Gaussian Mixture Model of Evolutionary Algorithms

Bo Song
Dept. of Electrical and Electronic Engineering
The University of Hong Kong, Hong Kong
bsong@hku.hk

Victor O. K. Li
Dept. of Electrical and Electronic Engineering
The University of Hong Kong, Hong Kong
vli@eee.hku.hk

ABSTRACT

This paper proposes a novel finite Gaussian mixture model to study the population dynamics of evolutionary algorithms on continuous optimization problems. While previous research taking on a dynamical system view has established the transition equation between the density functions of consecutive populations, the equation usually does not have closed-form solutions and can only be applied to very few optimization problems. In this paper, we address this issue by approximating both the population density function of each generation and the objective function by finite Gaussian mixtures. We show that by making such approximations the transition equation can be solved exactly and key statistics, such as the expected mean and the variance of fitness values of the population, can be calculated easily. We also prove that by choosing appropriate values of the parameters, the L^1-norm error between our model and the actual population density function can be made arbitrarily small, up until a predefined generation. We present experimental results to show that our model is useful in simulating and examining the dynamics of evolutionary algorithms.

Categories and Subject Descriptors

F.2.1 [**ANALYSIS OF ALGORITHMS AND PROBLEM COMPLEXITY**]: Numerical Algorithms and Problems; I.2.8 [**ARTIFICIAL INTELLIGENCE**]: Problem Solving, Control Methods, and Search—*Heuristic methods*; G.1.6 [**NUMERICAL ANALYSIS**]: Optimization

General Terms

Algorithms; Theory

Keywords

Evolutionary Algorithm; Population Dynamics; Gaussian Mixture

GECCO'14, July 12–16, 2014, Vancouver, BC, Canada.

http://dx.doi.org/10.1145/2576768.2598252 .

1. INTRODUCTION

Despite the efficiency and effectiveness of evolutionary algorithms at solving difficult optimization problems, the understanding of the underlying evolutionary processes and the behavior of these algorithms remain incomplete. Among the many theoretical analyses of evolutionary algorithms, the dynamical systems modelling approach is a primary and influential one. The idea is to consider the state space of all possible populations, and the stochastic evolution of the populations from one generation to the next is characterized by the transition matrix of a Markov chain. Equivalently, as a distribution of the population states corresponds to a probability density function determining the probability that each individual occurs in the population (henceforth denominated as population density function), the evolution of the algorithm can also be characterized by a difference equation (transition equation) between population density functions of consecutive generations. The transition equation describes exactly how the population evolves as the evolutionary algorithm progresses. In order to obtain simple forms of the transition equation and derive analytical results, an assumption that the evolutionary algorithm has infinite population size is often made, and properties regarding the transient and asymptotic behavior of the evolutionary algorithms are analyzed. In addition, given the initial population density function, it is possible to simulate and visualize the trajectories of key statistics of the algorithm such as the expected average fitness value and the expected mean center of the population on certain optimization problems, by recursively applying the transition equation.

Research efforts adopting a dynamical system view have been fruitful. Here we mention the fundamental monographs of Vose [13], Beyer [1], and the survey of Reeves and Rowe [10]. Usually, the research starts with the most general forms of population density functions (or population vectors as used in [10, 13]) and objective (fitness) functions. Then, the transition equation is derived by modeling the effects of various operators on the population, such as that of mutation and fitness proportional selection. Though under the infinite population size assumption the transition equation often has simple forms and general properties can be derived from the transition matrix of the Markov chain, the problem with this approach is that given particular optimization problems, it is usually impossible to construct the transition matrix in practice, and to solve the transition equation analytically. In other words, the transition equation cannot be applied recursively to simulate the running of the evolutionary algorithms generation by generation.

To resolve this issue, one major approach is to introduce the framework of statistical mechanics [10, 11]. The idea is that instead of using *microscopic* descriptions of population dynamics to describe precisely what happens to the population density function after each iteration, use *macroscopic* descriptions where key statistics of interest such as the average fitness value of the population become the target to be modeled and analyzed. For example, if the average fitness value is taken as the key statistic to describe a population, the first step of a statistical mechanics analysis is likely to be modelling the probability distribution of average fitness values over all possible populations in each generation. Then, by truncating a representation of this distribution (such as moments or cumulants) to a certain number of terms, a transition equation of the preserved terms can be derived and readily solved. The benefit of this approach is that by characterizing population dynamics by a few statistics, the scale of the model is greatly reduced, thus simulation and visualization of the dynamics of the system are possible. Also, most of the time the key statistics in the model are closely related to the properties of interest. Therefore, this level of description seems sufficient. However, the drawback of this approach is that the key statistics do not contain all the information to describe a population and truncating transformations of probability distributions further discards information, additional approximation errors are introduced. Besides, in general, the transition equation of the preserved terms is usually difficult to derive, and other principles such as the principle of maximum entropy or other modelling assumptions are needed to derive it. In fact, only a few discrete optimization problems have been successfully analyzed by this approach [10].

In this paper, we propose a different approach. We are concerned mainly with continuous optimization problems, and the target model to be approximated is the general model introduced by [8,9]. This model studies the dynamics of simple evolutionary algorithms with infinite population size, and the transition equation of the model is a microscopic level description of the evolution of the population density functions. Instead of using statistical mechanics to simulate the evolution of the system, we propose a novel finite Gaussian mixture model to solve the transition equation approximately. The key to our model is that both the objective function and the population density functions are approximated by finite Gaussian mixtures. Due to the nice properties of Gaussian functions, such approximations guarantee that the transition equation can be solved analytically and the solution is also a Gaussian mixture. Moreover, key statistics of interest such as the expected mean and the variance of fitness values of the population can also be easily calculated from the population density functions of Gaussian mixtures. Regarding the accuracy of our method, we prove that our framework works for nearly arbitrary continuous objective functions, and given a predefined generation number k and a tolerance level ϵ, by choosing appropriate parameter values, the L^1-norm error of our method can be less than ϵ up until the kth generation. We simulate our model running on low dimensional multimodal optimization problems and visualize and compare the simulation results with results obtained from actually running the algorithm. The experimental results show that our method can predict the behavior of evolutionary algorithms with fair accuracy. To the best of our knowledge, our effort is the first in adopting approximation techniques other than statistical mechanics and function transforms to study the population dynamics of evolutionary algorithms.

The rest of this paper is organized as follows. In Section 2 we provide a review on research efforts directly related to our work. In Section 3 we describe our Gaussian mixture model. The analysis of the L^1-norm error is presented in Section 4. Experimental results are presented in Section 5. Finally, in Section 6 we conclude the paper and suggest possible future work.

2. RELATED WORK

As our method is different from traditional approaches, we draw our inspirations from diverse sources. It is our understanding that the greatest difficulty in calculating an infinite population model in the general case lies in the fact that the fitness function can be of any form. This mandates that in order to solve the transition equation analytically and iterate it recursively, the approximation framework must assume "compatible" forms of both the fitness function and population density functions. In this regard, the research of [14,15] is illuminating. Though the crux of these papers is not to devise an approximation framework as in this one, in the analysis the authors directly applied the Walsh transform (the Fourier transform for binary representations) to mixing matrix (or population density function in this context) and at some point to the fitness functions. However, as we are more concerned with evolutionary algorithms on continuous optimization problems and it usually incorporates operators that directly use Gaussian functions, it is more natural and simpler to approximate the fitness function and population density functions as Gaussian mixtures.

There are also a few studies on population dynamics of evolutionary algorithms solving specific problems. Among them [3] is similar to this work in that it also assumes a specific form of population density functions. The population density functions are described by probabilistic graphical models in their research. For the symbolic regression problem considered in that paper, this representation is natural and exact. However, our study differs from that paper in that the latter is mainly an empirical study which does not consider transition functions explicitly and the graphical model is learned from experimental results.

Though to our knowledge we are the first to use Gaussian mixtures to study population dynamics of evolutionary algorithms, in other fields such as estimation and filtering theory, Gaussian mixtures have been used to study dynamical systems. Among them we mention [12], which seems most relevant to our study. That study is concerned with evolving state probability density functions of nonlinear dynamical systems. Its main contribution is in proposing two novel schemes to update the weights of components of Gaussian mixtures as system uncertainty propagates. However, unlike this paper their study assumes that the number of components in the Gaussian mixture remains the same all the time, which seems unrealistic for modelling evolutionary algorithms and also introduces additional error. Besides, the approximation and propagation error of density functions is not considered in their research.

With respect to the L^1-norm error of the model, the first part of our proof is based entirely on the fundamental results of [4–6]. Basically, the study of [6] proves that Gaussian mixtures can approximate any functions in L^p, $p \in [1, +\infty)$

with arbitrarily small L^p-norm error, while the study of [4,5] gives a more detailed analysis on the error of approximations with regard to parameter values. In particular, [5] is an application of the general results in the field approximate approximations introduced by the Swedish mathematician Vladimir Maz'ya (see [4] for a more recent survey).

3. THE GAUSSIAN MIXTURE MODEL

3.1 Notations and Preliminaries

In this paper, a function a of x is commonly denoted as $a(x)$, and a is used when there is no risk of confusion. Random variables and random vectors are represented in ***boldface***. The expectation and variance of $a(\boldsymbol{x})$ with $\boldsymbol{x} \sim f(x)$ are denoted as $E_f a(\boldsymbol{x})$ and $Var_f a(\boldsymbol{x})$, respectively, or $E_{\boldsymbol{x}} a(\boldsymbol{x})$ and $Var_{\boldsymbol{x}} a(\boldsymbol{x})$ if the distribution is clear in the context. Notice that if a maps $\boldsymbol{x}$ into a vector, $Var_{\boldsymbol{x}} a$ is actually the covariance matrix of $a(\boldsymbol{x})$.

The multidimensional Gaussian function or multivariate normal distribution with parameters μ, Σ is denoted as

$$f_N(\mu, \Sigma)(x) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right]$$

where $\mu \in \mathbb{R}^d$ is the mean vector and the symmetric positive definite matrix $\Sigma \in \mathbb{R}^{d\times d}$ is the covariance matrix. A finite Gaussian mixture in $\mathbb{R}^d$ with m components is a function $h : \mathbb{R}^d \mapsto \mathbb{R}$ with the form

$$h(x) = \sum_{i=1}^{m} \pi_i f_N(\mu_i, \Sigma_i)(x) \tag{1}$$

where π_i is called the weight of component i. Notice that in the paper we are using the term "finite Gaussian mixture" to mean any function with the form (1). If the finite Gaussian mixture represents a probability density function, it is also required that $\pi_i > 0$ and $\sum_{i=1}^{m} \pi_i = 1$.This requirement is often omitted if the context is clear.

We use Euclidean norm for vectors and denote it as $\|\cdot\|$, and the L^p norm for functions is denoted as $\|\cdot\|_p$ where $p \in [1, +\infty]$. For any two functions $a, b : \mathbb{R}^d \mapsto \mathbb{R}$, denote $(a * b)(x)$ or $a * b$ as the d-dimensional convolution of a and b, i.e.

$$(a * b)(x) = \int_{\mathbb{R}^d} a(y) b(x-y) dy$$

Throughout the paper the continuous optimization problem has the following form

$$\max g(x) \text{ s.t. } x \in \mathbb{R}^d$$

where $g : \mathbb{R}^d \mapsto \mathbb{R}$ is the objective function or the so-called fitness function. We further assume that

$$0 < g_{\min} \leqslant g(x) \leqslant g_{\max} < \infty \tag{2}$$

where $g_{\min}$ and $g_{\max}$ are the known lower and upper bounds, respectively. Note that by (2) $g \in L^\infty$.

The simple evolutionary algorithm considered in this paper includes only mutation and fitness proportional selection. It is similar to the algorithm analyzed in [8]. However, we restrict the mutation operator to be adding an independent multivariate normal random vector $\boldsymbol{w} \sim f_N(\mu_w, \Sigma_w)$ to each individual in the population. In fact it is also the most commonly used mutation operator for continuous optimization problems. The pseudo-code of the algorithm is shown in Algorithm 1.

Algorithm 1: The Simple Evolutionary Algorithm

Input: population size N; initial population density function f_0

1 set $k = 0$; sample N individuals $x_0^1, x_0^2, \ldots, x_0^N$ identically and independently according to f_0;
2 **while** *stopping criteria is not satisfied* **do**
3 (Selection) choose y_k^i from $x_k^1, x_k^2, \ldots, x_k^N$ such that $\Pr\{\boldsymbol{y_k^i} = x_k^j\} = \frac{g(x_k^j)}{\sum_{l=1}^N g(x_k^l)}$ for all $i, j = 1, 2, \ldots, N$;
4 (Mutation) sample x_{k+1}^i from the distribution of random vector $y_k^i + \boldsymbol{w}$ for all $i = 1, 2, \ldots, N$;
5 $k = k + 1$;
6 **end**

In this paper the population density function of generation k is denoted as f_k. In particular, f_0 is the density function of the first population and it depends on how the initial population is constructed. To simplify the analysis of approximation errors we further assume that

$$f_0 \text{ is continuous with compact support} \tag{3}$$

The key result of [8] is the large sample theory proof of the transition function between f_k and f_{k+1} as $N \to \infty$. It also serves as the target of approximation in our research. To reframe it in this context, this theorem reads as follows.

Theorem 1 (Qi & Palmieri [8])**.** *Let f_k be the population density function of the kth generation, $k = 0, 1, 2, \ldots$, and the objective function $g(x)$ satisfy (2). For the simple evolutionary algorithm as in Algorithm 1, as $N \to \infty$ the population density function f_{k+1} satisfies*

$$f_{k+1}(x) = \frac{f_k(x) \cdot g(x)}{\int_{\mathbb{R}^d} f_k(y) g(y) dy} * f_N(\mu_w, \Sigma_w)(x) \tag{4}$$

The transition equation (4) in Theorem 1 can be divided into two parts. The effects of fitness proportional selection constitute the first part. If we denote

$$f_k'(x) = \frac{f_k \cdot g}{\int_{\mathbb{R}^d} f_k(y) g(y) dy} \tag{5}$$

then f_k' is the population density function after selection. Similarly, the effect of mutation is expressed by

$$f_{k+1}(x) = f_k' * f_N(\mu_w, \Sigma_w) \tag{6}$$

As can be seen from (4), because g can be any real-valued function, it is highly probable that (4) cannot be analytically solved. Moreover, even if f_{k+1} can be obtained by substituting g and f_k into (4), it is not likely that subsequent population density functions starting from f_{k+2} can all be obtained by substituting g and its predecessors. It is for this reason that the approximations of g and f_k should have "compatible" forms.

3.2 The Gaussian Mixture Model

The core of our research is to use finite Gaussian mixtures to approximate g and f_k, so that the transition function (4) is analytically solvable and the solution is also in Gaussian mixture form.

3.2.1 Approximation of Fitness Function g and Initial Population Density Function f_0

This section explains the method to approximate the fitness function g and the initial population density function f_0 with finite Gaussian mixtures. The idea is to sample function values on a lattice in a region of interest, and then interpolate the function with Gaussian kernels. The approach coincides with [6] and [5] and we describe it in our context.

Consider an arbitrary continuous function $u \in L^p : \mathbb{R}^d \mapsto \mathbb{R}$ for some $p \in [1, +\infty]$. The approximation procedure requires two parameters $D > 0$ and $h > 0$ to control the sampling granularity and an additional parameter $r > 0$ to control the region of sampling. The formula of interpolation reads as

$$\hat{u}(x) = \sum_{z \in \mathbb{Z}^d, hz \in [-r,r]^d} h^d \cdot u(hz) \cdot f_N(hz, \frac{Dh^2}{2} I_d) \tag{7}$$

where I_d is the d-dimensional identity matrix. Basically this procedure samples the region $[-r, r]^d$ with step size h and interpolates each sample with the Gaussian function $f_N(0, \frac{Dh^2}{2} I_d)$. The result $\hat{u}$ has $(\lfloor \frac{2r}{h} \rfloor + 1)^d$ components.

Note that in (7), if $\hat{u}$ is constructed by sampling and interpolating the whole region of $\mathbb{R}^d$, the Gaussian mixture $\hat{u}$ will have infinite number of components. Therefore, to make $\hat{u}$ a finite mixture the parameter r is introduced.

To approximate f_0, because f_0 has a compact support, r_{f_0} can be a value such that $supp(f_0) \subset [-r_{f_0}, r_{f_0}]^d$ where $supp(\cdot)$ denote the support of a function. In practice however, it may happen that f_0 does not have a compact support. In that case, we suggest a method of assigning r_{f_0} for practical use. For f_0, given a threshold $0 < \epsilon < 1$ we choose r_{f_0} such that

$$\int_{\mathbb{R}^d \setminus [-r_{f_0}, r_{f_0}]^d} |f_0(x)| dx < \epsilon \int_{\mathbb{R}^d} |f_0(x)| dx = \epsilon$$

The idea is to choose r_{f_0} such that the sampling region covers most "masses" of f_0. Such r_{f_0} can always be obtained because $\|f_0\|_1$ is 1.

As g is in L^∞, there is no general way to derive r_g. For practical use however, it is possible to derive a region of interest either from prior knowledge or from previous runs of the algorithm. For example, if prior knowledge is available that the global maximum is likely to be within a region S where $r_s = \max_{x \in S} \|x\|$ is known, we can set r_g to be $\max(r_{f_0}, r_s)$.

After r_g and r_{f_0} are fixed, given the parameter values of D and h we can approximate g and f_0 according to formula (7). The error of approximation with regard to parameters D and h will be analyzed in Section 4. In fact, it is more convenient to organize the expressions of approximated functions into standard representations of Gaussian mixtures. From now on we will use the notations that the Gaussian mixture approximation of g is

$$\hat{g}(x) = \sum_{i=1}^{n} c_i f_N(\mu_i, \Sigma_i) \tag{8}$$

and the Gaussian mixture approximation of the kth population density function $f_k, k = 0, 1, 2, \ldots$ is

$$\hat{f}_k(x) = \sum_{i=1}^{m_k} \pi_{ki} f_N(\mu_{ki}, \Sigma_{ki}) \tag{9}$$

3.2.2 Approximation of Transition Equation

By approximating g and f_0 by $\hat{g}$ and $\hat{f}_0$, an approximated dynamical system is constructed. The input to the system is $\hat{g}$ and $\hat{f}_0$, and the evolution of the system follows the transition equation

$$\hat{f}_{k+1}(x) = \frac{\hat{f}_k \cdot \hat{g}}{\int_{\mathbb{R}^d} \hat{f}_k(y)\hat{g}(y)dy} * f_N(\mu_w, \Sigma_w) \tag{10}$$

.

The idea is that evolving the real system (with g and f_0 as its input and (4) as the transition equation) and obtaining its population dynamics is difficult, while evolving the approximated system and observing its population dynamics $\hat{f}_k$ and key statistics such as $E_{\hat{f}_k}\hat{g}$ is easy. The transition equation of the approximated system needs not change from (4) as they are both modeling the same operators, and the error is only introduced by the approximations of g and f_0. In this section, we explain how the approximated dynamical system evolves and calculate the approximated population density function $\hat{f}_k$ by solving the transition equation (10). The key result is presented in Theorem 2 which gives a recursive equation between $\hat{f}_k$ and $\hat{f}_{k+1}$.

Theorem 2. *Let $\hat{g}$ and $\hat{f}_k$ be the approximated fitness function and the population density function of the kth generation, respectively, $k = 0, 1, 2, \ldots$, and assume that $\hat{g}$ and $\hat{f}_k$ are Gaussian mixtures with known coefficients as represented in (8) and (9). Then for the transition function (10) the population density function $\hat{f}_{k+1}$ is also a Gaussian mixture and has the following form*

$$\hat{f}_{k+1} = \frac{\sum_{i=1}^{m_k} \sum_{j=1}^{n} \pi_{ki} c_j \tilde{c}_{ij} f_N(\tilde{\mu}_{ij} + \mu_w, \tilde{\Sigma}_{ij} + \Sigma_w)}{\sum_{i=1}^{m_k} \sum_{j=1}^{n} \pi_{ki} c_j \tilde{c}_{ij}} \tag{11}$$

where

$$\begin{aligned} \tilde{c}_{ij} &= f_N(\mu_j, \Sigma_{ki} + \Sigma_j)(\mu_{ki}) \\ \tilde{\Sigma}_{ij} &= (\Sigma_{ki}^{-1} + \Sigma_j^{-1})^{-1} \\ \tilde{\mu}_{ij} &= (\Sigma_{ki}^{-1} + \Sigma_j^{-1})^{-1}(\Sigma_{ki}^{-1}\mu_{ki} + \Sigma_j^{-1}\mu_j) \end{aligned} \tag{12}$$

Proof. Similar to (5) and (6), we consider the effects of selection and mutation separately. The approximated population density function after selection is

$$\hat{f}'_k(x) = \frac{\hat{f}_k \cdot \hat{g}}{\int_{\mathbb{R}^d} \hat{f}_k(y)\hat{g}(y)dy} = \frac{\hat{f}_k \cdot \hat{g}}{E_{\hat{f}_k}\hat{g}} \tag{13}$$

Consider $\hat{f}_k \cdot \hat{g}$ only. By using the fact (8.1.8 in [7]) that

$$f_N(\mu_a, \Sigma_a) \cdot f_N(\mu_b, \Sigma_b) = c f_N(\mu_c, \Sigma_c)$$

where

$$\begin{aligned} c &= f_N(\mu_b, \Sigma_a + \Sigma_b)(\mu_a) \\ \Sigma_c &= (\Sigma_a^{-1} + \Sigma_b^{-1})^{-1} \\ \mu_c &= (\Sigma_a^{-1} + \Sigma_b^{-1})^{-1}(\Sigma_a^{-1}\mu_a + \Sigma_b^{-1}\mu_b) \end{aligned}$$

$\hat{f}_k \cdot \hat{g}$ can be calculated as

$$\begin{aligned}\hat{f}_k\hat{g} &= \sum_{i=1}^{m_k}\sum_{j=1}^{n}\pi_{ki}c_j f_N(\mu_{ki},\Sigma_{ki})f_N(\mu_j,\Sigma_j)\\ &= \sum_{i=1}^{m_k}\sum_{j=1}^{n}\pi_{ki}c_j\tilde{c}_{ij}f_N(\tilde{\mu}_{ij},\tilde{\Sigma}_{ij})\end{aligned} \tag{14}$$

where $\tilde{c}_{ij}$, $\tilde{\mu}_{ij}$ and $\tilde{\Sigma}_{ij}$ satisfies (12). Also by (14)

$$E_{\hat{f}_k}\hat{g} = \sum_{i=1}^{m_k}\sum_{j=1}^{n}\pi_{ki}c_j\tilde{c}_{ij} \tag{15}$$

Combining (13), (14) and (15) we get

$$\hat{f}'_k(x) = \frac{\sum_{i=1}^{m_k}\sum_{j=1}^{n}\pi_{ki}c_j\tilde{c}_{ij}f_N(\tilde{\mu}_{ij},\tilde{\Sigma}_{ij})}{\sum_{i=1}^{m_k}\sum_{j=1}^{n}\pi_{ki}c_j\tilde{c}_{ij}} \tag{16}$$

Now the effect of mutation can be characterized as

$$\hat{f}_{k+1}(x) = \hat{f}'_k * f_N(\mu_w,\Sigma_w) \tag{17}$$

Substituting (16) into (17) and taking into account the fact (8.1.4 in [7]) that

$$f_N(\mu_a,\Sigma_a) * f_N(\mu_b,\Sigma_b) = f_N(\mu_a+\mu_b,\Sigma_a+\Sigma_b)$$

(11) can be easily verified and the proof is complete. □

Theorem 2 states that if $\hat{f}_k$ is a Gaussian mixture of m_k components, then $\hat{f}_{k+1}$ is also a Gaussian mixture and it has $m_k \cdot n$ components. Notice that it is the selection operator that increases the number of components, and for the mutation operator the component number remains the same. It can be conceived that for the approximated dynamical system, the population density function of the kth generation has $m_0 \cdot n^k$ components. As it grows exponentially, we simply limit the number of components in the mixture by discarding components with least weights in our implementation.

3.2.3 Key Statistics

In this section we show that key statistics of interest of the approximated dynamical system can also be derived from Gaussian mixture population density functions. In fact, in the proof of Theorem 2 the expected average fitness value of the population $E_{\hat{f}_k}\hat{g}$ is already calculated in (15). To be complete it is also included in the result.

Theorem 3. *Let $\hat{g}$ and $\hat{f}_k$ be the approximated fitness function and the population density function of kth generation as in Theorem 2, respectively. Then*

$$E_{\hat{f}_k}\hat{g} = \sum_{i=1}^{m_k}\sum_{j=1}^{n}\pi_{ki}c_j\tilde{c}_{ij} \tag{18}$$

$$Var_{\hat{f}_k}\hat{g} = \sum_{\gamma=1}^{m_k}\sum_{\alpha=1}^{n}\sum_{\beta=1}^{n}\pi_{k\gamma}c_\alpha c_\beta d_{\alpha\beta} + (\sum_{i=1}^{m_k}\sum_{j=1}^{n}\pi_{ki}c_j\tilde{c}_{ij})^2 \tag{19}$$

$$E_{\hat{f}_k}\boldsymbol{x} = \bar{\mu} \tag{20}$$

$$Var_{\hat{f}_k}\boldsymbol{x} = \sum_{i=1}^{m_k}\pi_{ki}\Sigma_{ki} + \sum_{i=1}^{m_k}\pi_{ki}(\mu_{ki}-\bar{\mu})(\mu_{ki}-\bar{\mu})^T \tag{21}$$

where

$$\begin{aligned}\tilde{c}_{ij} &= f_N(\mu_j,\Sigma_{ki}+\Sigma_j)(\mu_{ki})\\ d_{\alpha\beta} &= f_N(\mu_\beta,\Sigma_\alpha+\Sigma_\beta)(\mu_\alpha)\\ \bar{\mu} &= \sum_{i=1}^{m_k}\mu_{ki}\end{aligned} \tag{22}$$

Proof. (18) is already proved. For $Var_{\hat{f}_k}\hat{g}$, noticing that

$$Var_{\hat{f}_k}\hat{g} = E_{\hat{f}_k}\hat{g}^2 - (E_{\hat{f}_k}\hat{g})^2$$

and taking into account that $\hat{g}^2$ is also a Gaussian mixture with n^2 components, (19) can be proved by taking a similar approach to the one used in Theorem 2.

The remaining two equations are properties of finite Gaussian mixtures. The proof of (20) is straightforward and thus omitted. For (21), consider a random variable $\boldsymbol{s}$ following a discrete distribution that

$$\Pr(\boldsymbol{s}=i) = \pi_{ki}, i = 1,2,\ldots,m_k$$

Define a d-dimensional random vector $\boldsymbol{Y}$ following the conditional distribution that

$$(\boldsymbol{Y}|\boldsymbol{s}=i) \sim f_N(\mu_{ki},\Sigma_{ki})$$

It is known that the distribution of $\boldsymbol{Y}$ is equivalent to the distribution of $\boldsymbol{x}$ and they all follow the same Gaussian mixture distribution $\hat{f}_k$. By adopting the law of total covariance that

$$Var_{\hat{f}_k}\boldsymbol{Y} = E_{\boldsymbol{s}}(Var_{\boldsymbol{Y}|\boldsymbol{s}}\boldsymbol{Y}) + Var_{\boldsymbol{s}}(E_{\boldsymbol{Y}|\boldsymbol{s}}\boldsymbol{Y}) \tag{23}$$

and calculating the two terms in (23), (21) can be proved. □

Theorem 3 gives expressions of the expected average fitness value and the variance of fitness values of the population as well as the expected mean center and the covariance matrix of the population. Though our approximated model is a microscopic description, these statistics can still be calculated with ease due to the nice properties of Gaussian mixtures. Moreover, by explicitly yielding an expression of population density function of each generation, our model provides more details of the dynamics of evolutionary algorithms than statistical mechanics methods.

4. APPROXIMATION ERROR ANALYSIS

In this section we study the L^1-norm error between the approximated population density function and the real one. More precisely, we are mostly concerned with $\|f_k - \hat{f}_k\|_1$ for $k = 0,1,2,\ldots$, where f_k and $\hat{f}_k$ are specified in Theorem 1 and Theorem 2, respectively. The analysis is on the ideal case that the number of components in a finite Gaussian mixture can be arbitrarily large, i.e. the approximation system has infinite memory. As a result, the error of truncating mixtures in implementation is not considered here. As expounded in Section 3, the approximation error stems from the interpolation of f_0 and g. Approximating f_0 introduces the error $\|f_0 - \hat{f}_0\|_1$. Then, as the two systems evolve according to the transition equations (4) and (10) respectively, the approximation error propagates and amplifies under the influence of g and $\hat{g}$. Therefore, to analyze the error of the whole system, the errors of initially approximating f_0 and g as well as the amplification effect of the transition equations must be analyzed first.

4.1 Approximation Errors of f_0 and g

This section analyzes the approximation errors of f_0 and g. The key result in this section is entirely based on [6] and [4,5]. Due to the high degree of technicality in these papers it is both impractical and unnecessary to include all details of proofs in this paper. Therefore key theorems relevant to our research are included and the implications for our research are discussed.

Theorem 4 (Park & Sandberg [6]). *Let $K : \mathbb{R}^r \mapsto \mathbb{R}$ be an integrable bounded function such that K is continuous almost everywhere and $\int_{\mathbb{R}^r} K(x)dx \neq 0$. Then the family S_K is dense in L^p for every $p \in [1, \infty)$ where S_K consists of functions $q : \mathbb{R}^r \mapsto \mathbb{R}$ represented by*

$$q(x) = \sum_{i=1}^{M} \omega_i \cdot K\left(\frac{x - z_i}{\sigma}\right) \tag{24}$$

where $\sigma > 0$, $M \in \mathbb{N}$, $\omega_i \in \mathbb{R}$ and $z_i \in \mathbb{R}^r$.

Theorem 4 is the main result in [6]. It proves that under certain mild conditions (bounded, integrable and continuous almost everywhere) the function family S_K can approximate any function in L^p with arbitrarily small L^p-norm error. If we take the function $f_N(0, I_d)$ as K, apparently it satisfies the conditions in Theorem 4. Therefore the family $S_{f_N(0,I_d)}$ is dense in L^p. By merging constants into ω_i, $S_{f_N(0,I_d)}$ consists of functions of the form

$$q(x) = \sum_{i=1}^{M} \omega_i \cdot f_N(z_i, \sigma^2 I_d) \tag{25}$$

Therefore we have

Corollary 1. *The family of Gaussian mixtures of functions in the form of (25) is dense in L^p with $p \in [1, +\infty)$.*

Comparing (7) with (25), it can be seen that $\hat{u}$ is in $S_{f_N(0,I_d)}$ and $\hat{u}$ corresponds to an instance of (25) where the values of z_i, ω_i, σ and M are decided by r, h, D and $u(\cdot)$. In fact, though Corollary 1 states that the family $S_{f_N(0,I_d)}$ has the ability to universally approximate any functions in L^p, it does not necessarily prove that the approximation formula (7) can achieve the same goal. After carefully examining the proof of Theorem 4 in [6] however, we found that the authors in fact proved Theorem 4 by first proving that formula (7) is able to universally approximate any continuous functions with compact support in L^p as $\sqrt{D}h \to 0$, though this fact was not presented as a lemma or a theorem in the paper. Combining this fact with the fact that f_0 is continuous with compact support and is in L^1, we have

Theorem 5. *Let f_0 be the density function of the first population and satisfy (3). Then there exist appropriate values of r, D and h for f_0 such that by applying formula (7) to f_0 the approximation error $\|f_0 - \hat{f}_0\|_1$ can be arbitrarily small.*

For the approximation of g, because g satisfies (2), g is in L^∞ and does not have compact support. Therefore, the reasoning leading to Theorem 5 is not applicable to g. In this regard, [4,5] provides further analysis. Based on Lemma 2.1 in [5] and its related discussions we formulate one relevant result as a theorem and discuss its implications.

Theorem 6 (Maz'ya & Schmidt [5]). *For any function $u \in C^2(\mathbb{R}^n) \cap W^2_\infty(\mathbb{R}^n)$ it holds that*

$$\begin{aligned} |u(x) - \tilde{u}(x)| \leqslant |u(x)| &\sum_{v \in \mathbb{Z}^n \setminus \{0\}} \exp(-D\pi^2 \|v\|^2) \\ &+ Dh\pi \sum_{|\boldsymbol{\alpha}|=1} |\partial^{\boldsymbol{\alpha}} u(x)| \sum_{v \in \mathbb{Z}^n \setminus \{0\}} |v^{\boldsymbol{\alpha}}| \exp(-D\pi^2 \|v\|^2) \\ &+ (\sqrt{D}h)^2 \sum_{|\boldsymbol{\alpha}|=2} \rho_{\boldsymbol{\alpha}} \|\partial^{\boldsymbol{\alpha}} u\|_\infty \end{aligned} \tag{26}$$

where

$$\tilde{u}(x) = (\pi D)^{-\frac{n}{2}} \sum_{m \in \mathbb{Z}^n} u(hm) \exp\left(-\frac{\|x - hm\|^2}{Dh^2}\right) \tag{27}$$

and

$$\rho_{\boldsymbol{\alpha}} = \frac{1}{\boldsymbol{\alpha}!} \left\| D^{-n/2} \sum_{m \in \mathbb{Z}^n} \left| \left(\frac{\cdot - m}{\sqrt{D}}\right) \eta \left(\frac{\cdot - m}{\sqrt{D}}\right) \right| \right\|_\infty \tag{28}$$

In Theorem 6, $\boldsymbol{\alpha}$ is a multi-index $(\alpha_1, \alpha_2, \ldots, \alpha_n) \in \mathbb{Z}^n_{\geqslant 0}$ and $|\boldsymbol{\alpha}| = \alpha_1 + \cdots + \alpha_n$, $x^{\boldsymbol{\alpha}} = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$, $\boldsymbol{\alpha}! = \alpha_1! \cdots \alpha_n!$ and $\partial^{\boldsymbol{\alpha}} u(x) = \frac{\partial^{|\boldsymbol{\alpha}|}}{\partial x_1^{\alpha_1} \cdots \partial x_n^{\alpha_n}} u(x)$. The Sobolev space W_p^N, $p \in [1, +\infty]$ and $N \in \mathbb{N}$ consists of functions in L^p whose generalized derivatives up to order N also belong to L^p. C^N consists of continuous functions whose generalized derivatives up to order N exist and are all continuous.

To see the relevance of Theorem 6 to our research, first note that $\tilde{u}$ in (27) is equivalent to $\hat{u}$ in (7) when $r = +\infty$. Secondly, as $u \in W^2_\infty$, $\|u\|_\infty$ and $\|\partial^{\boldsymbol{\alpha}} u\|_\infty, |\boldsymbol{\alpha}| = 1, 2$ are all finite. Therefore the terms related to u in (26) can all be bounded by constants. As a consequence, if D is fixed and $h \to 0$, the second and third terms in (26) vanish and the remaining term or the so-called saturation error is $\|u(x)\|_\infty \sum_{v \in \mathbb{Z}^n \setminus \{0\}} \exp(-D\pi^2 \|v\|^2)$. It can be seen that this term can be arbitrarily small when $D \to +\infty$. Combining these facts, it can be concluded that $\|u - \tilde{u}\|_\infty$ can be arbitrarily small if the values of h and D are chosen appropriately.

For the approximation error of $\hat{u}$, it is proved in chapter 2.3.2 of [4] that $\|\tilde{u} - \hat{u}\|_\infty$ can be arbitrarily small by choosing appropriate values of r in (7) when D and h are all fixed. Because $\|u - \hat{u}\|_\infty \leqslant \|u - \tilde{u}\|_\infty + \|\tilde{u} - \hat{u}\|_\infty$, combining this fact with all previous reasoning it can be concluded that $\|u - \hat{u}\|_\infty$ can be arbitrarily small if the values of D, h and r are chosen appropriately. In summary, we have

Theorem 7. *Let $g \in C^2 \cap W^2_\infty$ be the objective function. Then there exist appropriate values of r, D and h for g such that by applying formula (7) to g the approximation error $\|g - \hat{g}\|_\infty$ can be arbitrarily small.*

4.2 Propagation Error

In this section we analyze the amplification effects of transition functions on approximation errors. More specifically, assume the error of the kth generation $\|f_k - \hat{f}_k\|_1$ is known, we want to derive an upper bound for $\|f_{k+1} - \hat{f}_{k+1}\|_1$.

Theorem 8. *Let g and $\hat{g}$ be the real and approximated fitness functions and f_k and $\hat{f}_k$ be the real and approximated population density functions of the kth generation, $k = 0, 1, 2, \ldots$. Assume that g satisfies (2) and $\|f_k - \hat{f}_k\|_1$ and $\|g - \hat{g}\|_\infty$ are known. Then*

$$\|f_{k+1} - \hat{f}_{k+1}\|_1 \leqslant \frac{2g_{\max}}{g_{\min}} \|f_k - \hat{f}_k\|_1 + \frac{2\|g - \hat{g}\|_\infty}{g_{\min}} \tag{29}$$

Furthermore, if $\|f_0 - \hat{f}_0\|_1$ *is known, it holds that*

$$\|f_k - \hat{f}_k\|_1 \leqslant \alpha^k \|f_0 - \hat{f}_0\|_1 + \|g - \hat{g}\|_\infty \cdot \frac{1-\alpha^k}{1-\alpha}\beta \quad (30)$$

where $\alpha = \frac{2g_{\max}}{g_{\min}}$ *and* $\beta = \frac{2}{g_{\min}}$.

Proof. To be concise we denote $\int_{\mathbb{R}^d} a(x)b(x)dx$ as $\int ab$. Adopting the notations in the proof of Theorem 2, we consider the effects of selection and mutation separately.

To analyze the effect of selection, first note that as g satisfies (2) and the Gaussian mixtures $\hat{f}_k, \hat{g} > 0$ are continuous functions,

$$0 < g_{\min} \leqslant \int f_k g = \left|\int f_k g\right| = \int |f_k g|$$

and

$$0 < \int \hat{f}_k \hat{g} = \left|\int \hat{f}_k \hat{g}\right| = \int \left|\hat{f}_k \hat{g}\right|$$

The L^1-norm difference after selection is

$$\begin{aligned}
&\left\|f'_k - \hat{f}'_k\right\|_1 = \left\|\frac{f_k g}{\int f_k g} - \frac{\hat{f}_k \hat{g}}{\int \hat{f}_k \hat{g}}\right\|_1 \\
=&\frac{\left\|\int \hat{f}_k \hat{g} \cdot \left(f_k g - \hat{f}_k \hat{g}\right) + \hat{f}_k \hat{g} \cdot \left(\int \hat{f}_k \hat{g} - \int f_k g\right)\right\|_1}{\int f_k g \int \hat{f}_k \hat{g}} \\
\leqslant&\frac{\int \hat{f}_k \hat{g} \cdot \left\|f_k g - \hat{f}_k \hat{g}\right\|_1 + \int \hat{f}_k \hat{g} \cdot \left|\int \hat{f}_k \hat{g} - \int f_k g\right|}{\int f_k g \int \hat{f}_k \hat{g}} \\
\leqslant&\frac{2\left\|f_k g - \hat{f}_k g + \hat{f}_k g - \hat{f}_k \hat{g}\right\|_1}{\int f_k g} \\
\leqslant&\frac{2}{\int f_k g}\left(\left\|g(f_k - \hat{f}_k)\right\|_1 + \left\|\hat{f}_k (g - \hat{g})\right\|_1\right) \\
\leqslant&\frac{2}{g_{\min}}\left(g_{\max}\left\|f_k - \hat{f}_k\right\|_1 + \|g - \hat{g}\|_\infty\right)
\end{aligned}$$

In the proof we used the Minkowski inequality $\|a+b\|_1 \leqslant \|a\|_1 + \|b\|_1$, Hölder's inequality $\|ab\|_1 \leqslant \|a\|_1 \|b\|_\infty$ and the inequality $\int a \leqslant \int |a|$.

To analyze the effect of mutation, first we denote $h = f_N(\mu_w, \Sigma_w)$. Obviously $\|h\|_1 = 1$. The L^1-norm difference after mutation is

$$\begin{aligned}
\left\|f_{k+1} - \hat{f}_{k+1}\right\|_1 &= \left\|f'_k * h - \hat{f}'_k * h\right\|_1 = \left\|(f'_k - \hat{f}'_k) * h\right\|_1 \\
&\leqslant \left\|f'_k - \hat{f}'_k\right\|_1 \|h\|_1 \text{ by Young's inequality} \\
&= \left\|f'_k - \hat{f}'_k\right\|_1
\end{aligned}$$

Combining the two inequalities, the proof of (29) is complete. (30) is obtained by expanding (29) recursively until $\|f_0 - \hat{f}_0\|_1$.

□

4.3 Overall Error

By applying Theorem 5 and Theorem 7, the initial approximation errors $\|f_0 - \hat{f}_0\|_1$ and $\|g - \hat{g}\|_\infty$ can be arbitrarily small. As a consequence, if k is fixed in (30), $\|f_k - \hat{f}_k\|_1$ can be arbitrarily small. Therefore the error introduced by the Gaussian mixture model can be arbitrarily small up to a predefined generation k. This result is summarized in the following theorem.

Theorem 9. *Let* f_0 *be the density function of the first population and satisfy (3), and* $g \in C^2 \cap W^2_\infty$ *be the objective function and satisfy (2). Then given a threshold* $0 < \epsilon < 1$ *and a generation number* $k \in \mathbb{N}$*, there exist appropriate values of* r*,* D *and* h *for* f_0 *and* g*, respectively such that by applying formula (7) to them and evolving the two systems according to Theorem 1 and Theorem 2, the approximation error* $\|f_l - \hat{f}_l\|_1 < \epsilon$ *for all* $l \in \mathbb{N}, l \leqslant k$.

5. EXPERIMENTAL RESULTS

In this section we apply the finite Gaussian model on a simple low dimensional optimization problem to illustrate the usefulness of the model in predicting the behavior of evolutionary algorithms. In the experiment the solution space is $\mathbb{R}^2$ and the objective function under consideration is an isotropic Gaussian mixture with two components

$$g(x) = c_1 f_N(\mu_1, \Sigma) + c_2 f_N(\mu_2, \Sigma)$$

where $\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} 50000 \\ 40000 \end{pmatrix}$, $\mu_1 = \begin{pmatrix} 2.5 \\ 2.5 \end{pmatrix}$, $\mu_2 = \begin{pmatrix} -2.5 \\ -2.5 \end{pmatrix}$ and $\Sigma = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$. Though the problem is simple it is sufficient to illustrate our proposed approach. Besides, we mention that in general finding the maximum or all modes of a multidimensional Gaussian mixture is not an easy task [2].

The initial population density function f_0 is a Gaussian function with $\mu_{f_0} = (0, -2.5)$ and $\Sigma_{f_0} = 64 I_2$. We evolve the Gaussian mixture model according to Theorem 2 for 50 generations. We also run Algorithm 1 with population size of 100, 1000, 10000 and 100000 to solve this problem once respectively. The mutation vector $\boldsymbol{w}$ follows $f_N(0, I_2)$, and the stopping criteria of the algorithm is the iteration number reaching 50. The expected average fitness value $E_{\hat{f}_k}\hat{g}$ and the expected population mean center $E_{\hat{f}_k}\boldsymbol{x}$ are calculated for each generation, and they are compared with the values observed by running the real algorithm. To facilitate calculation we set the component number limit to be 2^{15}, i.e. we preserve the largest 2^{15} components in the calculation. The simulation results are depicted in Figure 1 and Figure 2.

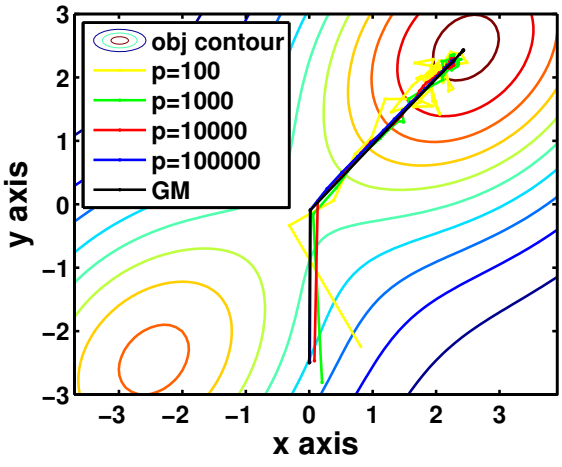

Figure 1: Trajectories of population mean center predicted by our proposed approximation and observed from running the real algorithm

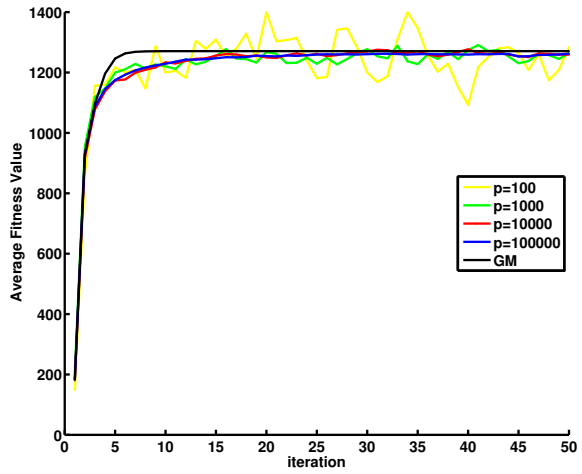

Figure 2: Comparison of average fitness values predicted by our proposed approximation with observed values from running the real algorithm

In Figure 1 the line in black (marked GM) is the trajectory of predicted population mean center calculated by $E_{\hat{f}_k}\boldsymbol{x}$, while lines in other colors (marked $p = 100$, 1000, 10000, and 100000) are the real population means observed by running the algorithm with different population sizes. Similarly, in Figure 2 we depict average fitness values of our prediction and that of real observations versus iteration number, respectively. From both figures it can be seen that our model can predict key statistics of evolutionary algorithm with fair accuracy, and as the population size grows, the error becomes smaller. This is due to the fact that our model is aimed at predicting the behavior of evolutionary algorithms with infinite population size. Since in the experiment f_0 and g are both Gaussian mixtures, the error comes from the finite population effect and the truncation of Gaussian mixtures during the calculation.

6. CONCLUSION

In this paper we propose a novel finite Gaussian mixture model to approximate the behavior of simple evolutionary algorithms. Our model is conceptually straightforward yet it has many nice properties. In the model the transition function can be easily iterated and key statistics of interest can be calculated by simple formula. As it also yields explicit expressions of population density functions, it can provide more insight into the dynamics of evolutionary algorithms than traditional approaches. In addition, based on [4–6], we proved that in theory our model can approximate the dynamics of simple evolutionary algorithm with arbitrarily small error. The experimental result illustrated the effectiveness and usefulness of the model.

For future work, it is possible to extend this research in many directions. Firstly, the errors due to finite population in real algorithm and the truncation of mixtures in the implementation can be analyzed. Secondly, to simplify things, the crossover operator is not considered in this paper. However, it would not be too difficult to apply our model on elementary crossover operators. Last but not least, the truncation method in the implementation is a very elementary method. It is possible to adopt more advanced mixture component number reduction algorithms in our framework to cope with more complex problems.

7. REFERENCES

[1] H.-G. Beyer. *The Theory of Evolution Strategies*. Springer, Berlin, 2001.

[2] M. Carreira-Perpinan. Mode-finding for mixtures of Gaussian distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1318–1323, 2000.

[3] E. Hemberg, C. Berzan, K. Veeramachaneni, and U.-M. O'Reilly. Introducing graphical models to analyze genetic programming dynamics. In *Proceedings of the Twelfth Workshop on Foundations of Genetic Algorithms XII*, FOGA XII '13, pages 75–86, 2013.

[4] V. Maz'ya. *Approximate approximations*. American Mathematical Society, Providence, R.I., 2007.

[5] V. Maz'ya and G. Schmidt. On approximate approximations using Gaussian kernels. *IMA Journal of Numerical Analysis*, 16(1):13–29, 1996.

[6] J. Park and I. W. Sandberg. Universal approximation using radial-basis-function networks. *Neural Computation*, 3(2):246–257, 1991.

[7] K. B. Petersen and M. S. Pedersen. The matrix cookbook. `http://www2.imm.dtu.dk/pubdb/p.php?3274`, Nov 2012. Version 20121115.

[8] X. Qi and F. Palmieri. Theoretical analysis of evolutionary algorithms with an infinite population size in continuous space. Part I: Basic properties of selection and mutation. *IEEE Transactions on Neural Networks*, 5(1):102–119, 1994.

[9] X. Qi and F. Palmieri. Theoretical analysis of evolutionary algorithms with an infinite population size in continuous space. Part II: Analysis of the diversification role of crossover. *IEEE Transactions on Neural Networks*, 5(1):120–129, 1994.

[10] C. R. Reeves and J. E. Rowe. *Genetic Algorithms - Principles and Perspectives : A Guide to GA Theory*. Kluwer Academic Publishers, Boston, 2003.

[11] J. L. Shapiro. Statistical mechanics theory of genetic algorithms. In L. Kallel, B. Naudts, and A. Rogers, editors, *Theoretical Aspects of Evolutionary Computing*, Natural Computing Series, chapter 5, pages 87–108. Springer Berlin Heidelberg, 2001.

[12] G. Terejanu, P. Singla, T. Singh, and P. D. Scott. Uncertainty propagation for nonlinear dynamic systems using Gaussian mixture models. *Journal of Guidance, Control, and Dynamics*, 31(6):1623–1633, 2008.

[13] M. D. Vose. *The Simple Genetic Algorithm : Foundations and Theory*. MIT Press, Cambridge, Mass., 1999.

[14] M. D. Vose and A. H. Wright. The simple genetic algorithm and the Walsh transform: Part I, theory. *Evolutionary Computation*, 6(3):253–273, 1998.

[15] M. D. Vose and A. H. Wright. The simple genetic algorithm and the Walsh transform: Part II, the inverse. *Evolutionary Computation*, 6(3):275–289, 1998.