Modeling and Analysis of Genetic Algorithms
Based on the Viewpoint of Mixture Systems

Jun-ichi Imai1 and Hiroyuki Shioya2
Muroran Institute of Technology
27–1 Mizumoto-cho, Muroran, 050–8585, JAPAN
E-mail: 1imai@mail.svbl.muroran-it.ac.jp
2shioya@csse.muroran-it.ac.jp

Masahito Kurihara
Graduate School of Engineering,
Hokkaido University
Kita 13, Nishi 8, Sapporo, 060–8628, JAPAN
E-mail: kurihara@main.eng.hokudai.ac.jp

Abstract: Some mathematical models have been proposed for theoretical analyses of genetic algorithms (GAs). However, these works have limited their objects to a few kinds of GAs in order to formulate them accurately. In this paper, we regard a GA as an information source that generates input-output data. That is, we regard a population and its next population generated by the GA as input and output respectively. Then we model the GA by learning from these data. Since this method uses only the input-output relations of data and ignores interior structures, we can describe a variety of GAs in a common form, and analyze them from a new point of view. We use some mixture models for a representation of these input-output relations in this paper. By using a mixture model for modeling a GA, we can represent the GA system as a combination of some partial systems. In this paper, we treat two types of mixture models, and investigate how these models are effective for analyzing GAs through some numerical experiments.

Key words: genetic algorithm, mixture model, vector field, modeling by learning

1. Introduction
Genetic algorithms (GAs) [1] are randomized search techniques based on the ideas from biological evolution. They have been used successfully for a variety of optimization problems.

For theoretical analyses of GAs, some mathematical models such as using differential equations [2], Markov chains [3–5], and discrete dynamical systems [6–8] have been proposed. Furthermore, the relationship between Markov chains and discrete dynamical systems has been discussed in [9–10]. However, these works have limited their objects to a few kinds of GAs in order to formulate them accurately. Vose’s GA model [10] is considered to be one of the most general models because its framework (such as a representation of populations by genotype distributions and a relationship between finite and infinite populations) can be applied to a variety of GAs. However even in this model, the heuristic function, which is one of main components of Vose’s model, is formulated only for simple GAs (SGAs).

In this paper, we regard a GA as an information source that generates input-output data. That is, we regard a population and its next population generated by the GA as input and output respectively. Then we model the GA by learning from these data. Since this method uses only the input-output relations of data and ignores interior structures, we can describe a variety of GAs in a common form, and analyze them from a new point of view.

We use some mixture models for a representation of these input-output relations in this paper. By using a mixture model for modeling a GA, we can represent the GA system as a combination of some partial systems. In this paper, we treat two types of mixture models, and investigate how these models are effective for analyzing GAs through some numerical experiments.

2. Preliminaries
2.1. Review of Vose’s GA model
Let $P$ be a population that consists of $n$ individuals. Each individual in $P$ is represented by a binary string of length $l$. $P$ corresponds to a multiset of $n$ elements of $\Omega$, where $\Omega = \{0, 1\}^l$ is the set of all types of binary strings of length $l$, that is, genotypes. Since $l$-digit binary representations of integers $0$ through $2^l - 1$ coincide with genotypes in $\Omega$, they are regarded as being the same.

$P$ is characterized by its genotype distribution $x = (x_0, x_1, \ldots, x_{2^l - 1})^T (\in \mathbb{R}_+^l)$ over $\Omega$, where $x_k = n_k / n$ ($k = 0, 1, \ldots, 2^l - 1$), $n_k$ denotes the number of individuals of genotype $k$ in $P$, and $^T$ denotes transpositions. $P$ is unambiguously determined by the corresponding distribution $x$ when $n$ is known.

Let $\Lambda$ be the set of all feasible distributions over $\Omega$. $\Lambda$ is expressed as

$$\Lambda = \left\{ x \left| \sum_{k=0}^{2^l - 1} x_k = 1, x_k \geq 0 \, (\forall k) \right. \right\}.$$  

(1)

Let $\Lambda_n (\subseteq \Lambda)$ be the set of distributions corresponding to populations of size $n$. 

If \( n \) is infinite, transitions of genotype distributions are deterministic because fluctuation due to finiteness of populations vanishes. When \( x \) is the distribution at the current generation, that at its next generation is denoted by \( G(x) \), where \( G : \Lambda \rightarrow \Lambda \) is a mapping that describes operations of the corresponding GA. Vose termed this mapping \( G \) the heuristic function. Then, the evolutionary process is described as a discrete dynamical system corresponding to a difference equation

\[
x(t+1) = G(x(t)),
\]
where \( x(t) \) denotes the distribution at the \( t \)-th generation.

When \( n \) is finite, the population at the next generation is obtained as the result of \( n \) samples chosen from \( \Omega \) according to \( G(x) \). Therefore, the conditional probability \( p(y|x) \) that \( y \) is the distribution at the next generation when \( x \) is the distribution at the current generation is given by

\[
p(y|x) = n! \prod_{k=0}^{2^n-1} \frac{(G(x)_k)^{n_k}}{(n_k)!}. \tag{3}
\]

When \( x \) is the distribution at the current generation, the conditional expectation vector of the distribution at the next generation is \( G(x) \) [11]. Transitions of finite-populations of size \( n \) are described as a Markov chain with state space \( \Lambda_n \) and transition probability from state \( x \in \Lambda_n \) to state \( y \in \Lambda_n \) given by Equation (3).

Vose has formulated the heuristic function \( G(x) \) for SGAs that consist of the selection, crossover and mutation operators. \( G \) can be expressed as a composite of a selection mapping \( F : \Lambda \rightarrow \Lambda \) and a mixing mapping \( M : \Lambda \rightarrow \Lambda \):

\[
G = M \circ F \tag{4}
\]

When the proportional selection is used as the selection operator, \( F \) is expressed as

\[
F(x) = \frac{F x}{1' F x}, \tag{5}
\]
where \( 1 = (1, 1, \ldots, 1)' \) is the \( 2^l \)-dimentional vector, \( F \) denotes the \( 2^l \)-dimentional diagonal matrix of which components are defined as

\[
F_{i,j} = \begin{cases} \text{fitness}(i), & \text{if } i = j \\ 0, & \text{otherwise} \end{cases} \tag{6}
\]
and \( \text{fitness}(i) \) denotes the fitness value assigned to genotype \( i \). \( M \) is expressed as

\[
M(x) = \left((\sigma_0 x)^' M(\sigma_0 x), \ldots, (\sigma_2 x)^' M(\sigma_2 x)\right), \tag{7}
\]
where \( M \) denotes the \( 2^l \)-dimentional symmetric matrix determined by the crossover rate \( p_c \) and mutation rate \( p_m \), and its component \( M_{i,j} \) denotes a probability that the individual of genotype \( 0 \) is generated from the parents of genotype \( i \) and \( j \). \( \sigma_0 \) denotes the \( 2^l \)-dimentional permutation matrix of which components are defined as

\[
(\sigma_0)_{i,j} = \begin{cases} 1, & \text{if } i \oplus j = k \\ 0, & \text{otherwise} \end{cases}, \tag{8}
\]

where \( \oplus \) denotes the bitwise exclusive-or operator.

### 2.2. The mapping that characterizes the genetic algorithm

The heuristic function \( G \) plays an important role in Vose’s GA model. \( G \) describes transitions of infinite-populations (Equation (2)) and appears in the description of transitions of finite-populations (Equation (3)). That is, \( G \) characterizes the corresponding GA in the sense that transitions of populations of any size can be described when \( G \) is known. Since characteristics of \( G \) such as the number of fixed points reflect that of the corresponding GA, we can investigate the characteristics of the GA through an analysis of the heuristic function \( G \).

It should be noted that there exists the heuristic function \( G \) for any kind of GA even though it is explicitly formulated only for SGAs. Therefore, the framework of Vose’s model except for the formulation of \( G \) (such as a representation of populations by genotype distributions, Markov chains for finite-populations, and discrete dynamical systems for infinite-populations) can be applied to a variety of GAs.

### 2.3. Heuristic functions and vector fields

As mentioned above, the transitions of infinite-populations are described as the discrete dynamical system corresponding to the heuristic function \( G \). Figure 1 illustrates an orbit of the dynamical system for \( l = 2 \) as a simple example. Parameter settings for this example are shown in Table 1. Since \( l = 2 \), the state space \( \Lambda \) is a 3-dimentional subspace embedded in \( \mathbb{R}^l \). \( \Lambda \) is denoted by a regular tetrahedron in Figure 1. A vertex of \( \Lambda \) denotes a population that
consists of individuals of same genotype, and the genotype \( k \) is indicated by “\((k)\)”.

The initial point of the dynamical system corresponds to the initial population for the GA. An orbit converges to an attractor through iterations of \( G \). This describes that the GA searches for a solution with higher fitness value and finally converges. Attractors correspond to the distributions of the populations dominated by individuals that represent the optimal solution (or the local optimal solution). The orbit in Figure 1 corresponds to the optimal solution and the local populations dominated by individuals that represent the optimal solution (or the local optimal solution).

In Figure 1, the genotype 0 (= “00”) and 3 (= “11”) correspond to the initial population for the GA. An orbit corresponds to the initial population for the GA. An orbit converges to an attractor through iterations of \( G \). This describes that the GA searches for a solution with higher fitness value and finally converges.

3. Modeling GAs by learning from data

It has already been mentioned that the heuristic function \( G \) characterizes the GA. However, we can obtain \( G \) only if the corresponding GA is simple and its settings such as a fitness function, kinds of operators, and values of parameters are all known. In order to solve this problem, we use a method for modeling GAs by learning from data.

3.1. Procedure for modeling

Prepare an input-output model \( f \) for modeling the GA (with string length \( l \)). Let \( f(x; \theta) \in \mathbb{R}^2 \) be an output of the model for an input \( x \in \mathbb{R}^l \), where \( \theta = (\theta_1, \theta_2, ..., \theta_d)^T \in \mathbb{R}^n \) is a parameter vector that specifies the model (such as connection weights for feed-forward neural networks). Decide the appropriate parameter according to the following procedure.

(i) Obtain the input-output data set \( S \).

Select a population \( P \) at random, and let the GA generate its next population \( Q \). Transform this pair of populations \( (P, Q) \) into the pair of the corresponding genotype distributions \( (x, y) \). Let \( S = \{(x_1, y_1), (x_2, y_2), ..., (x_d, y_d)\} \) be a set of \( d \) pairs of these input-output data.

(ii) Decide the appropriate parameter \( \hat{\theta} \) by using \( S \).

Use the least square method

\[
\hat{\theta} = \arg \min_{\theta} E(\theta),
\]

for learning from \( S \), where \( \| \cdot \| \) denotes the Euclidean norm. Use the steepest descent method for the parameter estimation.

3.2. Relationship with Vose’s model

When \( x \) is the genotype distribution at the current generation, \( G(x) \) is the conditional expectation vector of the distribution at the next generation. Therefore, if the model has the sufficient ability, it is expected that the obtained model approximates the heuristic function \( G \) (or the vector field \( V \)) that characterizes the GA.

Figure 3 illustrates the relationship among GAs, models and vector fields.

(i) Select a population \( P \) at random, and let the GA generate its next population \( Q \). Transform this pair of populations \( (P, Q) \) into the pair of the corresponding genotype distributions \( (x, y) \). Let \( S = \{(x_1, y_1), (x_2, y_2), ..., (x_d, y_d)\} \) be a set of \( d \) pairs of these input-output data.

(ii) Decide the appropriate parameter \( \hat{\theta} \) by using \( S \).

Use the least square method

\[
\hat{\theta} = \arg \min_{\theta} E(\theta),
\]

for learning from \( S \), where \( \| \cdot \| \) denotes the Euclidean norm. Use the steepest descent method for the parameter estimation.

As mentioned above, although the framework of Vose’s model can be applied to a variety of GAs, the heuristic function \( G \) is explicitly formulated only for SGAs. On the other hand, since this method uses only the input-output relations of data, we can obtain \( G \) even if the corresponding GA is
not simple, regardless of its interior structure. That is, this method for obtaining the heuristic function $G$ complements Vose’s GA model.

3.3. Analyses of GAs by using forms of models

Another purpose of this method is to describe the heuristic functions $G$ in a common form and to introduce a framework for analyzing GAs from a new point of view. Since interior structures of GAs are ignored, there is no limitation on the forms of the models only if they have sufficient abilities to represent the input-output relations. Although some forms are not direct representations of GAs, they give new means of analyzing them.

For example, modeling by using a feed-forward neural network or a radial basis function (RBF) network corresponds to the sigmoid-function expansion or the RBF expansion of $G$ respectively. Furthermore, we can use even the mapping $G (= M \ast F)$ formulated in Vose’s model for SGA. This modeling corresponds to obtaining parameters (such as fitness values assigned to each genotype, the crossover rate and mutation rate) of “the closest SGA (in the sense of Equation (11))” to the target GA.

4. Mixture models for GAs

Mixture models are tools for representing whole systems as mixtures of some partial systems (experts). In this field, there have been works such as on ensemble learning for neural networks [12] and on the mixture-of-experts architecture [13]. RBF networks and normalized Gaussian networks [14] are also considered to be kinds of mixture models of which experts are local units.

In this paper, we apply mixture models to modeling GAs. Although the mixture model is not the direct representation of GAs, we can represent the GA system as a combination of some experts by using this model.

4.1. Two types of mixture models

In this paper, we treat the following two types of mixture models.

G-mixture model: We prepare $m$ heuristic functions $G^i(x)$ ($i = 1, 2, \ldots, m$) formulated in Vose’s model that characterize SGAs under different settings as experts. These are mixed as follows (Figure 4):

$$f(x; t) = \sum_{i=1}^{m} t_i G^i(x),$$

where $t = (t_1, \ldots, t_m)$, $\sum_{i=1}^{m} t_i = 1$, $t_i \geq 0$ ($i = 1, 2, \ldots, m$). Equation (12) is formally similar to the ensemble learning for neural networks. However, the G-mixture model differs from the ensemble learning in that its experts should correspond to SGAs of which settings are different from each other.

F-mixture model: We use the selection mapping $F$ and mixing mapping $M$ of Vose’s model for SGAs. We prepare $m$ mappings $F^i(x)$ ($i = 1, 2, \ldots, m$) for different optimization problems as experts and the mixing operator $M(x)$. These are mixed as follows (Figure 5):

$$f(x; t) = M \left( \sum_{i=1}^{m} t_i F^i(x) \right),$$

where $t = (t_1, \ldots, t_m)$, $\sum_{i=1}^{m} t_i = 1$, $t_i \geq 0$ ($i = 1, 2, \ldots, m$). Although $F^i(x)$ are mixed linearly, Equation (13) is nonlinear with regard to the mixture rate $t_i$ because $M(x)$ is the nonlinear transform (Equation (7)).

4.2. Interpretations of mixture models

The G-mixture model represents a target GA by mixing some experts that represent some other GAs themselves. This corresponds to approximating the vector field $V$ by “the superposition of vector fields $V_i$.” The G-mixture model can also be regarded as the probabilistic switch of $V_i$ when we regard the mixture rate $t_i$ as a probability that the expert $i$ is selected. These two interpretations imply that this model uses GA systems characterized by experts as bases and represents the target GA as a combination of these existent structures.

In the F-mixture model, as in Vose’s model for SGAs, the selection mapping $F$ and mixing mapping $M$ are used. The mapping $F$ is dependent on the optimization problem that the GA solves and $M$
is not. Since the mixture rate $t_i$ is transformed by $M(x)$ nonlinearly, we cannot regard the $F$-mixture model as “the superposition of vector fields.” However, this model introduces an idea of “the mixture of optimization problems” instead.

These interpretations of models make the role of each expert clear. By using these mixture models with appropriate experts, it is expected that we can extract characteristics of the GA from the mixture rate $t_i$ that is, the distribution over bases.

### 4.3. Advantages of mixture models
There are two advantages of using these mixture models for modeling GAs.

The first one is reductions in computational costs for modeling. Since these models use existent structures of experts, parameters needed to decide are only mixture rates. Therefore, by using these models, we can obtain the heuristic function $G$ at lower computational costs than those using single models such as feed-forward neural networks.

The second one is extraction of characteristics of the GA. As have been mentioned, we can investigate the characteristics of the GA through an analysis of the heuristic function $G$. Therefore, there exists possibility that we can obtain some guidelines to choose parameters and operations for better performance from $G$. By using these models, we can represent the heuristic function $G$ (or the GA) as a combination of some partial systems (experts) and extract their characteristics from mixture rates over experts.

### 5. Experiments
#### 5.1. Experiment 1
We carried out experiments for modeling GAs that solve three optimization problems $f_1 - f_3$ ($l = 2$) shown in Table 2. $f_1$ and $f_2$ are the minimal deceptive problems [15] of type II and type I, respectively. Other settings of the target GA such as population size and operators are shown in Table 3. For each optimization problem, two hundred genotype distributions were chosen at random as input data. Then, output data for each input is the next distribution generated by the target GA. We also prepared the test data set $S_{test}$ including two hundred pairs by calculating images of input data under the true heuristic function $G$ for each optimization problem.

As experts, we prepared mappings $G$, $F$ corresponding to twenty-four GAs for optimization problems shown in Table 4. These optimization problems were chosen to cover all orders of genotypes according to the fitness value. The numbers of four figures in Table 4 that indicate experts represent genotypes arranged according to the fitness value. Other settings for the experts are common to Table 3.

The experimental results 1–(i) and 1–(ii) of Table 5 and Table 6 show the mean squared error for $S_{test}$. We can see from Table 5 that both of two models could represent the true input-output relations well. The first six experts in order of the mixture rate are shown in Table 6. The numbers of four figures, which indicate experts, coincide with those in Table 6. The underlined expert corresponds to experts (Experiment 1).

<table>
<thead>
<tr>
<th>Fitness Function (genotype $\rightarrow$ fitness value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$ 0 $\rightarrow$ 4.1, 1 $\rightarrow$ 2.2, 2 $\rightarrow$ 1.3, 3 $\rightarrow$ 3.6</td>
</tr>
<tr>
<td>$f_2$ 0 $\rightarrow$ 3.6, 1 $\rightarrow$ 3.2, 2 $\rightarrow$ 1.3, 3 $\rightarrow$ 2.9</td>
</tr>
<tr>
<td>$f_3$ 0 $\rightarrow$ 3.6, 1 $\rightarrow$ 3.2, 2 $\rightarrow$ 2.9, 3 $\rightarrow$ 1.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Population Size</th>
<th>$n = 10$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operators (Parameters)</td>
<td>Proportional Selection ($p_x = 0.6$)</td>
</tr>
<tr>
<td>$f_1$</td>
<td>0.70 $\times$ 10$^{-3}$</td>
</tr>
<tr>
<td>$f_2$</td>
<td>0.32 $\times$ 10$^{-3}$</td>
</tr>
<tr>
<td>$f_3$</td>
<td>0.70 $\times$ 10$^{-3}$</td>
</tr>
</tbody>
</table>
to the optimization problem of same order of genotypes as the target optimization problem. We can see from Table 6 the following two tendencies with only a few exceptions:

- The underlined expert (that corresponds to the optimization problem of which order of genotypes is same as the target optimization problem) had large mixture rate.

- Experts were chosen to cover orders of genotypes of which fitness values are close to each other. For example, experts that consist of combinations of “03”, “30” and “12”, “21” were chosen for \( f_i \).

The experimental results 1–(ii) are shown in Table 7. Although the accuracy was worse than the results 1–(i), both models could represent the true input-output relations sufficiently well.

In the experiment 1–(i), we attached importance to the order of genotypes of the optimization problem that the GA solves. In both models, the way in which the expert playing the main role and those for fine adjustment were chosen reflected the characteristics of the target optimization problems. The order of genotypes is the sufficient information to know the optimal solution and the local optimal solution of the optimization problems of which fitness functions are unknown. We could extract it through the observation of the GA and the estimation of the heuristic function \( G \). This demonstrates the effectiveness of modeling by using mixture models.

On the other hand, the GA can also be represented without experts corresponding to similar order of genotypes. The experimental result 1–(ii) shows that we could obtain the models with the sufficient accuracy. This is important for learning at small cost and the simplification of the representation of GAs.

### 5.2. Experiment 2

We carried out an experiment for modeling GA that solves a knapsack problem \( k^* \) \((l = 5)\) shown in Table 8. The knapsack problem is a kind of combinatorial optimization problem. Let \( a_j \) be a binary variable assigned to the \( j \)-th item, which equals to 1 if the \( j \)-th item is stuffed in the knapsack and 0 otherwise. A fitness value of genotype \( k = \langle a_1a_2...a_l \rangle_2 \) is given by

\[
\text{fitness}(k) = \left\{ \begin{array}{ll}
\sum_{j=1}^{l} v_ja_j & \text{if } \sum_{j=1}^{l} w_ja_j \leq W \\
0 & \text{otherwise,}
\end{array} \right.
\]

where \( v_j \) and \( w_j \) denote the value and weight of the \( j \)-th item respectively. \( W \) denotes the weight limit, and \( (\cdot)_2 \) denotes the binary representation. Other settings of the target GA such as population size and operators are shown in Table 9.

As experts, we prepare mappings \( G^i, F^i \) corresponding to two GAs for the knapsack problems \( k_1, k_2 \) shown in Table 10. The weight of each item and the weight limit for \( k_1, k_2 \) are common to those shown in Table 8. The value of each item is set to satisfy \( v_j = 0.75v'_j + 0.25v''_j \), where \( v'_j, v''_j \) denote the value of the \( j \)-th item for \( k_1, k_2 \) respectively. Other settings for the experts are common to Table 9.

The experimental results 2 (the estimated mixture rate \( t = (t_1, t_2) \)) are shown in Table 11. As mentioned above, the \( G \)-mixture model and \( F \)-mixture model are regarded as “the mixture of GAs” and “the mixture of optimization problems” respectively. Furthermore, the experts used in the \( G \)-mixture model are under the same settings except for optimization problems. Therefore, both of them can be regarded as “the decomposition of knapsack problems” in this experiment. We can see from Table 11 that both models could estimate mixture

### Table 7: Experimental results 1–(ii) (the mean squared error).

<table>
<thead>
<tr>
<th>G-mixture model</th>
<th>F-mixture model</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_1 )</td>
<td>( 8.26 \times 10^{-3} )</td>
</tr>
<tr>
<td>( f_2 )</td>
<td>( 2.04 \times 10^{-3} )</td>
</tr>
<tr>
<td>( f_3 )</td>
<td>( 0.77 \times 10^{-3} )</td>
</tr>
</tbody>
</table>

### Table 8: A knapsack problem that the target GA solves (Experiment 2).

<table>
<thead>
<tr>
<th>Knapsack Problem</th>
<th>( k^* )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item</td>
<td>1 2 3 4 5</td>
</tr>
<tr>
<td>Value</td>
<td>15 9 21 42 18</td>
</tr>
<tr>
<td>Weight</td>
<td>12 14 6 26 13</td>
</tr>
<tr>
<td>Weight Limit</td>
<td>50</td>
</tr>
</tbody>
</table>

### Table 9: Parameter settings (Experiment 2).

<table>
<thead>
<tr>
<th>Population Size</th>
<th>( n = 100 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operators (Parameters)</td>
<td>Proportional Selection One-Point Crossover ( (p_c=0.6) ) Mutation ( (p_m=0.001) )</td>
</tr>
<tr>
<td>Data Size</td>
<td>( d = 500 )</td>
</tr>
<tr>
<td>Training Times</td>
<td>500</td>
</tr>
</tbody>
</table>

### Table 10: Knapsack problems corresponding to experts (Experiment 2).

<table>
<thead>
<tr>
<th>Knapsack Problem</th>
<th>( k_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item</td>
<td>1 2 3 4 5</td>
</tr>
<tr>
<td>Value</td>
<td>17 3 22 40 22</td>
</tr>
<tr>
<td>Weight</td>
<td>12 14 6 26 13</td>
</tr>
<tr>
<td>Weight Limit</td>
<td>50</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Knapsack Problem</th>
<th>( k_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item</td>
<td>1 2 3 4 5</td>
</tr>
<tr>
<td>Value</td>
<td>9 27 18 48 6</td>
</tr>
<tr>
<td>Weight</td>
<td>12 14 6 26 13</td>
</tr>
<tr>
<td>Weight Limit</td>
<td>50</td>
</tr>
</tbody>
</table>
Table 11: Experimental results 2 (the mixture rate).

<table>
<thead>
<tr>
<th>G-mixture model</th>
<th>F-mixture model</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0.763, 0.237)</td>
<td>(0.762, 0.238)</td>
</tr>
</tbody>
</table>

5.3. Discussions
In these experiments, the settings except optimization problems are common to the target GAs and the GAs for experts, in order to make the meaning of “the mixture of GAs (or optimization problems)” clear. In practice, when only the input-output data of GA’s population transition are given, we cannot know the settings of the target GAs. However, by using mixture models, we can analyze the characteristics of these unknown GAs, based on the known settings and characteristics of experts and the way of mixing them (i.e. the mixture rate). Even if some settings are not common, we can model the target GAs because the mixture of some different experts absorbs these differences. For example, suppose that the mutation rate \( p_m \) of only the target GA is extremely high. Then, the rate of genotypes that have lower fitness values may increase in the population. This behavior can be represented by using the expert for the optimization problem in which these genotypes have higher fitness values. That is, we can understand the effect of mutation rate on the basis of the known optimization problems.

6. Conclusions
In this paper, we have discussed modeling and analysis of GAs based on the viewpoint of mixture systems. We have regarded a GA as an information source that generates input-output data. That is, we have regarded a population and its next population generated by the GA as input and output respectively. Then we have modeled the GA by learning from these data. Since this method uses only the input-output relations of data and ignores interior structures, we can describe a variety of GAs in a common form, and analyze them from the new point of view. In this paper, we have used the two types of mixture models for a representation of these input-output relations, and investigated how these models are effective for analyzing GAs through the numerical experiments.

As a future work, applying our framework to the macroscopic modeling of GAs should be addressed. When genotype distributions are used as representations of populations, there is the problem that the dimension of distributions increases exponentially as the string length increases. In order to solve this problem, we can use more macroscopic distributions than genotype distributions as representations of populations. For example, van Nimwegen et al. have used fitness distributions as representations of populations and formulated transitions of them [9]. All the change needed for applying this representation to our framework is to transform a pair of populations into a pair of fitness distributions instead of genotype distributions at the step (i) in the procedure. We will study analyses of GAs by using the macroscopic representations and the mixture models in the future.

References