational limitations cannot be used for realistic problems.

In this paper we avoid the curse of dimensionality by defining the Markov states differently than those that are traditionally used. We partition the Markov state space depending on which individual is most fit and which individual has been selected for immigration. Each dimension of our Markov matrix is only N^2 , is independent of the search space cardinality, and is even independent of whether the search space is continuous or discrete. The limitation of our approach is that our Markov analysis is only approximate, but the advantage is that we can deal with realistically sized problems.

2 A Simplified BBO Algorithm

An SBBO algorithm can be formulated by always using the best solution as the emigrating island, and using any other solution with equal likelihood as the immigrating island. The immigrating island is chosen from a uniform probability distribution and is thus independent of fitness. This is conceptually similar to the stud GA in which the best chromosome is always chosen as one of the parents, and the other parent is chosen using standard fitness-based selection (Khatib and Fleming, 1998; Silva et al., 2005). In SBBO, the migration curves of Figure 1 are modified to those shown in Figure 3. One generation of SBBO can be described as follows:

1. Find the fittest solution. Call this solution x_i .
2. Pick a random solution feature s .
3. Select the immigrating island x_j from a uniform probability distribution ($j \neq i$).
4. $x_j(s) \leftarrow x_i(s)$.

With SBBO we do not need to evaluate all fitness values each generation. We only need to keep track of the most fit island. After each generation, we compare the new fitness of the immigrating island with the previous best solution. This requires only one fitness evaluation per generation.

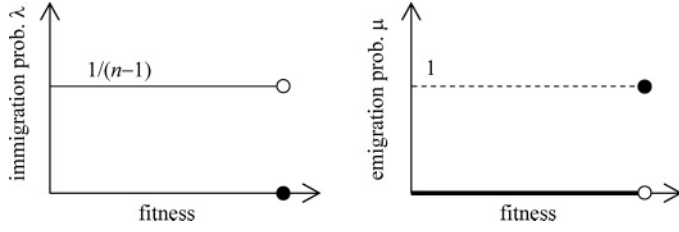


Figure 3: Illustration of SBBO migration curves in an n -island population. All solutions have an equal probability of immigration except for the most fit solution, which has a zero probability of immigration. The most fit solution has a 100% probability of emigration.

3 Probability Analysis

From this point on, we assume that we are trying to solve a minimization problem. High fitness corresponds to low cost. We try to minimize cost, which is always non-negative, and maximize fitness.

Suppose that the cost of a solution x_i is denoted as $f(x_i)$. We write

$$x_i = [x_{i1} \cdots x_{is}] \quad (1)$$

where s is the dimension of the problem and also the number of features in each solution. If $f(x_i)$ is separable, then it can be written as

$$f(x_i) = f_1(x_{i1}) + \cdots + f_s(x_{is}) \quad (2)$$

We make the assumption of separability in order to allow tractability in the following analysis. If $f(x_i)$ is not separable, then Equation (2) will not hold, but it may still hold approximately. The approximation error in our results from this point on are directly correlated with the approximation error of Equation (2) and are discussed further in Section 6.1.

If x_i is a randomly chosen island, then $f(x_i)$ is a random variable with mean \bar{f} and variance σ^2 . The parameters \bar{f} and σ can be approximated from a population of islands $\{x_i\}$. In the absence of any other information, we assume that the $f_j(\cdot)$ function values on the right-hand side of Equation (2) are independent identically distributed random variables. Then the expected value and variance of each $f_j(\cdot)$ can be written as

$$\begin{aligned} \bar{f}_j &= \bar{f}/s \\ \sigma_j^2 &= \sigma^2/s \end{aligned} \quad (3)$$

Now suppose that we migrate a single solution feature from the lowest-cost island to a randomly-chosen island. The contribution of a single solution feature to the cost of the immigrating island can be approximated as the random variable

$$y \sim (f_{\min}/s, \sigma^2/s) \quad (4)$$

where we are using the notation $y \sim (\mu, \sigma^2)$ to indicate that y is a random variable with mean μ and variance σ^2 . If y is a uniform random variable, then it can also be written as follows:

$$y \sim U[f_{\min}/s - \sigma\sqrt{3/s}, f_{\min}/s + \sigma\sqrt{3/s}] \quad (5)$$

where we are using the notation $y \sim U[a, b]$ to indicate that y is a uniform random variable whose probability density function (pdf) is nonzero only from a to b (Papoulis and Pillai, 2002).

The immigrating island can have any fitness $f \in [f_{\min}, f_{\max}]$. Therefore, before a solution feature is replaced in the immigrating island, that feature contributes a cost that can be approximated as the random variable

$$\begin{aligned} z &\sim (f/s, \sigma^2/s) \\ &\sim U[f/s - \sigma\sqrt{3/s}, f/s + \sigma\sqrt{3/s}] \end{aligned} \quad (6)$$

where the second expression for z applies if it is uniform. SBBO replaces a random solution feature in the immigrating island with a feature from the emigrating island. The probability that this feature replacement results in the immigrating island's fitness becoming lower than the emigrating island's fitness can be written as

$$\begin{aligned} p_s &= \text{Prob}(z - y > f - f_{\min}) \\ &= \text{Prob}(y < z - \Delta f) \end{aligned} \quad (7)$$

where Δf is the difference between the total cost of the immigrating and emigrating islands before migration. Figure 4 pictures this probability for two uniform random variables. The probability that the previously best island is replaced by a new best island due to migration is equal to the probability that a random variable taken from the z distribution, minus a random variable taken from the y distribution, is greater than $f - f_{\min}$.

Equation (7) can be written as

$$p_s = \int_{z_{\min}}^{z_{\max}} \int_{y_{\min}}^{z - \Delta f} \text{pdf}(y) \text{pdf}(z) dy dz \quad (8)$$

However, the upper limit of the inner integral must be greater than the lower limit of the outer integral. This implies that

$$\begin{aligned} z - \Delta f &> y_{\min} \\ z &> y_{\min} + \Delta f \end{aligned} \quad (9)$$

which changes the lower limit of the outer integral of Equation (8) accordingly. Also, Equation (8) applies to a specific immigrating island with fitness f . If the immigrating island is chosen randomly, then the fitness of the immigrating island can range from

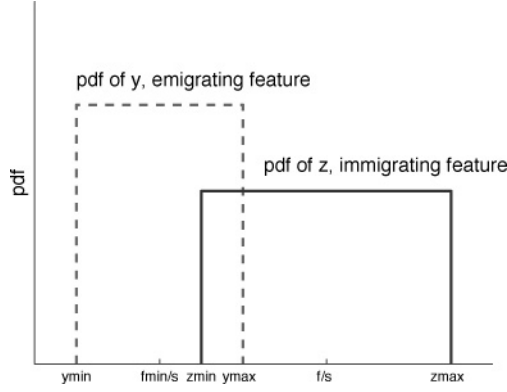


Figure 4: Sample probability distribution functions of an emigrating and immigrating solution feature. The probability that the immigrating island is better than the emigrating island after migration is equal to the probability that $z - y > f - f_{\min}$.

f_{\min} to f_{\max} . These ideas lead to a more general expression for Equation (8) as

$$p_s = \int_{f_{\min}}^{f_{\max}} \int_{y_{\min}+f-f_{\min}}^{z_{\max}} \int_{y_{\min}}^{z-f+f_{\min}} \text{pdf}(y) \text{pdf}(z) \text{pdf}(f) dy dz df \quad (10)$$

The pdf's in the above equation are entirely problem-dependent. Given a population, f_{\min} and f_{\max} are known. The pdf of f can be approximated by generating a large enough population of solutions. Similarly, the pdf of y and z can be approximated by quantifying the cost contribution of a large sample of randomly selected solution features.

Equation (10) gives the probability that the best individual in a population is replaced after a single generation of SBBO. This probability is approximately constant from one generation to the next, assuming that the population does not change much between generations. So the probability that there is no improvement in the global best of the population after each generation is $(1 - p_s)$. Therefore, on average we expect the best solution of the population to improve after k generations, where

$$(1 - p_s)^k = 0.5$$

$$k = \frac{\log(0.5)}{\log(1 - p_s)} \quad (11)$$

As the SBBO population increases, we have a better chance of beginning the optimization process with a smaller initial cost. But a larger population also typically causes the difference $(f_{\max} - f_{\min})$ to increase, which in turn causes p_s to decrease. This causes k , the expected number of generations between improvements in Equation (11), to increase. We see that a larger population size results in a better initial solution but a longer time between improvements in SBBO. This is illustrated qualitatively in Figure 5.

Small populations collapse to a single solution more quickly than large populations. This is intuitively obvious and is discussed quantitatively in the next section. As illustrated in Figure 5, improvements are not only more often for smaller populations, but also larger in magnitude. This is explored quantitatively in Section 5.

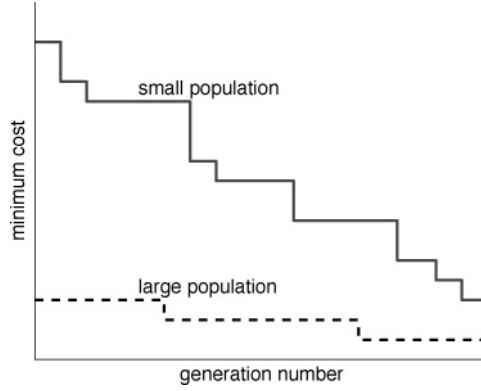


Figure 5: Illustration of the effect of population size on SBBO. A large population has a better initial solution, but more generations between improvements, and smaller improvements. A small population has a poorer initial solution, but fewer generations between improvements, and larger improvements.

4 Markov Analysis of the Simplified BBO

We define the Markov states of the n -island SBBO as follows.

- State $i \in [1, n]$: The i th island is the most fit population member and the immigrating island has not yet been selected.
- State $k \in [n + (i - 1)(n - 1) + j]$, where $i \in [1, n]$ and $j \in [1, n - 1]$: The i th island is the most fit population member and island m_{ij} has been selected for immigration, where

$$m_{ij} = \begin{cases} j & j < i \\ j + 1 & j \geq i \end{cases} \quad (12)$$

We see that an n -member population has n^2 states. As a simple illustrative example, consider a three-member population. The nine Markov states would consist of the following, where we use I to denote the immigrating island.

1. The first island is the most fit and I has not yet been selected.
2. The second island is the most fit and I has not yet been selected.
3. The third island is the most fit and I has not yet been selected.
4. The first island is the most fit and the second island has been selected for immigration.
5. The first island is the most fit and the third island has been selected for immigration.
6. The second island is the most fit and the first island has been selected for immigration.

7. The second island is the most fit and the third island has been selected for immigration.
8. The third island is the most fit and the first island has been selected for immigration.
9. The third island is the most fit and the second island has been selected for immigration.

The Markov transition probabilities for the general SBBO algorithm can be summarized as follows.

- If the SBBO is in state $i \in [1, n]$, then we have an equally likely probability of transitioning to state $n + i(n - 1) + j$ for each $j \in [1, n - 1]$. The probability of transitioning to each of these states is $1/(n - 1)$.
- If the SBBO is in state $n + i(n - 1) + j$ for some $i \in [1, n]$ and some $j \in [1, n - 1]$, then we have a probability p_s of transitioning to state m_{ij} , and a probability $(1 - p_s)$ of transitioning to state i . The probability p_s is given in Equation (10) and is the probability that migration to island m_{ij} results in that island becoming more fit than island i , which is the emigrating island. The probability $(1 - p_s)$ is the probability that after migration the emigrating island is still better than the immigrating island.

The state transition matrix is denoted as Q , where Q_{ij} is the probability of transitioning from state j to state i . In order to write Q , we first introduce some auxiliary variables.

$$L_{n-1} = \frac{1}{n-1} [1 \quad \cdots \quad 1]^T \in R^{(n-1) \times 1}$$

$$P_i = \begin{bmatrix} p_s & & & & & \\ & \ddots & & & & \\ & & p_s & & & \\ (1-p_s) & \cdots & (1-p_s) & (1-p_s) & \cdots & (1-p_s) \\ & & & p_s & & \\ & & & & \ddots & \\ & & & & & p_s \end{bmatrix} \leftarrow i\text{th row} \quad (13)$$

where the unspecified elements of P_i are zero. P_i is obtained by taking the $(n - 1) \times (n - 1)$ identity matrix, multiplying it by p_s , and then inserting as the i th row a vector which has each element equal to $(1 - p_s)$. P_i is therefore a matrix of dimension $n \times (n - 1)$.

Given these definitions we can write the $n^2 \times n^2$ Markov transition matrix for SBBO in block-matrix form as

$$Q = \begin{bmatrix} 0 & \cdots & 0 & P_1 & \cdots & P_n \\ L_{n-1} & & & & & \\ & \ddots & & & & \\ & & L_{n-1} & & & \end{bmatrix} \quad (14)$$

where the unspecified elements are equal to zero, and the number of L_{n-1} matrices is equal to n .

Q^2 describes the probability of transitioning from one state to another state after two state transitions. This can be calculated from Equation (14) as

$$Q^2 = \begin{bmatrix} \sum_{j=1}^n P_j L_{n-1,n}^{(j)} & 0 & \cdots & 0 \\ 0 & L_{n-1,n}^{(1)} P_1 & \cdots & L_{n-1,n}^{(1)} P_n \\ \vdots & \vdots & & \vdots \\ 0 & L_{n-1,n}^{(n)} P_1 & \cdots & L_{n-1,n}^{(n)} P_n \end{bmatrix} \quad (15)$$

where the $(n-1) \times n$ matrix $L_{n-1,n}^{(j)}$ contains all zeros except for the j th column, each of whose elements is equal to $1/(n-1)$. The upper left $n \times n$ block of Q^2 describes the transition probability from each of the first n states to each of the first n states after two state transitions. We use the symbol T to denote this transition matrix.

$$T = \sum_{j=1}^n P_j L_{n-1,n}^{(j)} \quad (16)$$

THEOREM 1: T is a regular transition matrix for $p_s \in (0, 1)$.

PROOF: See Appendix A.

THEOREM 2:

$$\begin{aligned} T^k &= \frac{1}{(n-1)^k} (N I_n + p_s 1_n)^k \\ &= \frac{1}{(n-1)^k} \left(N^k I_n + p_s \sum_{j=1}^k n^{k-j} N^{j-1} p_s^{k-j} \binom{k}{j-1} 1_n \right) \end{aligned} \quad (17)$$

where 1_n is the $n \times n$ matrix that contains all ones, and k -choose- j is denoted as

$$\binom{k}{j} = \frac{k!}{j!(k-j)!} \quad (18)$$

PROOF: See Appendix A.

Note that computing T^k requires about $\log_2 k$ matrix multiplications with the first expression of Equation (17), but does not require any matrix multiplications with the second expression of Equation (17).

THEOREM 3: SBBO reaches a population with uniform fitness as the number of generations approaches infinity.

PROOF: See Appendix A.

EXAMPLE: Appendix B gives a simple example of the theory up to this point.

5 Expected Improvement in the Simplified BBO

Section 3 gave the probability of improvement in one SBBO generation. This section discusses the expected improvement in the best population member, assuming that immigration improved the solution to the optimization problem.

Suppose that the best individual has cost f_{\min} , and it emigrates a solution feature to an individual with cost f . The improvement in the cost of the immigrating individual is denoted as Δf , which we consider to be positive if the cost decreases. Before migration, the replaced solution feature in the immigrating island contributed a cost z to the island as given in Equation (6). The emigrating solution feature contributes a cost y as given in Equation (4). The change in cost Δf due to immigration is therefore a random variable approximately given as

$$\begin{aligned}\Delta f &= z - y \\ &\sim (f/s - f_{\min}/s, 2\sigma^2/s) \\ &\sim (\mu_1, \sigma_1^2)\end{aligned}\tag{19}$$

where μ_1 and σ_1 are auxiliary variables defined by the above equation. We want to compute the expected value of cost improvement, given that improvement occurred due to migration. This can be written as

$$\begin{aligned}E(\Delta f | f - \Delta f < f_{\min}) &= E(\Delta f | \Delta f > f - f_{\min}) \\ &= \int_{f-f_{\min}}^{f_{\max}} \Delta f \text{pdf}'(\Delta f) d\Delta f\end{aligned}\tag{20}$$

The upper limit in the integral is f_{\max} because we are assuming that $f(x) \geq 0$ for all x , which implies that $\Delta f \leq f_{\max}$. The function $\text{pdf}'(\Delta f)$ is the same as the pdf given in Equation (19), except that it is scaled so that it has an area of one between $(f - f_{\min})$ and f_{\max} .

If Equation (19) is a Gaussian pdf, then Equation (20) can be written as

$$\begin{aligned}E(\Delta f | \Delta f > f - f_{\min}) &= \int_{f-f_{\min}}^{f_{\max}} \frac{x}{c_x \sigma_1 \sqrt{2\pi}} \exp\left(\frac{-(x - \mu_1)^2}{2\sigma_1^2}\right) dx \\ \text{where } c_x &= \frac{1}{2} \left[\text{erf}\left(\frac{f_{\max} - \mu_1}{\sigma_1 \sqrt{2}}\right) - \text{erf}\left(\frac{f - f_{\min} - \mu_1}{\sigma_1 \sqrt{2}}\right) \right] \\ \text{and } \text{erf}(x) &= \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2) dt\end{aligned}\tag{21}$$

Equation (21) is for a specific value of f . Since the immigrating island is randomly selected, we take the expected value of Equation (21) with respect to f to obtain

$$E(\Delta f | \Delta f > f - f_{\min}) = \int_{f_{\min}}^{f_{\max}} \int_{f-f_{\min}}^{f_{\max}} \frac{x}{c_x \sigma_1 \sqrt{2\pi}} \exp\left(\frac{-(x - \mu_1)^2}{2\sigma_1^2}\right) dx \text{pdf}'(f) df\tag{22}$$

The function $\text{pdf}'(f)$ is the pdf of f , but scaled so that it has an area of one between f_{\min} and f_{\max} . If f has a mean of \bar{f} and a variance of σ^2 , as discussed in the text following Equation (2), and if f is Gaussian, then Equation (22) can be written as

$$\begin{aligned}
 E(\Delta f | \Delta f > f - f_{\min}) &= \int_{f_{\min}}^{f_{\max}} \int_{f - f_{\min}}^{f_{\max}} \frac{x}{c_x c_y \sigma_1 \sigma 2\pi} \exp\left(\frac{-(x - \mu_1)^2}{2\sigma_1^2}\right) \\
 &\quad \times \exp\left(\frac{-(f - \bar{f})^2}{2\sigma^2}\right) dx df \\
 c_y &= \frac{1}{2} \left[\text{erf}\left(\frac{f_{\max} - \bar{f}}{\sigma\sqrt{2}}\right) - \text{erf}\left(\frac{f_{\min} - \bar{f}}{\sigma\sqrt{2}}\right) \right]
 \end{aligned} \tag{23}$$

6 Simulation Results

SBBO was simulated on some benchmark functions in order to confirm the theoretical results of the preceding sections. For each function, we ran 100 Monte Carlo SBBO simulations with population sizes of 4, 20, and 100. We recorded the average number of generations that was required to improve the best population member. We also recorded the average improvement of the best population member. We compared these numbers with the expected number of generations before improvement as given in Equation (11), and the expected improvement amount as given in Equation (23). Table 2 shows the results for four-dimensional benchmarks, and Table 3 shows the results for 10-dimensional benchmarks. More information about the benchmark functions, including their domains, can be found in Bäck (1996), Aluffi-Pentini et al. (1985), Yao et al. (1999), Cai and Wang (2006), Feng et al. (1998), and Li et al. (2008).

Note that Equations (11) and (23) in Tables 2 and 3 are calculated on the basis of pdf's, f_{\min} , f_{\max} , and \bar{f} . These quantities are not available analytically, and so they must be approximated on the basis of random populations. We therefore used 100 random populations to approximate these quantities so that we could calculate Equations (11) and (23).

We make the following observations about Tables 2 and 3.

1. For a given benchmark, the number of generations before improvement increases as the population size increases. This confirms the intuitive result that the probability per generation of finding a better solution decreases as the population size increases.
2. The expected number of generations before finding a better solution as calculated from Equation (11) matches reasonably well with the simulation results. The simulation results can vary widely from run to run as shown by the large standard deviations, but the theoretical results are well within one standard deviation from the mean of the simulation results.
3. The expected improvement amount from Equation (23) matches reasonably well with the simulation results. Also, as shown qualitatively in Figure 5, we see from Tables 2 and 3 that for each benchmark the expected improvement decreases as the population size increases.

Table 2: Theoretical and simulation results for four-dimensional benchmarks. The four right-most columns show the expected number of generations until the first improvement in the best problem solution, and the expected amount of that improvement. Simulation results are based on 100 Monte Carlo runs and are shown as mean ± 1 *SD*.

Benchmark function	Population size	Expected number of generations		Expected improvement	
		Eq. (11)	Simulation	Eq. (23)	Simulation
Ackley (nonseparable)	4	3.7	3.9 ± 4.5	0.48	1.3 ± 2.0
	20	30	20 ± 25	0.85	1.0 ± 1.2
	100	11	54 ± 59	1.2	0.90 ± 1.1
Griewank (nonseparable)	4	6.2	6.1 ± 3.4	11	11 ± 8.5
	20	20	19 ± 21	9.8	4.9 ± 6.2
	100	82	85 ± 144	0.0	1.6 ± 2.6
Penalty 1 (nonseparable)	4	7.4	5.1 ± 3.8	3.3E7	$(3.2 \pm 4.6)\text{E}7$
	20	17	21 ± 21	3.4E6	$(1.3 \pm 3.0)\text{E}6$
	100	49	64 ± 83	2.6E5	$(1.3 \pm 5.4)\text{E}5$
Penalty 2 (nonseparable)	4	6.6	6.4 ± 4.8	4.8E7	$(3.8 \pm 5.5)\text{E}7$
	20	16	17 ± 15	1.7E7	$(6.1 \pm 9.2)\text{E}6$
	100	53	74 ± 66	2.4E6	$(1.7 \pm 4.9)\text{E}6$
Rastrigin (separable)	4	3.6	3.9 ± 3.0	5.1	5.2 ± 5.7
	20	20.4	22 ± 29	4.3	4.1 ± 3.8
	100	66	97 ± 146	2.6	2.9 ± 3.0
Schwefel 2.21 (nonseparable)	4	3.6	2.6 ± 0.9	6.8	14.3 ± 9.7
	20	25	25 ± 29	10	7.7 ± 6.1
	100	186	138 ± 165	0.0	3.6 ± 3.9
Schwefel 2.26 (separable)	4	4.9	4.8 ± 3.9	156	148 ± 112
	20	9.8	12 ± 13	98	94 ± 68
	100	43	43 ± 62	90	70 ± 48
Shubert (separable)	4	4.9	3.8 ± 2.7	1.2	1.3 ± 1.1
	20	27	22 ± 29	0.69	0.52 ± 0.50
	100	46	47 ± 51	0.93	0.24 ± 0.24
Sine Product (separable)	4	5.8	4.8 ± 3.4	0.15	0.13 ± 0.12
	20	21	19 ± 17	0.097	0.055 ± 0.065
	100	34	45 ± 46	0.064	0.012 ± 0.018
Sphere (separable)	4	4.0	3.9 ± 2.7	3.6	3.9 ± 3.7
	20	25	24 ± 21	2.1	1.9 ± 2.0
	100	36	45 ± 46	0.0	0.5 ± 1.0
Step (separable)	4	7.4	8.2 ± 6.0	1,496	$1,522 \pm 1,241$
	20	17	17 ± 19	981	911 ± 890
	100	78	58 ± 57	68	235 ± 291
Weierstrass (separable)	4	5.3	5.0 ± 5.4	0.59	0.68 ± 0.37
	20	15	18 ± 16	0.48	0.45 ± 0.36
	100	53	60 ± 75	0.36	0.31 ± 0.33

6.1 Approximation Errors

Although we see good correlation between the theoretical and simulation results, there are still noticeable differences between the two sets of results. This is due to the approximations that were used in obtaining Equations (11) and (23). These approximations include the following assumptions.

1. We assumed separability, as seen in Equation (2). Many problems are separable, but many others are not. For real-world problems we cannot always tell whether

Table 3: Theoretical and simulation results for 10-dimensional benchmarks. The four right-most columns show the expected number of generations until the first improvement in the best problem solution, and the expected amount of that improvement. Simulation results are based on 100 Monte Carlo runs and are shown as mean ± 1 *SD*.

Benchmark function	Population size	Expected number of generations		Expected improvement	
		Eq. (11)	Simulation	Eq. (23)	Simulation
Ackley (nonseparable)	4	13	13 ± 14	0.074	0.18 ± 0.21
	20	65	40 ± 44	0.0	0.36 ± 0.38
	100	20	113 ± 113	3.6	0.38 ± 0.36
Griewank (nonseparable)	4	11	11 ± 9.4	16	18 ± 14
	20	46	34 ± 35	11	9.6 ± 9.4
	100	116	124 ± 122	10	6.1 ± 9.2
Penalty 1 (nonseparable)	4	13	15 ± 21	3.6E7	$(3.4 \pm 3.1)\text{E}7$
	20	44	44 ± 42	1.6E7	$(1.4 \pm 2.0)\text{E}7$
	100	192	161 ± 144	4.1E6	$(1.5 \pm 3.5)\text{E}6$
Penalty 2 (nonseparable)	4	13	15 ± 15	7.0E7	$(7.5 \pm 7.3)\text{E}7$
	20	43	43 ± 36	2.2E7	$(3.0 \pm 3.6)\text{E}7$
	100	197	134 ± 132	4.2E6	$(4.5 \pm 9.9)\text{E}6$
Rastrigin (separable)	4	8.6	9.9 ± 9.9	7.8	6.9 ± 6.3
	20	38	38 ± 46	4.8	5.5 ± 4.5
	100	128	126 ± 132	4.0	4.7 ± 4.8
Schwefel 2.21 (nonseparable)	4	13	16 ± 14	1.6	7.3 ± 5.7
	20	98	71 ± 65	3.3	5.1 ± 5.0
	100	17	272 ± 233	4.9	3.6 ± 3.0
Schwefel 2.26 (separable)	4	15	15 ± 16	147	145 ± 116
	20	45	49 ± 42	114	136 ± 112
	100	109	102 ± 136	139	104 ± 88
Shubert (separable)	4	11	11 ± 12	2.8	3.3 ± 3.1
	20	41	36 ± 39	1.4	1.4 ± 1.7
	100	133	129 ± 108	1.1	0.92 ± 1.0
Sine product (separable)	4	9	9 ± 8	0.19	0.21 ± 0.22
	20	48	47 ± 44	0.15	0.15 ± 0.15
	100	125	130 ± 114	0.14	0.056 ± 0.081
Sphere (separable)	4	11	11 ± 12	5.1	5.1 ± 4.7
	20	38	40 ± 35	3.7	3.9 ± 4.2
	100	140	150 ± 150	2.9	1.3 ± 1.7
Step (separable)	4	9.3	9.3 ± 10.6	2,081	$2,076 \pm 1,802$
	20	76	62 ± 79	1,280	$1,014 \pm 1,262$
	100	115	133 ± 122	392	$948 \pm 1,095$
Weierstrass (separable)	4	11	11 ± 11	0.48	0.58 ± 0.47
	20	37	39 ± 44	0.33	0.48 ± 0.35
	100	109	107 ± 127	0.58	0.37 ± 0.30

or not the problem is separable. We made the assumption of separability in order to allow tractability in our analysis. Table 4 gives the average approximation errors from the previous section, and shows that (as expected) the approximation errors are smaller for the separable cost functions than for the nonseparable functions.

2. We made the assumption of identical pdf's for each solution feature as seen in Equation (3). In other words, we assumed that each solution feature contributes an equal amount to the fitness of the cost function.

Table 4: Average percent errors between the theoretical and simulation results in Tables 2 and 3. As expected, the errors are smaller for the separable functions than for the nonseparable functions.

Dimensions	Nonseparable functions		Separable functions	
	Number of generations	Improvement	Number of generations	Improvement
4	26%	67%	14%	30%
10	29%	34%	5%	25%

3. We made the assumption of normality in Section 5. That is, we assumed that a uniformly distributed population of solutions gives a normally distributed fitness function. Although this is only an approximation, it is a reasonable one in view of the central limit theorem (Papoulis and Pillai, 2002), which says that a combination of independent random variables tends toward a Gaussian random variable.

In spite of these assumptions and approximations, the simulation and theoretical results in the previous section are still close enough that the theory can be used to at least qualitatively predict SBBO performance. However, future work should be pursued to relax some of these assumptions.

For example, given a real-world problem with an unknown functional form, separability can be tested by evaluating the cost function at various values of the independent variables. Furthermore, individual contributions of independent variables to fitness, and correlations between independent variables, can be tested. Depending on the results of these statistical tests, Equation (2) can be modified for a specific problem. For example, it could be rewritten as

$$f(x_i) = \alpha_1 f_1(x_{i1}) + \alpha_2 f_2(x_{i2}) + \rho_{12} f_1(x_{i1}) f_2(x_{i2}) \quad (24)$$

where α_1 , α_2 , and ρ_{12} are determined from statistical tests. The analysis in Sections 3 and following could then be repeated for this form of the cost function.

The assumption of Gaussian cost functions could also be relaxed, either for other general cases or for specific optimization problems. In the general case, the analysis in Section 5 could be repeated for fitness distributions other than Gaussian. In case we have a specific optimization problem, the fitness distribution could be numerically obtained by generating a random population of solutions. We could then fit a weighted sum of Gaussians to the empirical distribution (Alspach, 1974). The double integral on the right-hand side of Equation (23) would then be replaced with a weighted sum of integrals. Kernels other than Gaussian could also be used to approximate the cost function pdf (Simonoff, 1998; Devroye and Lugosi, 2001), which would change Equation (21) and the following equations.

In summary, the analysis in this paper has been conducted with specific assumptions (separability, identical pdf's, and normality), but the more important point is that our analysis provides a general framework that can be used even when the specific assumptions in this paper do not hold. The work presented in this paper opens up a wide range of future research possibilities.

Table 5: Average MATLAB computational effort (in seconds) on a 2.0 GHz PC for the calculation of the triple integral of Equation (10) and the double integral of Equation (23) for each benchmark. The computation time in the table is dominated by that of Equation (10).

Population Size	Dimension	
	4	10
4	149	98
20	99	81
100	77	117

6.2 Computational Effort

The computational effort required to obtain the theoretical results in this paper depends on the level of accuracy required. The bulk of the theory computation is dominated by the numerical calculation of the triple integral in Equation (10) and the double integral in Equation (23). In this paper we used the MATLAB integration functions `dblquad` and `triplequad`. The computer time required for these functions strongly depends on their error tolerance parameters. A smaller error tolerance gives more accurate integration results but requires more integrand evaluations. For the results in this paper, we used an error tolerance of 10^{-8} (the default MATLAB tolerance is 10^{-6}). Ironically, as the population size and the number of dimensions increase, the computational effort often decreases, as shown in Table 5. This is because a larger population size and a higher dimensionality results in a wider distribution of the cost function, that is, a larger standard deviation for $\text{pdf}(f)$ in Equation (10), and larger values for σ and σ_1 in Equation (23). This in turn results in smoother integrands and faster convergence of the integration routines.

However, the reason for our theoretical analysis is not to reduce computational effort relative to simulation. In fact, computational effort cannot even be fairly compared between theory and simulation. Theory only gives the expected number of generations to the first improvement, and the expected amount of improvement. Simulation, though, typically runs for a specified number of generations, or until some type of convergence is achieved.

The reason for our theoretical analysis is to propose new tools for the study of BBO populations, and more generally, for populations of other types of EAs. Since the characteristics of random population distributions depend on the specific problem, the problem dimension, and the population size, the results of our analysis will also change with these parameters. Our tools include analytical expressions that could therefore be studied to learn how BBO behavior (and perhaps other EA behavior in future work) depends on these parameters. In addition, changes to the BBO setup (elitism, mutation, migration curve shape, etc.) could possibly be incorporated into our analysis in the future to see how these changes affect BBO behavior.

These issues can all be studied by simulation also, but simulation results are difficult to generalize. For example, a simulation might tell us that a population size of n provides a good performance/computation tradeoff for a particular problem. But analysis might tell us that a population size of n provides a good performance/computation tradeoff for all problems with a prescribed level of separability and with a specific fitness distribution.

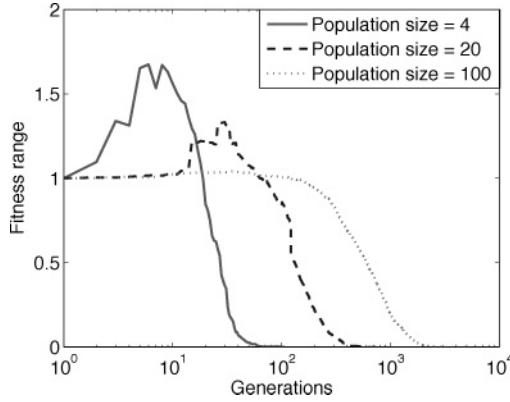


Figure 6: Normalized ($f_{\max} - f_{\min}$) averaged over 14 benchmarks and 100 simulations: four-dimensional results.

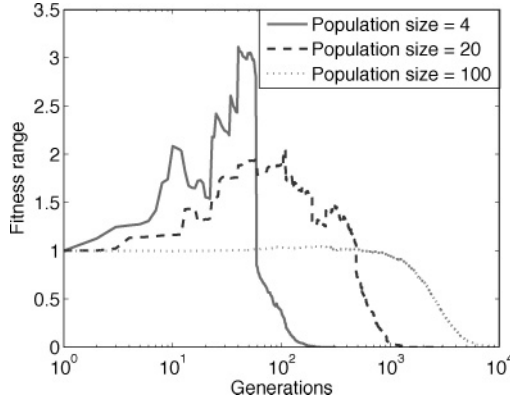


Figure 7: Normalized ($f_{\max} - f_{\min}$) averaged over 14 benchmarks and 100 simulations: ten-dimensional results.

6.3 Uniformity

Figures 6 and 7 show how the normalized fitness range of SBBO changes with the number of generations. Theorem 3 predicted that SBBO would reach a uniform population, and Figures 6 and 7 confirm this. In agreement with intuition, the number of generations before reaching uniformity is proportional to both the population size and the problem dimension. Interestingly, the fitness range increases at the beginning of the simulations as f_{\min} rapidly decreases. This is especially true for small populations (compare with Figure 5). Later in the simulations, all of the individuals improve until uniformity is eventually achieved.

Note that a uniform population is not necessarily desirable. There is no guarantee that a uniform population is optimal. It is only with mutation that we can be sure of obtaining an optimal solution for a given optimization problem. In fact, this is a conclusion that can be generally inferred from this paper. See Reeves and Rowe (2003) for a discussion of this idea as applied to GAs.

In theory, the number of generations until SBBO reaches uniformity could be predicted by Theorem 3. This theorem gives us the ratio of the largest and smallest elements of the Markov transition matrix, so when that ratio falls below some small threshold after a certain number of generations, we can assume that the population is uniform and no more improvement is possible. In practice, however, this is difficult to calculate because Theorem 3 assumes that p_s is constant, whereas p_s actually changes with each generation. The use of the Markov theory in Section 4 to predict the number of generations until uniformity is reached has therefore been left for future work.

7 Conclusion

A simplified BBO (SBBO) algorithm has been presented in this paper. SBBO has been analyzed using probability theory in order to find three related quantities: the probability per generation that its population optimum improves, the state transition matrix of the algorithm, and the expected amount of improvement in the population optimum. These quantities are inexact in light of assumptions and approximations that were made, but they match benchmark function simulation results reasonably well.

Further work could focus on reducing the approximation errors that were used to obtain the theoretical results in this paper. In this paper we assumed separable fitness functions, equal fitness contribution from each independent variable, and normal distributions for the fitness and for the contribution of each parameter to total fitness. The development in this paper could be generalized so that it matches other problem characteristics, or so that those assumptions could be relaxed. Future work could also include the extension of our analysis to other types of EAs.

Acknowledgment

This work was supported by NSF Grant 0826124 in the CMMI Division of the Engineering Directorate.

References

- Alspach, D. (1974). Gaussian sum approximations in nonlinear filtering and control. *Information Sciences*, 7:271–290.
- Aluffi-Pentini, F., Parisi, V., and Zirilli, F. (1985). Global optimization and stochastic differential equations. *Journal of Optimization Theory and Applications*, 47:1–16.
- Bäck, T. (1996). *Evolutionary algorithms in theory and practice*. Oxford, UK: Oxford University Press.
- Bäck, T., Hammel, U., and H.-P. Schwefel (1997). Evolutionary computation: Comments on the history and current state. *IEEE Transactions on Evolutionary Computation*, 1:3–17.
- Cai, Z., and Wang, Y. (2006). A multiobjective optimization-based evolutionary algorithm for constrained optimization. *IEEE Transactions on Evolutionary Computation*, 10:658–675.
- Chuan-Chong, C., and Khee-Meng, K. (1992). *Principles and techniques in combinatorics*. Singapore: World Scientific.
- Davis, T., and Principe, J. (1993). A Markov chain framework for the simple genetic algorithm. *Evolutionary Computation*, 1:269–288.
- Devroye, L., and Lugosi, G. (2001). *Combinatorial methods in density estimation*. Berlin: Springer.

- Eiben, A. (2000). Multiparent recombination. In T. Bäck, D. Fogel, and Z. Michalewicz (Eds.), *Evolutionary computation 1: Basic algorithms and operators* (pp. 289–302). London: Institute of Physics Press.
- Eiben, A. (2003). Multiparent recombination in evolutionary computing. In A. Ghosh and S. Tsutsui (Eds.), *Advances in Evolutionary Computing*, pp. 175–192.
- Feng, W., Brune, T., Chan, L., Chowdhury, M., Kuek, C., and Li, Y. (1998). Benchmarks for testing evolutionary algorithms. In *Asia-Pacific Conference on Control and Measurement*, pp. 134–138.
- Khatib, W., and Fleming, P. (1998). The stud GA: A mini revolution? In A. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel (Eds.), *Parallel Problem Solving from Nature*, pp. 683–691.
- Li, C., Yang, S., Nguyen, T., Yu, E., Yao, X., Jin, Y., H.-G. Beyer, and Suganthan, P. (2008). Benchmark generator for CEC 2009 competition on dynamic optimization. Technical report, Department of Computer Science, University of Leicester, U.K. Retrieved from www.cs.le.ac.uk/people/syang/publications.html.
- Lomolino, M., Riddle, B., and Brown, J. (2009). *Biogeography* (3rd ed.). Sunderland, MA: Sinauer Associates.
- MacArthur, R., and Wilson, E. (1967). *The theory of biogeography*. Princeton, NJ: Princeton University Press.
- Nix, A., and Vose, M. (1992). Modeling genetic algorithms with Markov chains. *Annals of Mathematics and Artificial Intelligence*, 5:79–88.
- Papoulis, A., and Pillai, S. (2002). *Probability, random variables, and stochastic processes*. New York: McGraw-Hill.
- Reeves, C., and Rowe, J. (2003). *Genetic algorithms: Principles and perspectives*. Dordrecht, The Netherlands: Kluwer.
- Silva, V., Khatib, W., and Fleming, P. (2005). Performance optimization of gas turbine engine. *Engineering Applications of Artificial Intelligence*, 18:575–583.
- Simon, D. (2008). Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation*, 12(6):702–713.
- Simon, D., Ergezer, M., and Du, D. (2009). Population distributions in biogeography-based optimization algorithms with elitism. In *IEEE Conference on Systems, Man, and Cybernetics*, pp. 1017–1022.
- Simon, D., Rarick, R., Ergezer, M., and Du, D. (2010). Analytical and numerical comparisons of biogeography-based optimization and genetic algorithms. Submitted for publication, <http://embeddedlab.csuohio.edu/BBO>.
- Simonoff, J. (1998). *Smoothing methods in statistics*. Berlin: Springer.
- Yao, X., Liu, Y., and Lin, G. (1999). Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, 3:82–102.

Appendix A

PROOF OF THEOREM 1: From Equation (16) we see that T is a sum of n matrices. If $p_s \in (0, 1)$, then each element in the j th column of the j th matrix in the sum is greater than zero. So when these matrices are added together, the result is a matrix that contains all nonzero elements.

Theorem 1 can also be proven with a more intuitive argument. T_{ij} is the probability that island j is replaced by island i as the most fit island after one migration. But the probability that island j migrates to island i is nonzero, $1/(n-1)$ to be exact, and the probability that this migration results in island i replacing island j as the most fit island is given by the nonzero quantity of Equation (10).

By expanding the sum in Equation (16) it can be seen that

$$T = \frac{1}{n-1} (NI_n + p_s 1_n) \quad (25)$$

where $N = n - 1 - np_s$, I_n is the $n \times n$ identity matrix, and 1_n is the $n \times n$ matrix that contains all ones. **QED**

PROOF OF THEOREM 2: It can be seen from Equation (25) that Equation (17) holds for $k = 1$. Suppose that Equation (17) holds for some value of k . Then

$$\begin{aligned} T^{k+1} &= \frac{1}{(n-1)^{k+1}} \left(N^k I_n + p_s \sum_{j=1}^k n^{k-j} N^{j-1} p_s^{k-j} \binom{k}{j-1} 1_n \right) (NI_n + p_s 1_n) \\ &= \frac{1}{(n-1)^{k+1}} \left(N^{k+1} I_n + p_s \sum_{j=1}^k n^{k-j} N^j p_s^{k-j} \binom{k}{j-1} 1_n + N^k p_s 1_n \right. \\ &\quad \left. + p_s^2 \sum_{j=1}^k n^{k-j+1} N^{j-1} p_s^{k-j} \binom{k}{j-1} 1_n \right) \\ &= \frac{1}{(n-1)^{k+1}} \left(N^{k+1} I_n + p_s \sum_{j=1}^k n^{k-j} N^j p_s^{k-j} \binom{k}{j-1} 1_n \right. \\ &\quad \left. + p_s \sum_{j=1}^{k+1} n^{k-j+1} N^{j-1} p_s^{k-j+1} \binom{k}{j-1} 1_n \right) \\ &= \frac{1}{(n-1)^{k+1}} \left(N^{k+1} I_n + p_s \sum_{j=1}^k n^{k-j} N^j p_s^{k-j} \binom{k}{j-1} 1_n \right. \\ &\quad \left. + p_s \sum_{j=1}^{k+1} n^{k-j} N^j p_s^{k-j} \binom{k}{j} 1_n + n^k p_s^{k+1} 1_n \right) \\ &= \frac{1}{(n-1)^{k+1}} \left(N^{k+1} I_n + p_s \sum_{j=1}^k n^{k-j} N^j p_s^{k-j} \left[\binom{k}{j-1} + \binom{k}{j} \right] 1_n \right. \\ &\quad \left. + n^k p_s^{k+1} 1_n \right) \end{aligned} \quad (26)$$

Recall from Chuan-Chong and Khee-Meng (1992, p. 69) that

$$\binom{k}{j-1} + \binom{k}{j} = \binom{k+1}{j} \quad (27)$$

Use this in Equation (26) to obtain

$$\begin{aligned} T^{k+1} &= \frac{1}{(n-1)^{k+1}} \left(N^{k+1} I_n + p_s \sum_{j=1}^k n^{k-j} N^j p_s^{k-j} \binom{k+1}{j} 1_n + n^k p_s^{k+1} 1_n \right) \\ &= \frac{1}{(n-1)^{k+1}} \left(N^{k+1} I_n + p_s \sum_{j=0}^k n^{k-j} N^j p_s^{k-j} \binom{k+1}{j} 1_n \right) \\ &= \frac{1}{(n-1)^{k+1}} \left(N^{k+1} I_n + p_s \sum_{j=1}^{k+1} n^{k+1-j} N^{j-1} p_s^{k+1-j} \binom{k+1}{j-1} 1_n \right) \end{aligned} \quad (28)$$

which completes the proof of Equation (17) by induction. **QED**

PROOF OF THEOREM 3: Equation (17) shows that the diagonal elements of T^k are all equal, and the off-diagonal elements are all equal. The difference between a diagonal element and an off-diagonal element is $[(n-1-np_s)/(n-1)]^k$. This difference approaches 0 as $k \rightarrow \infty$. Therefore,

$$\lim_{k \rightarrow \infty} T^k = 1_n/n \quad (29)$$

In other words, as the number of generations approaches infinity, each island in the population has an equal probability of being the most fit member of the population. This means that each island is equally fit. **QED**

Appendix B

Here we give a simple example to illustrate the theory of Sections 1–4. Suppose we have a two-dimensional ($s = 2$) fitness function that is uniformly distributed between $f_{\min} = 1$ and $f_{\max} = 2$. Then y in Equation (5) is uniformly distributed between $y_{\min} = 1/2 - 1/\sqrt{8}$ and $y_{\max} = 1/2 + 1/\sqrt{8}$, and z in Equation (6) is uniformly distributed between $z_{\min} = f/2 - 1/\sqrt{8}$ and $z_{\max} = f/2 + 1/\sqrt{8}$. The magnitudes of the pdf's of y and z are both equal to $\sqrt{2}$ between their minimum and maximum values. p_s can be calculated from Equation (10) as

$$p_s = (7/3 - \sqrt{2})/4 \approx 0.23 \quad (30)$$

Now suppose that we have $n = 3$ members in our SBBO population. The transition matrix of Equation (14) is then equal to

$$Q = \begin{bmatrix} 0 & 0 & 0 & (1-p_s) & (1-p_s) & p_s & 0 & p_s & 0 \\ 0 & 0 & 0 & p_s & 0 & (1-p_s) & (1-p_s) & 0 & p_s \\ 0 & 0 & 0 & 0 & p_s & 0 & p_s & (1-p_s) & (1-p_s) \\ 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (31)$$

Note that each column of Q sums up to one. The transition matrix of the first three states can be calculated from the upper left block of Equations (15) or (16) as

$$T = \begin{bmatrix} (1-p_s) & p_s/2 & p_s/2 \\ p_s/2 & (1-p_s) & p_s/2 \\ p_s/2 & p_s/2 & (1-p_s) \end{bmatrix} \quad (32)$$

where we again notice that each column of T sums to one. We can easily verify that $\lim_{k \rightarrow \infty} T^k = 1_n/3$, which confirms Theorem 3 and Equation (29).