Theoretical Analysis of Genetic Algorithms in Noisy Environments based on a Markov Model [*]

Takéhiko Nakama
Department of Applied Mathematics and Statistics
The Johns Hopkins University
3400 N. Charles Street
Baltimore, MD 21218
nakama@jhu.edu

ABSTRACT

In this study, we take a first step towards theoretically analyzing genetic algorithms (GAs) in noisy environments using Markov chain theory. We explicitly construct a Markov chain that models GAs applied to fitness functions perturbed by either additive or multiplicative noise that takes on finitely many values, and we analyze the chain to investigate the transition and convergence properties of the GAs. For the additive case, our analysis shows that GAs eventually (i.e., as the number of iterations goes to infinity) find at least one globally optimal solution with probability 1. In contrast, GAs may eventually with probability 1 fail to do so in the multiplicative case, and we establish a condition that is both necessary and sufficient for eventually finding a globally optimal solution. In addition, our analysis shows that the chain has a stationary distribution that is also its steady-state distribution. Based on this property, we derive an upper bound for the number of iterations sufficient to ensure with certain probability that a GA has reached the set of globally optimal solutions and continues to include in each subsequent population at least one globally optimal solution whose observed fitness value is greater than that of any suboptimal solution.

Categories and Subject Descriptors

G.3 [**Mathematics of Computing**]: Probability and Statistics—*probabilistic algorithms, Markov processes*;
F.2.m [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity—*miscellaneous*

General Terms

Algorithms, theory, performance

Keywords

Evolutionary computation, genetic algorithms, Markov chain analysis, noisy environments, additive noise, multiplicative noise, perturbed fitness functions, convergence

1. INTRODUCTION AND SUMMARY

In many practical problems, objective functions are perturbed by random noise, and genetic algorithms (GAs) have been widely proposed as an effective optimization tool for dealing with noisy objective functions (e.g., Beyer [2], Chen, Subprasom, and Ji [3]). Although quite a few studies (e.g., Goldberg and Rudnick [6], Miller and Goldberg [11], Nissen and Propach [12], Beyer [2], Arnold [1], Di Pietro, While, and Barone [5]) have examined GAs and more general evolutionary algorithms in noisy environments using either numerical or other theoretical methods, Markov chain analysis has not been applied to such GAs. (For a survey of the literature, we refer the reader to Beyer [2] and Jin and Branke [8].) This is quite contrastive to the noiseless case; Markov chain theory has been effectively used to reveal important properties of GAs applied to noiseless fitness functions (e.g., Vose and Liepins [19], Nix and Vose [13], Davis and Principe [4], Rudolph [16], Suzuki [17], Vose [18]).

In this study, we take a first step towards theoretically analyzing GAs in noisy environments using Markov chain theory. More precisely, we apply Markov chain theory to investigate the transition and convergence properties of GAs applied to fitness functions that are perturbed by either additive or multiplicative noise (these two types of noise are fully characterized in Sections 3.1 and 4.1). Theoretical studies that examine evolutionary computation schemes applied to perturbed fitness functions typically assume that the noise is additive and has an independent and identical Gaussian distribution with mean zero (e.g., Miller and Goldberg [11], Beyer [2]). To our knowledge, this study is the first to elucidate the transition and convergence properties of GAs applied to fitness functions perturbed by multiplicative noise, which has been considered in a variety of practical optimization problems such as control optimization (e.g., Hopkins [7]), image restoration (e.g., Rudin and Osher [15]), and portfolio optimization (e.g., Primbs [14]). For the multiplicative noise case, the degree of fitness disturbance may not be the same at all points in the search space. For example, multiplicative noise can perturb large fitness values more severely than small fitness values. Also, if multiplicative noise is negative, then the sign of perturbed fitness is the opposite of that of noiseless fitness. In this study, we do

---

[*] Supported by the Acheson J. Duncan Fund for the Advancement of Research in Statistics.

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.

not assume that multiplicative noise is positive with probability 1, so we cannot simply take the logarithm of (noisy) observed fitness.

For analytical tractability, we assume in this study that both the additive noise and the multiplicative noise take on finitely many values. However, we do not make any assumptions about their expected values or variances. The two types of noise are fully characterized in Sections 3.1 and 4.1. We are currently extending the Markov chain analysis described in this study to other noisy environments—for example, environments with continuous noise.

GAs in our study are assumed to implement the elitist strategy and to reevaluate the fitness value of each population member except for that of the chromosome preserved by the elitist strategy every time a new population is formed. It will become clear that these strategies are essential for ensuring that GAs eventually reach the set of globally optimal solutions and continue to include at least one globally optimal solution in each subsequent population with probability 1 when they are applied to fitness functions perturbed by the additive noise described above. The reassessment of the fitness value also significantly reduces memory requirements compared to the strategy of storing a fitness value for every chromosome that has been included in some population formed since the beginning of the execution. For analytical simplicity, we will not consider noise compensation techniques such as averaging (e.g., Beyer [2], Di Pietro, While, and Barone [5]) in this study. However, we are currently extending our Markov chain analysis to GAs executed with these techniques in noisy environments. Sections 2 and 3.1 fully describe the implementation of GAs considered in this study.

We explicitly construct a Markov chain that models GAs applied to perturbed fitness functions. This Markov chain is fundamentally different from those considered for the noiseless environment in one essential aspect: Each state of the chain for the noisy case is not a population but a set of *ordered pairs*, each of which consists of a chromosome and its observed noise value [the ordered pairs are defined at (3.4)]. Markov chains considered to analyze the noiseless case typically have a state space that consists of distinct populations. However, these Markov chains fail to explicitly capture the evolution of GAs in noisy environments. The construction of the Markov chain for the additively noisy case is fully explained in Section 3.2. We explicitly compute the transition probabilities of the chain; see Theorem 3.1. It turns out that these probabilities can be effectively used to bound the number of iterations sufficient to ensure with certain probability that a GA selects a globally optimal solution upon termination.

We first describe the Markov chain analysis for the additive noise case. The additive case turns out to be much simpler than the multiplicative case, although the analytical techniques applied to the additive case can essentially be carried over to the multiplicative case. Under the assumptions mentioned above, we show that the Markov chain in the additive case (and in the multiplicative case) is indecomposable; it has only one positive recurrent communication class (a communication class is a set of states that are accessible to each other and is positive recurrent if the expected return time for each of its states is finite). This is Theorem 3.2. For the additive case, it follows immediately from this theorem that GAs eventually find at least one globally optimal solution with probability 1. (In this regard, the multiplicative case is strikingly different from the additive case.) Theorem 3.3 states that the chain has a stationary distribution that is also its steady-state distribution. Theorem 3.4 provides an upper bound for the number of iterations sufficient to ensure with certain probability that a GA has reached the set of globally optimal solutions and continues to include in each subsequent population at least one globally optimal solution whose observed fitness value is greater than that of any suboptimal solution. We describe the details of these results in Section 3.3.

Although many of the results for the multiplicative noise case are analogous to those for the additive noise case, there is one essential difference between the two cases: Although GAs eventually find at least one globally optimal solution with probability 1 in the additive case, they may *eventually fail* with probability 1 in the multiplicative case (see Definition 4.2.1). In Theorem 4.1, we establish a condition that is both necessary and sufficient for GAs in the multiplicative case to eventually find at least one globally optimal solution with probability 1. Furthermore, in Theorem 4.2, we identify a condition (Condition 4.2.2) that is both necessary and sufficient for GAs to eventually with probability 1 *fail* to find any globally optimal solution.

Space limitations on this paper force us to omit the proofs of these theorems; we will provide them in our full-length paper.

2. PRELIMINARIES

We assume that, as in traditional GAs, the search space S consists of 2^L binary strings of length L. These 2^L candidate solutions are also referred to as chromosomes. Since the search space is finite, these chromosomes will be labeled by integers $1, \ldots, 2^L$. Let f denote the (noiseless) fitness function. The objective of GAs is to find $i \in S$ such that

$$f(i) \geq f(j) \quad \forall\, j \in S.$$

In order to rigorously explain our Markov chain analysis, we describe the basic steps involved in executing GAs applied to noisy environments. The implementation of GAs considered in this study consists of the following steps:

1. An initial population of M chromosomes is formed, and the fitness of each of the chromosomes is evaluated.

2. A chromosome with the highest observed fitness value is selected, and this chromosome is included in the next population by the elitist strategy.

3. Selection is performed on the chromosomes in the current population, and $\frac{M-1}{2}$ pairs of chromosomes are formed (M is assumed to be odd).

4. Crossover is performed on each of the pairs to generate offspring.

5. Mutation is performed on each offspring, and this completes the formation of the next population of size M.

6. The fitness of each of the new chromosomes is evaluated. If this population satisfies some stopping criterion, then the algorithm terminates. Otherwise, steps 2–6 are repeated.

We consider executing GAs with the elitist strategy. This guarantees that the best candidate solution in the current population is included in the next population. In noisy environments, the best candidate solution in each population is a chromosome with the highest *observed* fitness value. Hence it is important to realize that the elitist strategy guarantees the monotonic improvement of (noisy) *observed* fitness but may fail to monotonically improve (noiseless) fitness.

When GAs terminate, note that they select a chromosome that has the highest *observed* fitness value among the chromosomes in the last population as a (candidate for a) globally optimal solution. Since the observed fitness value may not be the same as the (noiseless) fitness value of the chromosome due to noise, GAs may not choose a globally optimal solution even if it is included in the last population; in order for GAs to correctly identify a globally optimal solution contained in the last population, the chromosome must have the highest observed fitness value. This observation is important for properly characterizing GAs in noisy environments.

Another important strategy is that in step 6, GAs reevaluate the fitness value of each population member except for that of the chromosome preserved by the elitist strategy every time a new population is formed. It will become clear that this reevaluation of the fitness value is also essential for ensuring that GAs eventually find at least one globally optimal solution.

As stated earlier, we will not consider executing GAs with noise compensation techniques such averaging in this study, but note that these techniques may be totally ineffective for multiplicative noise. In Section 3.1, we specify the details of the genetic operations described in this section. We construct a Markov chain that models these GAs and derive its transition probabilities in Section 3.2.

3. GENETIC ALGORITHMS WITH ADDITIVE NOISE

We will first analyze GAs applied to fitness functions perturbed by bounded additive discrete noise. This case turns out to be much simpler than the multiplicative noise case, where fitness functions are perturbed by bounded multiplicative discrete noise (see Section 4). However, the analytical techniques applied to the additive case can essentially be carried over to the multiplicative case.

Although we attempt to closely follow the notation developed by previous studies that conducted Markov chain analysis of GAs (for example, see Vose and Liepins [19], Nix and Vose [13], Davis and Principe [4], Rudolph [16], Suzuki [17], Vose [18]), our notation inevitably becomes complicated due to the inclusion of noise. However, all of our new notation extends the conventional notation rather naturally. (The majority of the notation used in this study simply duplicates the notation firmly established by previous studies; see Suzuki [17] and Vose [18] in particular.) We will use simple examples to explain our notation.

3.1 Mathematical Details of Additive Noise and Genetic Operations

First, we select M chromosomes from S *with replacement* to form an initial population $\mathcal{P}_0$. The population generated during the k-th iteration (the k-th population) will be denoted by $\mathcal{P}_k$. Let $m(i, \mathcal{P}_k)$ denote the number of instances of chromosome i included in the k-th population $\mathcal{P}_k$. Note that $\mathcal{P}_k$ is a multiset of chromosomes for each k. Let $i(j, \mathcal{P}_k)$ represent the j-th instance of chromosome i in $\mathcal{P}_k$ (thus $1 \leq j \leq m(i, \mathcal{P}_k)$). We need this notation because it is necessary to distinguish all the elements in the multiset $\mathcal{P}_k$; all the elements must be distinguished in order to precisely characterize the mathematical properties of noise considered in this study and to define the states of the Markov chain we construct in Section 3.2.

Example 3.1.1. Consider a search space $S = \{1, 2, 3, 4\}$ (thus $L = 2$ for this search space). Suppose that the k-th population $\mathcal{P}_k$ consists of two instances of chromosome 1, one instance of chromosome 2, and two instances of chromosome 4 (hence $M = |\mathcal{P}_k| = 5$). Then

$$\mathcal{P}_k = \{1, 1, 2, 4, 4\}.$$

When the two instances of chromosome 1 in $\mathcal{P}_k$ must be distinguished, the first instance of 1 will be denoted by $1(1, \mathcal{P}_k)$, and the second instance of 1 will be denoted by $1(2, \mathcal{P}_k)$. When it is necessary to distinguish all the elements in $\mathcal{P}_k$, we write

$$\mathcal{P}_k = \{1(1, \mathcal{P}_k), 1(2, \mathcal{P}_k), 2(1, \mathcal{P}_k), 4(1, \mathcal{P}_k), 4(2, \mathcal{P}_k)\}. \quad (3.1)$$

We have $m(1, \mathcal{P}_k) = 2$ since there are two instances of chromosome 1 in this population. We also have $m(2, \mathcal{P}_k) = 1$, $m(3, \mathcal{P}_k) = 0$, and $m(4, \mathcal{P}_k) = 2$. We will refer to this simple example several times to explain our notation.

At each iteration, we evaluate the fitness value of each chromosome in the population. Instead of deterministic fitness values, we consider fitness values consisting of both deterministic and random components; we suppose that a random noise is added upon each fitness function evaluation. Hence if we let $F(i(j, \mathcal{P}_k))$ denote the observed fitness value of chromosome $i(j, \mathcal{P}_k)$ (the j-th instance of chromosome i in $\mathcal{P}_k$), then it can be written as

$$F(i(j, \mathcal{P}_k)) = f(i) + X_{i(j,\mathcal{P}_k)}, \quad (3.2)$$

where $X_{i(j,\mathcal{P}_k)}$ represents the random noise added to the fitness value of $i(j, \mathcal{P}_k)$. Note that $f(i(j, \mathcal{P}_k)) = f(i)$ for each i, j, and $\mathcal{P}_k$ because this is the deterministic component of the observed fitness value of chromosome i.

Example 3.1.2. Again, we consider the search space S and the k-th population $\mathcal{P}_k$ described in Example 3.1.1 [see (3.1)]. The observed fitness values of the five chromosomes in $\mathcal{P}_k$ will be expressed as follows:

$$\begin{aligned}
F(1(1, \mathcal{P}_k)) &= f(1) + X_{1(1,\mathcal{P}_k)}, \\
F(1(2, \mathcal{P}_k)) &= f(1) + X_{1(2,\mathcal{P}_k)}, \\
F(2(1, \mathcal{P}_k)) &= f(2) + X_{2(1,\mathcal{P}_k)}, \\
F(4(1, \mathcal{P}_k)) &= f(4) + X_{4(1,\mathcal{P}_k)}, \\
F(4(2, \mathcal{P}_k)) &= f(4) + X_{4(2,\mathcal{P}_k)}.
\end{aligned}$$

For instance, the deterministic component of the observed fitness value of $1(1, \mathcal{P}_k)$ is $f(1)$, which is the same as that of $1(2, \mathcal{P}_k)$. Similarly, the deterministic component of the observed fitness value of $4(1, \mathcal{P}_k)$ is $f(4)$, which is the same as that of $4(2, \mathcal{P}_k)$.

For analytical simplicity, we assume that $X_{i(j,\mathcal{P}_k)}$ are independent and identically distributed; the random component has the same distribution for each observed fitness value, and it is independent of the random component added to any other instance of fitness function evaluation. We further assume that $X_{i(j,\mathcal{P}_k)}$ is discrete and takes on finitely many values. Thus for each $i(j,\mathcal{P}_k)$, we define $X_{i(j,\mathcal{P}_k)}$ by

$$X_{i(j,\mathcal{P}_k)} = x_n \text{ with probability } p_n, \quad 1 \le n \le N, \tag{3.3}$$

where N represents the number of distinct possible values of $X_{i(j,\mathcal{P}_k)}$ (thus $N < \infty$).

Let $i^*(j^*,\mathcal{P}_k)$ represent an instance of a chromosome in $\mathcal{P}_k$ that has the highest *observed* fitness value in the population $\mathcal{P}_k$ (it is the j^*-th instance of chromosome i^* in $\mathcal{P}_k$). The elitist strategy guarantees the inclusion of $i^*(j^*,\mathcal{P}_k)$ in the next population $\mathcal{P}_{k+1}$. (If the highest observed fitness value is achieved by more than one $i(j,\mathcal{P}_k) \in \mathcal{P}_k$, then break a tie by selecting one of them uniformly at random to determine $i^*(j^*,\mathcal{P}_k)$).

Using the notation defined above, we describe the mathematical details of selection. At each iteration, selection is performed to form pairs of chromosomes. In this study, we consider forming $\frac{M-1}{2}$ pairs for concreteness (thus M is assumed to be odd). First, the fitness value of each chromosome in the current population $\mathcal{P}_k$ is evaluated. Each of the observed fitness values has the form shown in (3.2). GAs subsequently select $M-1$ chromosomes from $\mathcal{P}_k$ *with replacement* to form these pairs. Our Markov chain analysis of GAs in the noisy environment is valid for any selection scheme—for example, proportional selection, ranking selection, and tournament selection (e.g., Miller and Goldberg [11], Vose [18], Leung, Duan, Xu, and Wong [10]).

Crossover and mutation in the noisy case do not differ from those in the noiseless case. For each of the $\frac{M-1}{2}$ pairs formed by selection, crossover is performed with some predetermined probability in order to generate two new chromosomes from the pair. Due to the elitist strategy, crossover does not operate on the chromosome $i^*(j^*,\mathcal{P}_k)$. Similarly, the elitist strategy does not allow mutation to alter $i^*(j^*,\mathcal{P}_k)$. For each of the other chromosomes in $\mathcal{P}_k$, mutation inverts each bit of an individual chromosome with some predetermined probability μ. We assume $0 < \mu < 1$. Although adaptive mutation rates are not considered in this study, we are currently investigating how to optimize the mutation rate using Markov chain analysis.

A new population $\mathcal{P}_{k+1}$ emerges upon completing selection, crossover, and mutation. The algorithm computes the fitness value of each chromosome in $\mathcal{P}_{k+1}$, and these steps are repeated until a stopping criterion is satisfied.

3.2 Framework of Markov Chain Analysis

The key to understanding our Markov chain analysis of GAs in noisy environments is that we construct a Markov chain, call it (Z_k), whose state space consists of multisets not of chromosomes but of the *ordered pairs* defined below. The state space of a Markov chain constructed to model GAs in the noiseless case is typically the set of all possible distinct populations that can be formed from S (for example, see Davis and Principe [4], Suzuki [17], Vose [18]). However, this Markov chain fails to explicitly capture the evolution of GAs in noisy environments.

For the k-th iteration of a GA in the additively noisy environment, the corresponding state of the chain can be derived from the k-th population $\mathcal{P}_k$ as follows. We pair each chromosome $i(j,\mathcal{P}_k)$ in $\mathcal{P}_k$ with the value of the additive noise observed when its fitness value is evaluated [see (3.2)] and form an ordered pair

$$(i, X_{i(j,\mathcal{P}_k)}), \tag{3.4}$$

where j and $\mathcal{P}_k$ are suppressed in the first entry because they are unnecessary. In this manner, we form M ordered pairs from the M chromosomes in $\mathcal{P}_k$. These M ordered pairs compose a state of the Markov chain for the noisy case (thus each state of this chain is a multiset of the ordered pairs).

Example 3.2.1. Consider the search space S and the population $\mathcal{P}_k$ described in Examples 3.1.1–3.1.3. We have $\mathcal{P}_k = \{1, 1, 2, 4, 4\}$. This population leads to the following state $\mathcal{T}$ of the Markov chain for the noisy environment:

$$\begin{aligned}\mathcal{T} = \{&(1, X_{1(1,\mathcal{P}_k)}), (1, X_{1(2,\mathcal{P}_k)}), (2, X_{2(1,\mathcal{P}_k)}),\\ &(4, X_{4(1,\mathcal{P}_k)}), (4, X_{4(2,\mathcal{P}_k)})\}.\end{aligned}$$

Each state of this particular Markov chain consists of five ordered pairs.

It is important to recognize that in order to explicitly model the evolution of GAs in the noisy environment using any Markov chain, we need the second entry of each ordered pair in (3.4) (or a quantity that is mathematically equivalent to the second entry) because the selection process of GAs applied to the noisy environment is based on a function of not only the chromosome but also the i.i.d. noise defined at (3.3). We analyze the Markov chain (Z_k) to uncover the transition and convergence properties of GAs in the noisy environment. We denote by $\mathfrak{T}$ the state space of (Z_k). Let $m(i,\mathcal{T})$ denote the number of instances of chromosome i included in the ordered pairs of $\mathcal{T} \in \mathfrak{T}$ (thus $m(i,\mathcal{T})$ is analogous to $m(i,\mathcal{P}_k)$ defined at the beginning of Section 3.1). Similarly, we denote by $m(x_n,\mathcal{T})$ the number of instances of noise value x_n contained in the ordered pairs of $\mathcal{T} \in \mathfrak{T}$. Note that

$$\sum_{i=1}^{2^L} m(i,\mathcal{T}) = \sum_{n=1}^{N} m(x_n,\mathcal{T}) = M$$

for each $\mathcal{T} \in \mathfrak{T}$.

Example 3.2.2. Consider the following state $\mathcal{T} \in \mathfrak{T}$:

$$\mathcal{T} = \{(1, x_2), (1, x_3), (2, x_1), (4, x_3), (4, x_3)\}.$$

As described earlier, the first entry in each ordered pair is a chromosome, and the second entry is the noise value observed when the fitness of the chromosome is evaluated. We have $m(1,\mathcal{T}) = 2$ since two instances of chromosome 1 are contained in the ordered pairs of $\mathcal{T}$. Similarly, we have $m(2,\mathcal{T}) = 1$ and $m(4,\mathcal{T}) = 2$. We also have $m(x_1,\mathcal{T}) = 1$, $m(x_2,\mathcal{T}) = 1$, and $m(x_3,\mathcal{T}) = 3$.

We are now ready to describe the transition probabilities of the Markov chain (Z_k). In Section 3.3, we will show that these probabilities can be effectively used to bound the number of iterations sufficient to ensure with certain probability that a GA selects a globally optimal solution upon termination. The following theorem shows the exact transition probabilities of (Z_k).

Theorem 3.1. *Let* (Z_k) *denote the Markov chain with state space* $\mathfrak{T}$ *that models GAs in the noisy environment. Let* $\mathcal{T}$ *and* $\mathcal{T}'$ *denote states in* $\mathfrak{T}$*, and let* $i^*(\mathcal{T})$ *denote a chromosome in (an ordered pair of)* $\mathcal{T} \in \mathfrak{T}$ *that has the highest observed fitness value. If the observed fitness value of* $i^*(\mathcal{T}')$ *is greater than or equal to that of* $i^*(\mathcal{T})$*, then for each* k*,*

$$\begin{aligned} & P\{Z_{k+1} = \mathcal{T}' | Z_k = \mathcal{T}\} \\ &= (M-1)! \prod_{i=1}^{2^L} \frac{1}{\tilde{m}(i,\mathcal{T}')!} \phi(i,\mathcal{T})^{\tilde{m}(i,\mathcal{T}')} \\ &\qquad \times (M-1)! \prod_{n=1}^{N} \frac{1}{\tilde{m}(x_n,\mathcal{T}')!} p_n^{\tilde{m}(x_n,\mathcal{T}')}, \end{aligned} \tag{3.5}$$

where

$$\phi(i,\mathcal{T}) = P\{\textit{chromosome } i \textit{ is generated from state } \mathcal{T}\}, \tag{3.6}$$

$$\tilde{m}(i,\mathcal{T}') = \begin{cases} m(i,\mathcal{T}') - 1 & \textit{if } i = i^*(\mathcal{T}) \\ m(i,\mathcal{T}') & \textit{otherwise,} \end{cases}$$

and

$$\tilde{m}(x_n,\mathcal{T}') = \begin{cases} m(x_n,\mathcal{T}') - 1 & \textit{if } i^*(\mathcal{T}) \textit{ is paired with } x_n \\ & \textit{in } \mathcal{T} \textit{ (and in } \mathcal{T}'\textit{)} \\ m(x_n,\mathcal{T}') & \textit{otherwise.} \end{cases}$$

On the other hand, if the observed fitness value of $i^*(\mathcal{T}')$ *is less than that of* $i^*(\mathcal{T})$*, then*

$$P\{Z_{k+1} = \mathcal{T}' | Z_k = \mathcal{T}\} = 0.$$

The probabilities $\phi(i,\mathcal{T})$ defined at (3.6) depend on the selection scheme employed by GAs. They can be computed exactly.

3.3 Convergence Analysis

Using the Markov chain (Z_k) constructed in Section 3.2, we analyze the convergence properties of GAs applied to fitness functions perturbed by the additive discrete noise described in Section 3.1. Without loss of generality, we arrange the N labels $x_1, x_2, \ldots, x_N$ representing the N possible values of $X_{i(j,\mathcal{P}_k)}$ defined at (3.3) in descending order:

$$x_1 > x_2 > \cdots > x_N. \tag{3.7}$$

This is simply for notational convenience. Let S^* denote the set of chromosomes that are globally optimal solutions. We have

$$f(i) \geq f(j) \quad \forall\, i \in S^*,\ \forall\, j \in S. \tag{3.8}$$

The following theorem guarantees that the GAs applied to the noisy environment eventually find at least one globally optimal solution with probability 1.

Theorem 3.2. *The Markov chain* (Z_k) *is indecomposable: It has only one positive recurrent communication class, which consists of states in* $\mathfrak{T}$ *that each contain at least one ordered pair* (i, x_1) *with* $i \in S^*$*.*

Let $\mathfrak{A}$ denote the only positive recurrent communication class of the chain. Thus the chain hits $\mathfrak{A}$ and stays there with probability 1. Since each state in $\mathfrak{A}$ contains at least one globally optimal solution paired with a noise value that maximizes its observed fitness value, Theorem 3.2 immediately implies the following essential property of GAs in the additive noise case: With probability 1, the algorithms eventually (i.e., as the number of iterations goes to infinity) reach the set of globally optimal solutions and continue to include in each subsequent population at least one globally optimal solution whose *observed* fitness value is greater than that of any suboptimal solution. It is important that the globally optimal solution has the highest *observed* fitness value because GAs will otherwise fail to select it as a (candidate for a) globally optimal solution when they terminate. Thus GAs eventually find at least one optimal solution with probability 1.

The next theorem follows from Theorem 3.2 and ensures the convergence of the chain to stationarity.

Theorem 3.3. *The Markov chain* (Z_k) *has a unique stationary distribution that is also its steady-state distribution: There exists a unique distribution* π *on* $\mathfrak{T}$ *such that*

$$\pi = \pi K,$$

where K *is the* $|\mathfrak{T}| \times |\mathfrak{T}|$ *transition kernel of* (Z_k)*, and for any states* $\mathcal{T}$ *and* $\mathcal{T}'$ *in* $\mathfrak{T}$*,*

$$\pi(\mathcal{T}) = \lim_{k\to\infty} P\{Z_k = \mathcal{T} | Z_0 = \mathcal{T}'\}.$$

The stationary distribution π in Theorem 3.3 satisfies

$$\pi(\mathcal{T}) > 0 \quad \forall\, \mathcal{T} \in \mathfrak{A}, \tag{3.9}$$

and

$$\pi(\mathcal{T}) = 0 \quad \forall\, \mathcal{T} \in \mathfrak{T} \setminus \mathfrak{A}. \tag{3.10}$$

Thus the number of nonzero entries in π equals $|\mathfrak{A}|$. Let $\pi^{(k)}$ denote the distribution of the chain (Z_k) at time k. From Theorem 3.3 and (3.9)–(3.10), we have

$$\lim_{k\to\infty} \sum_{\mathcal{T}\in\mathfrak{A}} \pi^{(k)}(\mathcal{T}) = 1, \tag{3.11}$$

Thus, in order to determine how many iterations are sufficient to guarantee with certain probability that GAs select a globally optimal solution upon termination, we need to analyze the convergence rate of (3.11). Clearly, (3.11) is equivalent to

$$\lim_{k\to\infty} \sum_{\mathcal{T}\in\mathfrak{T}\setminus\mathfrak{A}} \pi^{(k)}(\mathcal{T}) = 0, \tag{3.12}$$

and the remaining part of this section focuses on the convergence rate analysis of (3.12).

In order to analyze the convergence rate, we need to establish more notation. Each ordered pair (i, x_n) can be associated with its observed fitness value, which we denote by $F((i, x_n))$:

$$F((i, x_n)) := f(i) + x_n. \tag{3.13}$$

Let W denote the number of distinct possible values of (3.13), and let $F_1, F_2, \ldots, F_W$ denote the W distinct values. For notational convenience, they will be arranged in descending order:

$$F_1 > F_2 > \cdots > F_W.$$

Let

$$H_j := \{\mathcal{T} \in \mathfrak{T} | \max_{(i,x_n)\in\mathcal{T}} F((i,x_n)) = F_j\}.$$

Thus, the highest observed fitness value of each state in H_j equals F_j. Note that H_1 denotes the set of states in $\mathfrak{T}$ that contain at least one ordered pair (i, x_1) with $i \in S^*$; hence $H_1 = \mathfrak{A}$, and we have $\mathfrak{T} \setminus \mathfrak{A} = \bigcup_{j=2}^{W} H_j$. For the remaining analysis, we rewrite (3.12) as

$$\lim_{k\to\infty} \sum_{\mathcal{T}\in\bigcup_{j=2}^{W} H_j} \pi^{(k)}(\mathcal{T}) = 0. \tag{3.14}$$

We assign $|\mathfrak{T}|$ labels $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_{|\mathfrak{T}|}$ to the $|\mathfrak{T}|$ states in $\mathfrak{T}$ as follows. The first $|H_1|$ labels $(\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_{|H_1|})$ represent states that belong to H_1 (it does not matter exactly how these $|H_1|$ states are represented by the $|H_1|$ labels). The next $|H_2|$ labels $(\mathcal{T}_{|H_1|+1}, \mathcal{T}_{|H_1|+2}, \ldots, \mathcal{T}_{|H_1|+|H_2|})$ represent states that belong to H_2. We continue this process until we label all the states in $\mathfrak{T}$. With these labels, the i-j-th entry of the $|\mathfrak{T}| \times |\mathfrak{T}|$ transition kernel $K$of the Markov chain (Z_k) naturally represents the one-step transition probability from state $\mathcal{T}_i$ to state $\mathcal{T}_j$. As shown in Figure 1, K is a block lower triangular matrix, and the j-th diagonal block $K(j)$ is a $|H_j| \times |H_j|$ matrix $(1 \le j \le W)$. The eigenvalues of this transition kernel K provide bounds for the convergence rate of (3.14). Let $\sigma(K)$ denote the spectrum of the kernel. Since K is a block lower triangluar matrix, the eigenvalues of K are the eigenvalues of the W diagonal blocks $K(1)$, $K(2)$,..., $K(W)$:

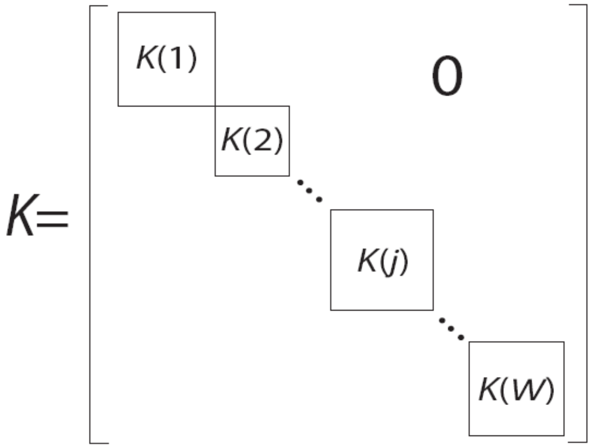

Figure 1: Transition Kernel of (Z_k). The $|\mathfrak{T}| \times |\mathfrak{T}|$ transition matrix K of the Markov chain (Z_k) is a block lower triangular matrix whose j-th diagonal block $K(j)$ is a $|H_j| \times |H_j|$ matrix $(1 \le j \le W)$.

$$\sigma(K) = \bigcup_{j=1}^{W} \sigma(K(j)). \tag{3.15}$$

Let $\lambda_{j,l}$ denote the eigenvalues of the j-th diagonal block $K(j)$:

$$\sigma(K(j)) = \{\lambda_{j,1}, \lambda_{j,2}, \ldots, \lambda_{j,|H_j|}\}. \tag{3.16}$$

As stated in Theorems 3.2 and 3.3, the Markov chain (Z_k) has only one positive recurrent class, which is also aperiodic. Hence the eigenvalue 1 of the transition kernel K has multiplicity 1, and it belongs to $\sigma(K(1))$. Moreover, there are no other eigenvalues of modulus 1:

$$|\lambda| < 1 \quad \forall\, \lambda \in \sigma(K),\ \lambda \neq 1. \tag{3.17}$$

This is shown in Karlin and Taylor [9] (see Chapter 10).

We have $\pi^{(k)} = \pi^{(0)} K^k$, where $\pi^{(0)}$ is the initial distribution of the chain. Thus, the convergence rate of (3.14) can be analyzed by examining the $(|\mathfrak{T}| - |H_1|) \times (|\mathfrak{T}| - |H_1|)$ submatrix of K that is obtained by eliminating the first $|H_1|$ rows and the first $|H_1|$ columns of K. This submatrix will be denoted by $\tilde{K}$. From (3.15)–(3.17), we know that each entry of $\tilde{K}^k$ goes to zero as k approaches infinity (the modulus of each eigenvalue of $\tilde{K}$ is strictly less than 1), and the convergence rate of (3.14) is determined by how fast each entry of $\tilde{K}^k$ goes to zero. We have the following theorem:

THEOREM 3.4. *There exists a constant $C < \infty$ such that for each k,*

$$\sum_{\mathcal{T}\in\bigcup_{j=2}^{W} H_j} \pi^{(k)}(\mathcal{T}) \le C\lambda^{*k},$$

where $\lambda^* = \max\{|\lambda| : \lambda \in \bigcup_{j=2}^{W} \sigma(K(j))\}$.

Note that $\lambda^* < 1$. With regard to (3.11), it follows from Theorem 3.4 that

$$\sum_{\mathcal{T}\in\mathfrak{A}} \pi^{(k)}(\mathcal{T}) \ge 1 - C\lambda^{*k}.$$

Thus the convergence rate of the probability in (3.11) or (3.12) is roughly determined by the maximum modulus λ^* of eigenvalues in the spectrum of the $(|\mathfrak{T}|-|H_1|)\times(|\mathfrak{T}|-|H_1|)$ submatrix $\tilde{K}$ of the transition kernel K.

4. GENETIC ALGORITHMS WITH MULTIPLICATIVE NOISE

In Sections 4.1–4.2, we analyze GAs applied to fitness functions perturbed by multiplicative noise that takes on finitely many values. We will continue to use the notation established in Sections 2 and 3. The Markov chain construction described in Section 3.2 can be carried over to the multiplicative noise case with few modifications. However, the convergence analysis for the multiplicative case becomes more complicated compared to the additive case described in Section 3.3. First we characterize the multiplicative noise mathematically.

4.1 Mathematical Details of Multiplicative Noise

In the multiplicative noise case, we suppose that random noise multiplies the deterministic component of the fitness value upon each fitness function evaluation. Hence the observed fitness value $F(i(j, \mathcal{P}_k))$ of chromosome $i(j, \mathcal{P}_k)$ can be written as

$$F(i(j, \mathcal{P}_k)) = X_{i(j,\mathcal{P}_k)} f(i), \tag{4.1}$$

where $X_{i(j,\mathcal{P}_k)}$ represents the random multiplicative noise that perturbs the fitness value of $i(j, \mathcal{P}_k)$. Here we do not assume that the multiplicative noise $X_{i(j,\mathcal{P}_k)}$ is positive with probability 1, so we cannot simply take the logarithm of (4.1). For analytical simplicity, we again assume that $X_{i(j,\mathcal{P}_k)}$

are independent and identically distributed and that each $X_{i(j,\mathcal{P}_k)}$ is discrete and takes on finitely many values. Thus for each $i(j,\mathcal{P}_k)$, we again define $X_{i(j,\mathcal{P}_k)}$ by (3.3). We will denote by $\mathfrak{X}$ the set of the N possible values of the multiplicative noise $X_{i(j,\mathcal{P}_k)}$:

$$\mathfrak{X} := \{x_1, x_2, \ldots, x_N\}. \tag{4.2}$$

4.2 Convergence Analysis

The construction of the Markov chain described in Section 3.2 remains essentially the same for the multiplicative noise case. Each state of this chain is again a multiset of the ordered pairs defined at (3.4), and Theorem 3.1 holds for the multiplicative noise case as well. We leave the details to the reader.

The following is an essential condition for characterizing convergence properties of GAs in the multiplicatively noisy environment:

Condition 4.2.1. The maximum observed fitness value of globally optimal solutions is greater than that of any suboptimal solution, i.e.,

$$\max_{1 \le n \le N} \{x_n f(i)\} > \max_{1 \le n \le N} \{x_n f(j)\} \tag{4.3}$$
$$\forall\ i \in S^*,\ \forall\ j \in S \setminus S^*.$$

Here define the right-hand side of (4.3) to be $-\infty$ for the trivial case that $S^* = S$, i.e., any chromosome in S is a globally optimal solution. Note that since S^* denotes the set of globally optimal solutions, all the chromosomes in S^* have the same maximum (noiseless) fitness value [see (3.8)]. Thus Condition 4.2.1 can also be stated as follows: There exists a chromosome $i \in S^*$ such that

$$\max_{1 \le n \le N} \{x_n f(i)\} > \max_{1 \le n \le N} \{x_n f(j)\} \quad \forall\ j \in S \setminus S^*.$$

Example 4.2.1. Suppose that the (noiseless) fitness value of each chromosome is positive:

$$f(i) > 0 \quad \forall\ i \in S.$$

Then, excluding the trivial case that $S^* = S$, Condition 4.2.1 is satisfied if and only if $X_{i(j,\mathcal{P})}$ defined at (3.3) takes on at least one positive value with positive probability, i.e., there exists $x \in \mathfrak{X}$ such that $x > 0$ [$\mathfrak{X}$ is defined at (4.2)].

Example 4.2.2. Suppose that there exists at least one chromosome i in S whose fitness value $f(i)$ is positive. Then Condition 4.2.1 is satisfied if, for instance, $X_{i(j,\mathcal{P})}$ is positive with probability 1:

$$x > 0 \quad \forall\ x \in \mathfrak{X}.$$

The following theorem shows a striking difference between the additive and multiplicative cases.

Theorem 4.1. *Suppose that GAs are executed in the multiplicatively noisy environment described in Section 4.1. Then Condition 4.2.1 is necessary and sufficient for GAs to eventually find at least one globally optimal solution with probability 1.*

This theorem shows that we can verify Condition 4.2.1 in order to determine whether GAs in the multiplicative noise case will eventually find at least one globally optimal solution with probability 1. Examining the proof of this theorem, we can also identify a condition that is both necessary and sufficient for GAs to *eventually fail* with probability 1. First we define the eventual failure mathematically.

Definition 4.2.1 (Eventual Failure). GAs are said to *eventually fail* if, after some (random) finite number of iterations, they include in each subsequent population at least one suboptimal solution whose observed fitness value is greater than that of any globally optimal solution. In other words, GAs are said to eventually fail if, after sufficiently many iterations, they are guaranteed to never find any globally optimal solution. This may be counterintuitive because the probability of the undesirable event of failing to find any globally optimal solution approaches 1 as the number of iterations goes to infinity.

The following is another essential condition for characterizing the convergence properties of GAs in the multiplicatively noisy environment:

Condition 4.2.2. There exists a suboptimal solution such that its maximum observed fitness value is greater than that of globally optimal solutions, i.e., there exists a chromosome i in $S \setminus S^*$ such that

$$\max_{1 \le n \le N} \{x_n f(i)\} > \max_{1 \le n \le N} \{x_n f(j)\} \quad \forall\ j \in S^*. \tag{4.4}$$

Example 4.2.3. Suppose that the (noiseless) fitness value of each chromosome is positive:

$$f(i) > 0 \quad \forall\ i \in S.$$

Then, excluding the trivial case that $S^* = S$, Condition 4.2.2 is satisfied if and only if $X_{i(j,\mathcal{P})}$ defined at (3.3) is negative with probability 1: $x < 0 \quad \forall\ x \in \mathfrak{X}$ [$\mathfrak{X}$ is defined at (4.2)].

We are now in position to state the following theorem:

Theorem 4.2. *Suppose that GAs are executed in the multiplicatively noisy environment described in Section 4.1. Then Condition 4.2.2 is necessary and sufficient for GAs to eventually fail with probability 1.*

Theorems 4.1–4.2 reveal an intriguing difference between the additive and multiplicative cases; although GAs eventually find at least one globally optimal solution with probability 1 in the additive case, they may eventually fail with probability 1 in the multiplicative case.

One can show that the chain for the multiplicative noise case converges to stationarity regardless of whether Condition 4.2.1 (or Condition 4.2.2) holds. Thus we can obtain a result analogous to Theorem 3.4 for the multiplicative case. We describe the details of this analysis in our full-length paper.

5. DISCUSSION

To our knowledge, this study is the first to rigorously examine transition and convergence properties of GAs applied to noisy environments by explicitly constructing a Markov model. Mathematically, a Markov chain is completely determined by its transition kernel, and we thoroughly described the kernel of the chain that models GAs in noisy environments. Our analysis shows that GAs eventually (i.e., as the number of iterations goes to infinity) find at least one globally optimal solution with probability 1 when they are applied to fitness functions perturbed by any i.i.d. additive noise that takes on finitely many values. For the multiplicative noise case, however, they may be unable to do so depending on whether Condition 4.2.1 is satisfied; in fact, they eventually fail (see Definition 4.2.1) with probability 1 if Condition 4.2.2 is satisfied.

This striking difference between the additive and multiplicative cases probably has significant implications in practice. For example, one must be cautious about using genetic algorithms to find a globally optimal solution if it is suspected that the fitness function is perturbed by multiplicative noise. On the other hand, it is reassuring that GAs are guaranteed to find at least one globally optimal solution as the number of iterations goes to infinity if the fitness function is perturbed by the additive noise described in Section 3.1.

Note that GAs considered in our study are assumed to reassess the fitness value of each population member except for that of the chromosome preserved by the elitist strategy every time a new population is formed. If the algorithms do not do this, then, examining the arguments described in Sections 3.2–3.3, it is easy to see that the probability that they eventually find at least one globally optimal solution is less than 1 even for the additive noise case. Thus the reevaluation of the fitness value is important for ensuring that GAs accomplish their objective. Moreover, it significantly reduces memory requirements as compared to the strategy of storing the fitness value of every chromosome that has been included in any population formed during the runtime.

Theorem 3.4 shows that the convergence rate of the probability in (3.11) or (3.12) is roughly determined by the maximum modulus λ^* of the eigenvalues of the $(|\mathfrak{T}|-|H_1|)\times(|\mathfrak{T}|-|H_1|)$ submatrix $\tilde{K}$ of the transition kernel K described in Section 3.3. Both λ^* and C can be explicitly computed from the Jordan canonical form of $\tilde{K}$. Hence the transition probabilities in Theorem 3.1 can be effectively used to bound the number of iterations sufficient to ensure with certain probability that a GA selects a globally optimal solution upon termination.

We are currently extending our Markov chain analysis to other noisy environments—for example, environments with more general discrete noise and with continuous noise. We believe that our Markov chain-theoretic approach to analyzing GAs in noisy environments will further elucidate essential theoretical and practical properties of the algorithms.

6. REFERENCES

[1] D. V. Arnold. *Noisy Optimization with Evolution Strategies.* Kluwer Academic Publishers, Boston, 2002.

[2] H. G. Beyer. Evolutionary algorithms in noisy environments: theoretical issues and guidelines for practice. *Computer Methods in Applied Mechanics and Engineering*, 186:239–267, 2000.

[3] A. Chen, K. Subprasom, and Z. Ji. A simulation-based multi-objective genetic algorithm (SMOGA) procedure for BOT network design problem. *Optimization and Engineering*, 7:225–247, 2006.

[4] T. E. Davis and J. Principe. A Markov chain framework for the simple genetic algorithm. *Evolutionary Computation*, 1:269–288, 1993.

[5] A. Di Pietro, L. White, and L. Barone. Applying evolutionary algorithms to problems with noisy, time-consuming fitness functions. In *Proceedings of the 2004 Congress on Evolutionary Computation*, volume 2, pages 1254– 1261, 2004.

[6] D. E. Goldberg and M. W. Rudnick. Genetic algorithms and the variance of fitness. *Complex Systems*, 5:265–278, 1991.

[7] W. E. Hopkins. Optimal stabilization of families of linear stochastic differential equations with jump coefficients and multiplicative noise. *SIAM Journal on Control and Optimization*, 25:1587–1600, 1987.

[8] Y. Jin and J. Branke. Evolutionary optimization in uncertain environments—a survey. *IEEE Transactions on Evolutionary Computation*, 3:303–317, 2005.

[9] S. Karlin and H. M. Taylor. *A Second Course in Stochastic Processes.* Academic Press, New York, 1981.

[10] K. S. Leung, Q. H. Duan, Z. B. Xu, and C. K. Wong. A new model of simulated evolutionary computation—convergence analysis and specifications. *IEEE Transactions on Evolutionary Computation*, 5:3–16, 2001.

[11] B. L. Miller and D. E. Goldberg. Genetic algorithms, selection schemes, and the varying effects of noise. *Evolutionary Computation*, 4:113–131, 1996.

[12] V. Nissen and J. Propach. On the robustness of population-based versus point-based optimization in the presence of noise. *IEEE Transactions on Evolutionary Computation*, 2:107–119, 1998.

[13] A. Nix and M. D. Vose. Modeling genetic algorithm with Markov chains. *Annals of Mathematics and Artificial Intelligence*, 5:27–34, 1992.

[14] J. A. Primbs. Portfolio optimization applications of stochastic receding horizon control. In *Proceedings of the 2007 American Control Conference*, volume 1, pages 1811–1816, 2007.

[15] L. I. Rudin and S. Osher. Total variation based image restoration with free local constraints. In *Proceedings of the 2004 International Conference on Image Processing*, volume 1, pages 31–35, 2004.

[16] G. Rudolph. Convergence analysis of canonical genetic algorithms. *IEEE Transactions on Neural Networks*, 5:96–101, 1994.

[17] J. Suzuki. A Markov chain analysis on simple genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 25:655–659, 1995.

[18] M. D. Vose. *The Simple Genetic Algorithm.* MIT Press, Cambridge, Massachusetts, 1999.

[19] M. D. Vose and G. E. Liepins. Punctuated equilibria in genetic search. *Complex Systems*, 5:31–44, 1991.