A New Method for Modeling the Behavior of Finite Population Evolutionary Algorithms

Tatsuya Motoki motoki@ie.niigata-u.ac.jp
Department of Information Engineering, Niigata University, Ikarashi 2-8050, Niigata 950-2181, Japan

Abstract

As practitioners we are interested in the likelihood of the population containing a copy of the optimum. The dynamic systems approach, however, does not help us to calculate that quantity. Markov chain analysis can be used in principle to calculate the quantity. However, since the associated transition matrices are enormous even for modest problems, it follows that in practice these calculations are usually computationally infeasible. Therefore, some improvements on this situation are desirable. In this paper, we present a method for modeling the behavior of finite population evolutionary algorithms (EAs), and show that if the population size is greater than 1 and much less than the cardinality of the search space, the resulting exact model requires considerably less memory space for theoretically running the stochastic search process of the original EA than the Nix and Vose-style Markov chain model. We also present some approximate models that use still less memory space than the exact model. Furthermore, based on our models, we examine the selection pressure by fitness-proportionate selection, and observe that on average over all population trajectories, there is no such strong bias toward selecting the higher fitness individuals as the fitness landscape suggests.

Keywords

Finite population evolutionary algorithms, Markov chain analysis, exact model, approximate model, fitness-proportionate selection, selection pressure, success probability.

1 Introduction

Although the development of evolutionary algorithm (EA) theory has been very slow in contrast with the rapid growth of the number of EA applications, it has been achieved steadily. We can now find several theoretical research topics on EA in books by Reeves and Rowe (2003), DeJong (2006), and Eiben and Smith (2003) or overview papers by Beyer et al. (2002), Eiben and Rudolph (1999), Bäck et al. (1997), and Whitley and Vose (1995). Nowadays, EAs are also discussed in the community of theoretical computer scientists. For example, Wegener (2000, 2001), Droste et al. (2002), and Wegener and Witt (2005) considered the problem of maximizing an unknown pseudo-Boolean function $f: \{0,1\}^n \to \Re$, and analyzed the behavior of the (1+1)EA on different classes of pseudo-Boolean functions through two complexity measures: the expected running time until some optimal solution is encountered, and the *success probability*, that is, the probability that some optimal solution is encountered within a given time limit. In the EA community, however, Markov chain analysis and the dynamic systems approach are considered to be prevalent approaches to the understanding of EAs at the microscopic level.

In the work on Markov chain analysis, researchers recognize that for their EAs, the sequence of successively produced populations is a Markov chain, and examine the fundamental properties of EAs by analyzing the associated Markov chain. For example, Goldberg and Segrest (1987) considered a single-locus, binary allele, crossover-less, finite population genetic algorithm (GA), determined the transition matrix of the Markov chain modeling that GA, and exactly calculated the expected time of first passage to "convergence" under different selection ratios and mutation rates. Based on qualitative Markov models, Eiben et al. (1991) characterized the limit behavior of EAs by the properties of the variation and selection operators. Nix and Vose (1992) showed how to construct the exact transition matrix for the Markov chain model of a simple generational GA that adopts a standard fixed-length binary string representation, fitness-proportionate selection, one-point crossover, and bit-flipping mutation, and investigated the asymptotic behavior as population size increases. Davis and Principe (1993) also modeled a simple GA as a Markov chain, and considered the asymptotic steady state distributions as the mutation rate decreased. Rudolph (1994) analyzed the convergence behavior of canonical GAs by means of Markov chain analysis. DeJong et al. (1995) explored the use of Markov chain analysis to model and understand the transient behavior of finite population GAs observed while in transition to steady states. Experimentally, they considered two-locus, binary allele, quintuple-sized population GAs, observed the effects that fitness functions, choice of operators, and other variables have on the transient behavior of GAs, and determined the expected waiting time until an optimum is first encountered through numerical calculation using the associated transition matrices. Aytug and Koehler (1996, 2000) derived bounds on the number of generations needed to see all populations or all strings (and hence, an optimal solution) with a specified probability, based on a Markov chain analysis. Schmitt et al. (1998) and Schmitt (2001, 2004) described a GA as a Markov chain on probability distributions over populations that are seen to be ordered tuples of individuals, separately applied spectral theory to the matrices describing mutation, crossover, or selection operations to isolate their key properties, and extensively discussed the limit probability distribution over populations. Poli et al. (2004) presented a Markov chain model for genetic programming and variable-length GAs with homologous crossover. Mitavskiy and Rowe (2006a, 2006b) described a generalized Geiringer theorem in terms of the limiting distribution of the associated Markov chain, and derived a schema-based version of the theorem for nonlinear genetic programming with homologous crossover.

In the work on the dynamic systems approach, researchers consider an EA to be a discrete dynamic system, and examine the dynamics of a population as it moves from generation to generation. For example, Vose and Liepins (1991) represented a population as a vector whose ith component is the proportion of the ith candidate solution in the population, and modeled simple GAs as dynamic systems where the combined effect of selection and recombination is formulated as a mathematical operator. Fixed points and their stability for this combined operator were investigated. Nix and Vose (1992) also used this mathematical operator to obtain a Markov chain model, and showed that the fixed points of this operator govern the way that a population will trace a trajectory as it evolves. Vose (1993, 1996, 1999a, 1999b), and Vose and Wright (1994) have studied these dynamic systems further. van Nimwegen et al. (1997, 1999) described the dynamics of crossover-less GAs in terms of flows in the space of fitness distributions, and showed how finite populations induce metastability. Arora et al. (1994) and Rabani et al. (1995) mentioned the connection between GAs and quadratic dynamic systems, which have been classically used to model various natural phenomena in physics and biology.

Note that, as explored by DeJong et al. (1995, p. 117) and Reeves and Rowe (2003, p. 280), practitioners are interested in the likelihood of the population at a given generation containing a copy of the optimum, and in how long it will take to find the optimum. The dynamic systems approach, however, does not help us to calculate these quantities, because this approach mainly examines the direction and intensity of the force of evolution acting on a population, and hence can hardly be used to take an average of any designated quantities over all possible population trajectories. Specifically, in Vose's (1999b) dynamic systems model, we represent a population as a vector, called a *population vector*, that records the proportion of each candidate solution in the population, and consider an operator $\mathcal{G}$ that given the current population vector p produces the probability distribution, according to which individuals in the next generation are sampled. Given the current population vector p, we can represent the expected next population vector as $\mathcal{G}(p)$. However, $\mathcal{G}^2(p)$ does not generally mean the average of all possible population vectors after two generations. On the other hand, as shown by DeJong et al. (1995), Markov chain analysis can be used in principle to calculate the above interesting quantities. However, since the associated transition matrices are enormous even for modest problems, it follows that in practice these calculations are usually computationally infeasible. Therefore, some improvements on this situation are desirable.

In this paper, we present a method for modeling the behavior of finite population EAs, and show that if the population size is greater than 1 and much less than the cardinality of the search space, the resulting exact model requires considerably less memory space for theoretically running the stochastic search process of the original EA than the Nix and Vose-style Markov chain model. We also present some approximate models that use still less memory space than the exact model. Furthermore, based on our models, we examine the selection pressure by fitness-proportionate selection, and observe that on the average over all population trajectories, there is no strong bias toward selecting the higher fitness individuals as the fitness landscape suggests.

The paper is organized as follows. In Section 2, we first provide a review of earlier relevant works on Markov chain analysis and the dynamic systems approach, and cover the difficulty in modeling finite population EAs. In Section 3, we restrict our attention to finite population crossover-less EAs, and present a method for modeling the behavior of such EAs; then we derive some approximate models from our exact model. In Section 4 we next extend the discussion so that EAs under consideration can include crossover. In Section 5 we examine the selection pressure by fitness-proportionate selection, and compare our models with the Nix and Vose-style Markov chain model on the basis of time and space complexities of procedures for theoretically running the stochastic search process through the models themselves. Finally, we summarize our work in Section 6.

2 Basic Concepts

2.1 Markov Chain Model

Evolutionary algorithms maintain and evolve a population of candidate solutions over time. Each EA iterates selection and reproduction operations to produce a sequence (trajectory) of populations in such a manner that satisfies the Markov property: every noninitial population of an EA only depends on the contents of the previous population in a probabilistic manner. So, if the search space is discrete and denumerable, the probabilistic behavior of an EA can essentially be described as a Markov chain. More formally, let $P^{(t)}$ be the population at some evolutionary time (e.g., generation) $t \in \{0, 1, 2, \ldots\}$,

let τ be a bijective mapping from the space of possible populations to $\{1, 2, 3, \ldots\}$, and let $X_t = \tau(P^{(t)})$. Then, the sequence of random variables $X_0, X_1, X_2, \ldots$ forms a Markov chain, because the EA only uses the current population to determine the next one and so the sequence satisfies the Markov property.

Specifically, Nix and Vose (1992) show how to construct the transition matrix for the Markov chain model of a simple generational GA that adopts a standard fixed-length binary string representation, fitness-proportionate selection, one-point crossover, and bit-flipping mutation. Let l be the length of the individual binary string, let $n = 2^l$ be the number of possible strings, and let r be the population size. Then the total number of possible (multiset) populations is given by $N = \binom{r+n-1}{n-1}$. They use an $n \times N$ matrix $Z = (z_{i,j})$ to describe an enumeration of possible populations $P_1, P_2, \ldots, P_N$. The ith column $\phi_i = \langle z_{0,i}, \ldots, z_{n\text{-}1,i} \rangle^T$ of Z represents the ith population P_i, since the (y, i)th entry $z_{y,i}$ of Z is defined to be the number of occurrences of binary string y in P_i, where integer y is unambiguously interpreted as its binary representation depending on the context. The matrix Z naturally defines a bijection $\tau : \{P_1, P_2, \ldots, P_N\} \to \{1, 2, \ldots, N\}$ by $\tau(P_i) = i$. So, vectors $\phi_1, \phi_2, \ldots, \phi_N$ essentially represent the states of the Markov chain. Nix and Vose (1992) precisely describe the (i, j)th entry $Q_{i,j}$ of the transition matrix for the Markov chain as

$$Q_{i,j} = r! \prod_{y=0}^{n-1} \frac{(\{\mathcal{M}(\mathcal{F}(\phi_i))\}_y)^{z_{y,j}}}{z_{y,j}!}, \tag{1}$$

where $\mathcal{F}$ is a mathematical operator that plays the effect of selection, $\mathcal{M}$ is a mathematical operator that plays the combined effect of crossover and mutation operations, and the notation $\{\cdot\}_y$ is used to denote the yth component of a vector. Operators $\mathcal{F}$ and $\mathcal{M}$ are transformations over *population vectors*, that is, vectors $\langle x_0, \ldots, x_{n\text{-}1} \rangle^T$ representing a particular population in such a manner that x_y is the proportion of string y in the population. For fitness-proportionate selection,

$$\mathcal{F}(x) = \frac{\mathrm{diag}(f(0), \ldots, f(n-1))x}{\langle f(0), \ldots, f(n-1) \rangle x} \tag{2}$$

where $f(y)$ is the fitness of string y and $\mathrm{diag}(f(0), \ldots, f(n-1))$ is the diagonal matrix with (i, i)th entry $f(i)$. Let $m_{i,j}(y)$ be the probability that string y results from the recombination process based on parent strings i and j, and let M_y be an $n \times n$ matrix having (i, j)th entry $m_{i,j}(y)$. Then

$$\mathcal{M}(x) = \langle x^T M_0 x, x^T M_1 x, \ldots, x^T M_{n\text{-}1} x \rangle^T. \tag{3}$$

Generally, the (i, j)th entry $Q_{i,j}$ of the transition matrix Q for an EA specifies the probability that the EA in state ϕ_i will be in state ϕ_j in the next generation. Thus, calculating the powers of Q yields the probabilities of state transitions for multiple generation changes; in fact, $Q^k_{i,j}$ is the probability that the EA in state ϕ_i will be in state ϕ_j after k generations. Unfortunately, since the number of possible populations N grows rapidly with population size r and search space size n, the Q matrices for realistic EA systems are too large to calculate values of Q^k. For example, when $r = 10$ and $n = 128$, we have $N = \binom{10+128-1}{128-1} \approx 4.59 \times 10^{14}$ and so the matrix Q has $N^2 \approx 2.10 \times 10^{29}$ entries. In order to construct and manipulate the transition matrix Q, we also need an effective enumeration of possible populations. Moreover, even if we could calculate values of Q^k, we cannot easily conceive interrelation between possible individuals.

2.2 Dynamic Systems Model (Vose's Infinite Population Model)

We can consider an EA to be a discrete dynamic system, and examine the dynamics of a population as it moves from generation to generation. In this approach, a population is represented as a population vector that records the proportion of each candidate solution in the population, and the behavior of an EA is viewed as a trajectory of population vectors over the set

$$\Lambda = \left\{ x \in \Re^n \mid x_k \geq 0 \text{ for all } k \text{ and } \sum_{k=1}^{n} x_k = 1 \right\} \tag{4}$$

called the *simplex*, where $\Re$ is the set of real numbers and n is the size of the search space.

Vose and Wright (1994) consider a generational operator $\mathcal{G} : \Lambda \to \Lambda$ that given the current population vector p produces the probability distribution, according to which individuals in the next generation are sampled. To construct the next population from the current population vector p, the distribution $\mathcal{G}(p)$ is sampled multinomially so that if the next population size is r, the probability that q is the next population vector after p is given by

$$r! \prod_{j=0}^{n-1} \frac{(\{\mathcal{G}(p)\}_j)^{rq_j}}{(rq_j)!}. \tag{5}$$

If the population size is infinite, we see that the process of constructing a next population becomes deterministic, and so the trajectory of populations is uniquely given by $p, \mathcal{G}(p), \mathcal{G}^2(p), \ldots$. In general, however, the next population vector cannot be uniquely determined from the current one. Vose and Wright (1994) show that $\mathcal{G}(p)$ only gives the expected population vector over all possible next generations. On the other hand, Nix and Vose (1992) show that with probability arbitrarily close to 1, population trajectories converge to iterations of $\mathcal{G}$ as population size grows. So, the generational operator $\mathcal{G}$ governs the way that a population will trace a trajectory on Λ. Fixed points of $\mathcal{G}$ are thus examined to understand the asymptotic behavior of EA.

Markov chain models capture every aspect of EA behavior, but its formalization is complex and its use in calculating multistep transition probabilities is not practical without enormous computational space. Dynamic systems models, on the other hand, collapse all aspects of EA behavior into a single population trajectory $p, \mathcal{G}(p), \mathcal{G}^2(p), \ldots$ which encodes the asymptotic behavior of EA and cannot be used to predict the transient behavior of finite population EAs. For any current population vector p, the operator $\mathcal{G}$ certainly produces the expected next population vector $\mathcal{G}(p)$. Under conditions of finite population, however, this does not imply that $p, \mathcal{G}(p), \mathcal{G}^2(p), \ldots$ is the average trajectory over all actually possible trajectories. Thus, $\{\mathcal{G}^k(p)\}_i$ does not necessarily mean the probability of the ith element in the search space will occur in the population after k generations. The dynamic systems model does not give us any probabilistic information. In order to estimate the success probability at each generation, we need to trace (implicitly) all possible trajectories and average successibility over them.

Prügel-Bennett (2003) shows difficulty in using only population vectors to model the evolution of EAs.

2.3 Difficulty in Modeling Finite Population EAs

This section shows that for finite population EAs with fitness-proportionate selection, we cannot easily derive the expected proportion of each candidate solution in the next

population from that in the current population. Let n be the size of the search space, and let f_i be the fitness of type-i individual. Consider an evolutionary search process that proceeds as follows:

$\mathcal{P}^{(0)} \leftarrow$ an initial population of r individuals;
$t \leftarrow 0$;
repeat {
  $\mathcal{Q}^{(t)} \leftarrow$ a multiset of r individuals that are selected from $\mathcal{P}^{(t)}$ through fitness-proportionate scheme;
  $\mathcal{P}^{(t+1)} \leftarrow$ a population of r individuals that are obtained from elements in $\mathcal{Q}^{(t)}$ by mutation;
  $t \leftarrow t+1$;
}

Let $X_i^{(t)}$ be the proportion of type-i individuals in the population $\mathcal{P}^{(t)}$, and let $Y_i^{(t)}$ be the proportion of type-i individuals in $\mathcal{Q}^{(t)}$. Note that $X_i^{(t)}$ and $Y_i^{(t)}$ are both random variables. We want to obtain recurrence relations for tracing the sequence of population vectors $\langle E[X_1^{(t)}], \ldots, E[X_n^{(t)}]\rangle^T$ and $\langle E[Y_1^{(t)}], \ldots, E[Y_n^{(t)}]\rangle^T$.

THEOREM 2.1 (Effect of Mutation): *Let $m_{i,j}$ be the probability that type-i individual results from the mutation based on type-j parent individual, and let M be an $n \times n$ matrix having (i,j)th entry $m_{i,j}$. Then, for every generation t,*

$$\begin{bmatrix} E\big[X_1^{(t+1)}\big] \\ \vdots \\ E\big[X_n^{(t+1)}\big] \end{bmatrix} = M \begin{bmatrix} E\big[Y_1^{(t)}\big] \\ \vdots \\ E\big[Y_n^{(t)}\big] \end{bmatrix}.$$

PROOF: Let Λ_0 be the range of $\langle Y_1^{(t)}, \ldots, Y_n^{(t)}\rangle^T$. When Λ_0 is finite, we can calculate $E[X_i^{(t+1)}]$ as follows:

$$\begin{aligned}
&E\big[X_i^{(t+1)}\big] \\
&= \sum_{\langle p_1,\ldots,p_n\rangle^T \in \Lambda_0} E\big[X_i^{(t+1)} | Y_1^{(t)} = p_1, \ldots, Y_n^{(t)} = p_n\big] \Pr\big\{Y_1^{(t)} = p_1, \ldots, Y_n^{(t)} = p_n\big\} \\
&= \sum_{\langle p_1,\ldots,p_n\rangle^T \in \Lambda_0} \left\{ \sum_{j=1}^{n} p_j m_{i,j} \right\} \Pr\big\{Y_1^{(t)} = p_1, \ldots, Y_n^{(t)} = p_n\big\} \\
&= \sum_{j=1}^{n} m_{i,j} \sum_{\langle p_1,\ldots,p_n\rangle^T \in \Lambda_0} p_j \Pr\big\{Y_1^{(t)} = p_1, \ldots, Y_n^{(t)} = p_n\big\} \\
&= \sum_{j=1}^{n} m_{i,j} \sum_{p_j : \text{possible value of } Y_j^{(t)}} p_j \Pr\big\{Y_j^{(t)} = p_j\big\} \\
&= \sum_{j=1}^{n} m_{i,j} E\big[Y_j^{(t)}\big]
\end{aligned}$$

When Λ_0 is infinite, we can also carry out a similar calculation. □

The above theorem shows that the effect of mutation on population vectors is represented as a linear transformation. On the other hand, the effect of fitness-proportionate selection, as shown below, cannot be represented in such a simple way when the population is finite.

THEOREM 2.2 (Effect of Fitness-Proportionate Selection): *Let* Λ_0 *be the range of* $\langle X_1^{(t)}, \ldots, X_n^{(t)}\rangle^T$. *If* Λ_0 *is finite, then for every type* i *of individual and every generation* t,

$$E[Y_i^{(t)}] = \sum_{\langle p_1,\ldots,p_n\rangle^T \in \Lambda_0} \frac{f_i p_i}{\sum_{j=1}^n f_j p_j} Pr\{X_1^{(t)} = p_1, \ldots, X_n^{(t)} = p_n\}.$$

PROOF: In fitness-proportionate selection, the probability of a type-i individual being selected is given by

$$\frac{f_i \times (\text{the proportion of type-}i\text{ individual in the population})}{\sum_{j=1}^n f_j \times (\text{the proportion of type-}j\text{ individual in the population})}.$$

Therefore, if Λ_0 is finite,

$$\begin{aligned} E\big[Y_i^{(t)}\big] &= \sum_{\langle p_1,\ldots,p_n\rangle^T \in \Lambda_0} E\big[Y_i^{(t)} \mid X_1^{(t)} = p_1, \ldots, X_n^{(t)} = p_n\big] \Pr\big\{X_1^{(t)} = p_1, \ldots, X_n^{(t)} = p_n\big\} \\ &= \sum_{\langle p_1,\ldots,p_n\rangle^T \in \Lambda_0} \frac{f_i p_i}{\sum_{j=1}^n f_j p_j} \Pr\big\{X_1^{(t)} = p_1, \ldots, X_n^{(t)} = p_n\big\}. \end{aligned}$$

□

COROLLARY 2.3 (Effect of Fitness-Proportionate Selection on Infinite Population): *If the population is infinite and* $Pr\{X_1^{(0)} = E[X_1^{(0)}], \ldots, X_n^{(0)} = E[X_n^{(0)}]\} = 1$, *then for every generation* t

$$\begin{bmatrix} E\big[Y_1^{(t)}\big] \\ \vdots \\ E\big[Y_n^{(t)}\big] \end{bmatrix} = \frac{diag(f_1, \ldots, f_n)}{\sum_{j=1}^n f_j E[X_j^{(t)}]} \begin{bmatrix} E\big[X_1^{(t)}\big] \\ \vdots \\ E[X_n^{(t)}] \end{bmatrix}.$$

PROOF: If $\Pr\{X_1^{(t)} = E[X_1^{(t)}], \ldots, X_n^{(t)} = E[X_n^{(t)}]\} = 1$, then the range of $\langle X_1^{(t)}, \ldots, X_n^{(t)}\rangle^T$ becomes $\{\langle E[X_1^{(t)}], \ldots, E[X_n^{(t)}]\rangle^T\}$, and so it follows from Theorem 2.2 that

$$\begin{aligned} E\big[Y_i^{(t)}\big] &= \sum_{\langle p_1,\ldots,p_n\rangle^T \in \Lambda_0} \frac{f_i p_i}{\sum_{j=1}^n f_j p_j} \Pr\big\{X_1^{(t)} = p_1, \ldots, X_n^{(t)} = p_n\big\} \\ &= \frac{f_i E[X_i^{(t)}]}{\sum_{j=1}^n f_j E[X_j^{(t)}]}. \end{aligned}$$

Therefore, we have the following assertion:

$$\begin{vmatrix} \text{if } \Pr\big\{X_1^{(t)} = E\big[X_1^{(t)}\big], \ldots, X_n^{(t)} = E\big[X_n^{(t)}\big]\big\} = 1, \text{ then} \\ \big\langle E\big[Y_1^{(t)}\big], \ldots, E\big[Y_n^{(t)}\big]\big\rangle^T = \frac{\text{diag}(f_1,\ldots,f_n)}{\sum_{j=1}^n f_j E\big[X_j^{(t)}\big]} \big\langle E\big[X_1^{(t)}\big], \ldots, E\big[X_n^{(t)}\big]\big\rangle^T. \end{vmatrix} \quad (6)$$

Now, note that under conditions of infinite population, we see from the law of large numbers that $X_i^{(t+1)}$'s are uniquely determined from $Y_i^{(t)}$'s and that $Y_i^{(t)}$'s are uniquely determined from $X_i^{(t)}$'s. Thus, if the population is infinite and $\Pr\{X_1^{(0)} = E[X_1^{(0)}], \ldots, X_n^{(0)} = E[X_n^{(0)}]\} = 1$, then we can see that for every t $\Pr\{X_1^{(t)} = E[X_1^{(t)}], \ldots, X_n^{(t)} = E[X_n^{(t)}]\} = \Pr\{Y_1^{(t)} = E[Y_1^{(t)}], \ldots, Y_n^{(t)} = E[Y_n^{(t)}]\} = 1$, and hence we have from the above assertion (6) that for every generation t $\langle E[Y_1^{(t)}], \ldots, E[Y_n^{(t)}]\rangle^T = \frac{\mathrm{diag}(f_1, \ldots, f_n)}{\sum_{j=1}^n f_j E[X_j^{(t)}]} \langle E[X_1^{(t)}], \ldots, E[X_n^{(t)}]\rangle^T$. □

If the population is infinite and $\Pr\{X_1^{(0)} = E[X_1^{(0)}], \ldots, X_n^{(0)} = E[X_n^{(0)}]\} = 1$, we see from Corollary 2.3 that the effect of fitness-proportionate selection on $E[Y_i^{(t)}]$'s is determined from the values of $E[X_1^{(t)}], \ldots, E[X_n^{(t)}]$. If the population is finite, however, we have difficulty in representing the effect of fitness-proportionate selection on $E[Y_i^{(t)}]$'s because we see from Theorem 2.2 that $E[Y_i^{(t)}]$ depends not only on $E[X_i^{(t)}]$'s but also on the joint distribution of $X_1^{(t)}, \ldots, X_n^{(t)}$.

EXAMPLE 2.4 (Influence of the Joint Distribution of $X_i^{(t)}$ on $E[Y_i^{(t)}]$): Consider two cases:

$$(a)\quad \Pr\left\{X_1^{(t)} = p_1, \ldots, X_n^{(t)} = p_n\right\} = \begin{cases} 1/2 & \text{if } p_1 = 1,\ p_2 = \cdots = p_n = 0 \\ 1/3 & \text{if } p_1 = 0,\ p_2 = 1,\ p_3 = \cdots = p_n = 0 \\ 1/6 & \text{if } p_1 = \cdots = p_{n\text{-}1} = 0,\ p_n = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$(b)\quad \Pr\left\{X_1^{(t)} = p_1, \ldots, X_n^{(t)} = p_n\right\} = \begin{cases} 1 & \text{if } p_1 = 1/2,\ p_2 = 1/3,\ p_3 = \cdots = p_{n\text{-}1} = 0,\ p_n = 1/6 \\ 0 & \text{otherwise} \end{cases}$$

In both cases $E[X_1^{(t)}] = 1/2$, $E[X_2^{(t)}] = 1/3$, $E[X_3^{(t)}] = \cdots = E[X_{n-1}^{(t)}] = 0$, $E[X_n^{(t)}] = 1/6$. From Theorem 2.2, however, we see that two cases have different consequences on $\langle E[Y_1^{(t)}], \ldots, E[Y_n^{(t)}]\rangle^T$; in case (a) we can easily obtain $\langle E[Y_1^{(t)}], \ldots, E[Y_n^{(t)}]\rangle^T = \langle 1/2, 1/3, 0, \ldots, 0, 1/6\rangle^T$, while in case (b) we have $\langle E[Y_1^{(t)}], \ldots, E[Y_n^{(t)}]\rangle^T = \frac{1}{3f_1+2f_2+f_n}\langle 3f_1, 2f_2, 0, \ldots, 0, f_n\rangle^T$.

3 Modeling the Behavior of Finite Population Crossoverless EA

The Markov chain model describes the behavior of a finite population EA in a probabilistically exact way. In this section, we give another such model for a finite population crossoverless EA, and then derive some approximate models from our exact model. Our exact model can be seen as a variant of Markov chain model.

Each Markov chain model has possible populations in an EA as states, and uses a linear transformation to obtain the probability distribution over all possible next populations. In this section, on the other hand, we pay attention to each member of population rather than the whole population. Each population is now represented as an ordered tuple of candidate solutions, as in the papers of Rudolph (1994), Schmitt et al.

(1998), Schmitt (2001, 2004), or Mitavskiy and Rowe (2006a, 2006b). Each position in an ordered tuple representing a population is occupied by a member of the population, and is identified by a coordinate number, called an *address*, between 1 and the population size. We will consider quantities such as the (conditional) probability of the individual at address i exhibiting a particular candidate solution at generation t, and give update rules that indicate how these quantities change over time. These quantities will be used to express probability distributions over ordered-tuple populations.

REMARK 3.1 (Methods for Representing Populations): Suppose, for example, that our search space is {a, b}, and that the population size is fixed at 3. Then we represent possible populations as eight ordered tuples (a,a,a), (a,a,b), (a,b,a), (b,a,a), (a,b,b), (b,a,b), (b,b,a), and (b,b,b), and turn our minds to probability distributions over these ordered-tuple populations, while the Nix and Vose-style Markov chain model represents possible populations as four multisets {a,a,a}, {a,a,b}, {a,b,b}, and {b,b,b}, and turns our minds to probability distributions over these multiset populations. Obviously, we can easily convert a probability distribution over ordered-tuple populations into a distribution over multiset populations through equations such as Pr{population is {a,a,b}} = Pr{population is (a,a,b)} + Pr{population is (a,b,a)} + Pr{population is (b,a,a)}, although we cannot uniquely determine the opposite conversion.

In this section, our discussion assumes the following crossoverless EA:

$\mathcal{P}^{(0)} \leftarrow$ an initial ordered-tuple population of r individuals;
$t \leftarrow 0$;
repeat {
  $\mathcal{Q}^{(t)} \leftarrow$ an ordered-tuple of r individuals that are selected from $\mathcal{P}^{(t)}$ through fitness-proportionate scheme;
  $\mathcal{P}^{(t+1)} \leftarrow$ an ordered-tuple population of r individuals that are obtained from elements in $\mathcal{Q}^{(t)}$ by mutation;
  $t \leftarrow t + 1$;
}

Fitness-proportionate selection and mutation are assumed to be alternately applied. No crossover is assumed. Since we do not specify any stopping rule, it follows that the search process runs forever. In order to observe the behavior of EAs and easily cope with the cumulative feature of the success probability, we now consider two kinds of random variables $P^{(t)}$ and $Q^{(t)}$ defined as follows:

$$P^{(t)} = \begin{cases} \mathcal{P}^{(t)} & \text{if } t = 0 \text{ or } P^{(t\text{-}1)} \text{ does not contain any optimal solution} \\ \text{OPT} & \text{otherwise,} \end{cases}$$

$$Q^{(t)} = \begin{cases} \mathcal{Q}^{(t)} & \text{if } P^{(t)} \text{ does not contain any optimal solution} \\ \text{OPT} & \text{otherwise,} \end{cases}$$

where we use OPT to denote a population that only contains multiple copies of some distinguished optimal solution. Next, suppose that the search space has n elements, called type-1 through type-n candidate solutions, and that the type-n solution is the unique optimum constituting OPT. Let r be the population size, let f_i be the fitness of

type-i solution, let

$$m_{i,j} = \begin{cases} \Pr\{\text{type-}j \text{ solution mutates to type-}i\} & \text{if } j \neq n \\ 0 & \text{if } j = n, i \neq n \\ 1 & \text{if } j = n, i = n, \end{cases}$$

and let M be an $n \times n$ matrix having (i, j)th entry $m_{i,j}$.

Note that $P^{(t)}$ and $Q^{(t)}$ are new random variables introduced for discussing the behavior of the above EAs, and their instances are not maintained by the EAs. After the EA described in the above pseudocode finds an optimal solution, the population $P^{(t)}$ (and $Q^{(t)}$) will fill up with the optimum; so OPT behaves as an absorbing state in the associated Markov chain. If we need to implement the evolution of the virtual population $P^{(t)}$, it suffices to adopt some special form of selection that gives complete preference to the optimum, and some special form of mutation that persists in the optimum; for instance (fitness-proportionate) selection after reassigning infinite fitness value to the optimum, and mutation that rejects departure from the optimum. Now, we can easily see that $\Pr\{P^{(t)} \text{ contains an optimum solution}\}$ describes the success probability at generation t. We will, therefore, keep in mind a Markov chain that models the stochastic process $\{P^{(t)} | \ t \geq 0\}$, and consider how to observe the transition of the probability distribution over populations $P^{(t)}$.

To understand the basic idea quickly, we first give results for the special cases where the population size equals 2, and then generalize the results.

3.1 When the Population Size = 2

In this section, we first consider the special cases where the population size equals 2. Concerning the virtual population $P^{(t)}$, let $x_i^{(t)}$ be the probability of the individual at address 1 exhibiting the type-i solution, and let $q_i^{(t)}(k)$ be the conditional probability of the individual at address 2 being type-k given the individual at address 1 being type-i. Then, it follows that the probability distribution over ordered-tuple populations $P^{(t)}$ is determined by

$$\Pr\left\{P^{(t)} = (\text{type-}i \text{ individual}, \ \text{type-}j \text{ individual})\right\} = x_i^{(t)} q_i^{(t)}(j).$$

We will give update rules that indicate how $(n+1)$ probability distributions $x^{(t)} = \langle x_1^{(t)}, \ldots, x_n^{(t)} \rangle^T$ and $q_i^{(t)} = \langle q_i^{(t)}(1), \ldots, q_i^{(t)}(n) \rangle^T$, where $i \in \{1, 2, \ldots, n\}$, change over time. Before this we give a theorem that shows the combined effect of fitness-proportionate selection and mutation on production of next-generation individuals.

THEOREM 3.2 (Crossoverless EA, Population Size = 2): *Consider a finite population crossoverless EA, and suppose that the population size equals 2. Then,*

(a) $\Pr\{$*the individual at address 1 in* $P^{(t+1)}$ *is type-a*$\}$

$$= \sum_{i \neq n} \sum_{j \neq n} x_i^{(t)} q_i^{(t)}(j) \frac{f_i m_{a,i} + f_j m_{a,j}}{f_i + f_j} + \sum_{i=n \text{ or } j=n} \sum x_i^{(t)} q_i^{(t)}(j) m_{a,n}.$$

(b) $\Pr\{$*the individuals at address 1 and 2 in* $P^{(t+1)}$ *are type-a and type-b respectively*$\}$

$$= \sum_{i \neq n} \sum_{j \neq n} x_i^{(t)} q_i^{(t)}(j) \frac{f_i m_{a,i} + f_j m_{a,j}}{f_i + f_j} \frac{f_i m_{b,i} + f_j m_{b,j}}{f_i + f_j} + \sum_{i=n \text{ or } j=n} \sum x_i^{(t)} q_i^{(t)}(j) m_{a,n} m_{b,n}.$$

PROOF: We use a notation (type-i, type-j) to denote a population that contains a type-i individual at address 1 and a type-j individual at address 2. We use $\mathcal{P}^{(t)}$ and $\mathcal{Q}^{(t)}$ described in our EA procedure to denote the population and the mating pool at generation t, respectively. We also use $Q^{(t)}$ defined after our EA procedure to denote the virtual mating pool at generation t.

(a) Since $\Pr\{P^{(t)} = \text{(type-}i\text{, type-}j\text{)}\} = x_i^{(t)} q_i^{(t)}(j)$ for every type numbers i and j, we may calculate as follows:

$$\begin{aligned}
&\Pr\left\{P^{(t+1)} = \text{(type-}a\text{,type-}b\text{) for some } b\right\} \\
&\quad = \sum_i \sum_j \Pr\left\{P^{(t)} = \text{(type-}i\text{,type-}j\text{)}\right\} \Pr\left\{P^{(t+1)} = \text{(type-}a\text{,type-}b\text{)} \right. \\
&\qquad\qquad \left. \text{for some } b \mid P^{(t)} = \text{(type-}i\text{,type-}j\text{)}\right\} \\
&\quad = \sum_i \sum_j x_i^{(t)} q_i^{(t)}(j) \Pr\left\{P^{(t+1)} = \text{(type-}a\text{,type-}b\text{)} \right. \\
&\qquad\qquad \left. \text{for some } b \mid P^{(t)} = \text{(type-}i\text{,type-}j\text{)}\right\} \qquad (7)
\end{aligned}$$

In order to proceed with the calculation, suppose that $P^{(t)} = \text{(type-}i\text{,type-}j\text{)}$. Then, we can observe from the choice of "type-n" that $P^{(t)}$ contains the optimum if and only if $i = n$ or $j = n$, and so we see that the subsequent evolutionary process proceeds in two ways, depending on whether the disjunctive condition "$i = n$ or $j = n$" is satisfied. If $i = n$ or $j = n$, the population $P^{(t)}$ contains the optimum, and so it follows from the definitions of $P^{(t)}$ and $Q^{(t)}$ that $Q^{(t)} = P^{(t+1)} = \text{OPT} = \text{(type-}n\text{,type-}n\text{)}$. Therefore,

$$\begin{aligned}
&\Pr\left\{P^{(t+1)} = \text{(type-}a\text{, type-}b\text{) for some } b \,\middle|\, P^{(t)} = \text{(type-}i\text{, type-}j\text{)}, (i = n \text{ or } j = n)\right\} \\
&\quad = \begin{cases} 1 & \text{if } a = n \\ 0 & \text{otherwise} \end{cases} \\
&\quad = m_{a,n}. \qquad (8)
\end{aligned}$$

On the other hand, if $i \neq n$ and $j \neq n$, the population $P^{(t)}(= \mathcal{P}^{(t)}) = \text{(type-}i\text{,type-}j\text{)}$ does not contain the optimum, and so through fitness-proportionate selection

$$\begin{cases} \text{type-}i \text{ individual will be selected with probability } \dfrac{f_i}{f_i + f_j}, \text{ and} \\ \text{type-}j \text{ individual will be selected with probability } \dfrac{f_j}{f_i + f_j} \end{cases}$$

for an individual at address 1 in $\mathcal{Q}^{(t)}(=Q^{(t)})$. If the selected individual is type-i it mutates to type-a with probability $m_{a,i}$, and if the selected individual is type-j it mutates to type-a with probability $m_{a,j}$. Since the obtained mutant is automatically adopted as the

individual at address 1 in $\mathcal{P}^{(t+1)}(= P^{(t)})$, it follows that

$$\begin{aligned}&\Pr\{P^{(t+1)} = \text{(type-}a\text{,type-}b\text{) for some } b \mid P^{(t)} = \text{(type-}i\text{,type-}j\text{)}, i\neq n,\ j\neq n\}\\&= \frac{f_i}{f_i+f_j}m_{a,i} + \frac{f_j}{f_i+f_j}m_{a,j}.\end{aligned} \tag{9}$$

By substituting Equations (8) and (9) into Equation (7), we can obtain the required result.

(b) We can proceed as in the proof of (a). That is, we start a calculation as follows:

$$\begin{aligned}&\Pr\{P^{(t+1)} = \text{(type-}a\text{,type-}b\text{)}\}\\&= \sum_i \sum_j \Pr\{P^{(t)} = \text{(type-}i\text{,type-}j\text{)}\}\Pr\{P^{(t+1)} = \text{(type-}a\text{,type-}b\text{)} \mid \\&\qquad\qquad P^{(t)} = \text{(type-}i\text{,type-}j\text{)}\}\\&= \sum_i \sum_j x_i^{(t)} q_i^{(t)}(j)\Pr\{P^{(t+1)} = \text{(type-}a\text{,type-}b\text{)} \mid P^{(t)} = \text{(type-}i\text{,type-}j\text{)}\}\end{aligned} \tag{10}$$

To proceed with the calculation, suppose that $P^{(t)} = \text{(type-}i\text{,type-}j\text{)}$. If $i = n$ or $j = n$, it also follows that $Q^{(t)} = P^{(t+1)} = \text{OPT} = \text{(type-}n\text{,type-}n\text{)}$. Therefore,

$$\begin{aligned}&\Pr\{P^{(t+1)} = \text{(type-}a\text{,type-}b\text{)} \mid P^{(t)} = \text{(type-}i\text{,type-}j\text{)}, (i = n \text{ or } j = n)\}\\&= \begin{cases}1 & \text{if } a = b = n\\ 0 & \text{otherwise}\end{cases}\\&= m_{a,n}m_{b,n}.\end{aligned} \tag{11}$$

On the other hand, if $i \neq n$ and $j \neq n$, then in the same way as in (a) we also have

$$\begin{aligned}&\Pr\{P^{(t+1)} = \text{(type-}a\text{,type-}b\text{) for some } b \mid P^{(t)} = \text{(type-}i\text{,type-}j\text{)}, i \neq n,\ j \neq n\}\\&= \frac{f_i}{f_i+f_j}m_{a,i} + \frac{f_j}{f_i+f_j}m_{a,j}.\end{aligned}$$

Likewise,

$$\begin{aligned}&\Pr\{P^{(t+1)} = \text{(type-}a\text{,type-}b\text{) for some } a \mid P^{(t)} = \text{(type-}i\text{,type-}j\text{)}, i \neq n,\ j \neq n\}\\&= \frac{f_i}{f_i+f_j}m_{b,i} + \frac{f_j}{f_i+f_j}m_{b,j}.\end{aligned}$$

Since individuals in $P^{(t+1)}$ are independently produced from $P^{(t)}$, it follows that

$$\begin{aligned}
&\Pr\{P^{(t+1)} = \text{(type-}a\text{,type-}b\text{)} \mid P^{(t)} = \text{(type-}i\text{,type-}j\text{)}, i \neq n, j \neq n\} \\
&\quad = \Pr\{P^{(t+1)} = \text{(type-}a\text{,type-}b\text{) for some } b \mid P^{(t)} = \text{(type-}i\text{,type-}j\text{)}, i \neq n, j \neq n\} \\
&\qquad \times \Pr\{P^{(t+1)} = \text{(type-}a\text{,type-}b\text{) for some } a \mid P^{(t)} = \text{(type-}i\text{,type-}j\text{)}, i \neq n, j \neq n\} \\
&\quad = \left(\frac{f_i m_{a,i} + f_j m_{a,j}}{f_i + f_j}\right)\left(\frac{f_i m_{b,i} + f_j m_{b,j}}{f_i + f_j}\right). \qquad (12)
\end{aligned}$$

By substituting Equations (11) and (12) into Equation (10), we can obtain the required result. □

COROLLARY 3.3 (Crossoverless EA, Population Size = 2): *Consider a finite population crossoverless EA, and suppose that the population size equals 2. Then, for every noninitial generation t and type numbers a and b,*

$$x_a^{(t)} q_a^{(t)}(b) = x_b^{(t)} q_b^{(t)}(a).$$

PROOF: For every generation $t \geq 0$ and type numbers a and b, it follows from Theorem 3.2 (b) that

$$\begin{aligned}
&x_a^{(t+1)} q_a^{(t+1)}(b) \\
&\quad = \Pr\{P^{(t+1)} = \text{(type-}a\text{,type-}b\text{)}\} \\
&\quad = \sum_{i \neq n} \sum_{j \neq n} x_i^{(t)} q_i^{(t)}(j) \frac{f_i m_{a,i} + f_j m_{a,j}}{f_i + f_j} \frac{f_i m_{b,i} + f_j m_{b,j}}{f_i + f_j} + \sum_{i=n \text{ or } j=n} \sum x_i^{(t)} q_i^{(t)}(j) m_{a,n} m_{b,n} \\
&\quad = \Pr\{P^{(t+1)} = \text{(type-}b\text{,type-}a\text{)}\} \\
&\quad = x_b^{(t+1)} q_b^{(t+1)}(a).
\end{aligned}$$

□

Based on Theorem 3.2, we now give the update rules that govern how probability distributions $x^{(t)} = \langle x_1^{(t)}, \ldots, x_n^{(t)} \rangle^T$ and $q_i^{(t)} = \langle q_i^{(t)}(1), \ldots, q_i^{(t)}(n) \rangle^T$, where $i \in \{1, 2, \ldots, n\}$, change over time.

Update Rule for $x^{(t)} = \langle x_1^{(t)}, \ldots, x_n^{(t)} \rangle^T$ to Change: When the population size equals 2, from Theorem 3.2(a) we have a rule

$$\begin{aligned}
x_a^{(t+1)} &= \Pr\{\text{the individual at address 1 is type-}a \text{ at generation } t+1\} \\
&= \sum_{i \neq n} \sum_{j \neq n} x_i^{(t)} q_i^{(t)}(j) \frac{f_i m_{a,i} + f_j m_{a,j}}{f_i + f_j} + \sum_{i=n \text{ or } j=n} \sum x_i^{(t)} q_i^{(t)}(j) m_{a,n}. \qquad (13)
\end{aligned}$$

In order to show the specific effects of fitness-proportionate selection and mutation separately, let $y_k^{(t)}$ be the probability of the type-k solution occurring at address 1 in $Q^{(t)}$.

Then, we can calculate $y_k^{(t)}$ as in the proof of Theorem 3.2(a):

$$
\begin{aligned}
y_k^{(t)} &= \sum_i \sum_j \Pr\{P^{(t)} = (\text{type-}i,\text{type-}j)\}\Pr\{\text{type-}k \text{ solution occurs} \\
&\qquad\qquad \text{at address 1 in } Q^{(t)} \mid P^{(t)} = (\text{type-}i,\text{type-}j)\} \\
&= \begin{cases}
\displaystyle\sum_{j\neq n} \Pr\{P^{(t)} = (\text{type-}k,\text{type-}j)\}\Pr\{\text{type-}k \text{ solution at address 1 is} \\
\qquad\qquad \text{selected from } P^{(t)}(=\mathcal{P}^{(t)}) \text{ in a single} \\
\qquad\qquad \text{selection process} \mid P^{(t)} = (\text{type-}k,\text{type-}j)\} \\
\quad + \displaystyle\sum_{i\neq n} \Pr\{P^{(t)} = (\text{type-}i,\text{type-}k)\}\Pr\{\text{type-}k \text{ solution at address 2 is} \\
\qquad\qquad \text{selected from } P^{(t)}(=\mathcal{P}^{(t)}) \text{ in a single} \\
\qquad\qquad \text{selection process} \mid P^{(t)} = (\text{type-}i,\text{type-}k)\} \\
\qquad\qquad\qquad \text{if } k \neq n \\
\displaystyle\sum_{i=n \text{ or } j=n}\sum \Pr\{P^{(t)} = (\text{type-}i,\text{type-}j)\} \qquad \text{if } k = n
\end{cases} \\
&= \begin{cases}
\displaystyle\sum_{j\neq n} \frac{f_k}{f_k + f_j}\left(x_k^{(t)} q_k^{(t)}(j) + x_j^{(t)} q_j^{(t)}(k)\right) & \text{if } k \neq n \\
\displaystyle x_n^{(t)} + \sum_{j\neq n} x_j^{(t)} q_j^{(t)}(n) & \text{if } k = n
\end{cases}
\end{aligned}
$$

Therefore, we have

$$
\begin{bmatrix} y_1^{(t)} \\ \vdots \\ y_n^{(t)} \end{bmatrix} = S^{(t)} \begin{bmatrix} x_1^{(t)} \\ \vdots \\ x_n^{(t)} \end{bmatrix}, \tag{14}
$$

where $S^{(t)} = [s_{i,j}^{(t)}]$ is an $n \times n$ matrix having (i, j)th entry

$$
s_{i,j}^{(t)} = \begin{cases}
\dfrac{f_i}{f_i + f_i} q_i^{(t)}(i) + \displaystyle\sum_{a\neq n} \frac{f_i}{f_i + f_a} q_i^{(t)}(a) & \text{if } i = j < n \\
\dfrac{f_i}{f_j + f_i} q_j^{(t)}(i) & \text{if } i < n,\ j < n,\ i \neq j \\
q_j^{(t)}(n) & \text{if } i = n,\ j < n \\
0 & \text{if } i < n,\ j = n \\
1 & \text{if } i = n,\ j = n.
\end{cases}
$$

We also have

$$
x^{(t+1)} = M\left\langle y_1^{(t)}, \ldots, y_n^{(t)} \right\rangle^T. \tag{15}
$$

Update Rule for $q_i^{(t)} = \langle q_i^{(t)}(1), \ldots, q_i^{(t)}(n)\rangle^T$ to Change: When the population size equals 2, from Theorem 3.2(a) and (b) we have a rule

$$q_a^{(t+1)}(b) = \frac{\Pr\left\{P^{(t+1)} = (\text{type-}a, \text{type-}b)\right\}}{\Pr\left\{P^{(t+1)} = (\text{type-}a, \text{type-}b) \text{ for some } b\right\}}, \tag{16}$$

where

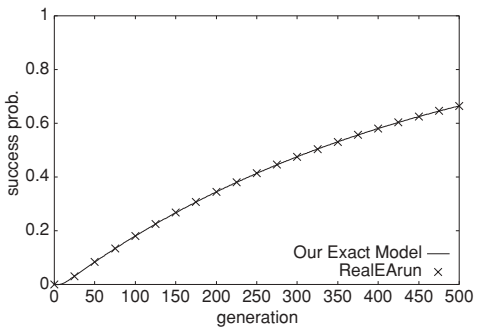
$$\begin{aligned}
&\Pr\{P^{(t+1)} = (\text{type-}a, \text{type-}b)\} \\
&= \sum_{i\neq n}\sum_{j\neq n} x_i^{(t)} q_i^{(t)}(j) \frac{f_i m_{a,i} + f_j m_{a,j}}{f_i + f_j} \frac{f_i m_{b,i} + f_j m_{b,j}}{f_i + f_j} \\
&\quad + \sum_{i=n \text{ or } j=n}\sum x_i^{(t)} q_i^{(t)}(j) m_{a,n} m_{b,n}
\end{aligned}$$

and

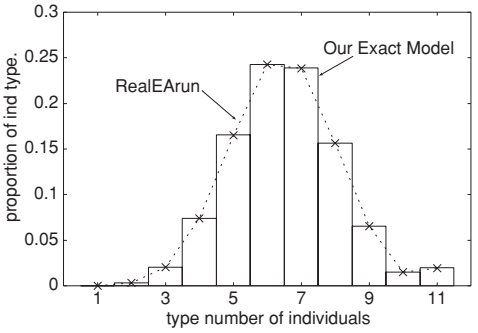
$$\begin{aligned}
&\Pr\left\{P^{(t+1)} = (\text{type-}a, \text{type-}b) \text{ for some } b\right\} \\
&= \sum_{i\neq n}\sum_{j\neq n} x_i^{(t)} q_i^{(t)}(j) \frac{f_i m_{a,i} + f_j m_{a,j}}{f_i + f_j} + \sum_{i=n \text{ or } j=n}\sum x_i^{(t)} q_i^{(t)}(j) m_{a,n}.
\end{aligned}$$

Theorem 3.2 describes how to use distributions $x^{(t)}, q_1^{(t)}, \ldots, q_n^{(t)}$ to compute the distribution over ordered-tuple populations $P^{(t+1)}$ in the next generation, while Equations (13) and (16) describe how to convert a distribution over ordered-tuple populations $P^{(t+1)}$ into $x^{(t+1)}$ and $q_i^{(t+1)}$, respectively. Through the quantity $q_i^{(t)}(j)$, we can directly measure the interrelation between the event of type-i solution occurring in $P^{(t)}$ and the event of type-j solution occurring in $P^{(t)}$.

EXPERIMENT 3.4 (Onemax Problem, String Length = 10, Population Size = 2): In order to compare our predictions with empirical results, this paper chooses the onemax (or ones-counting) problem that has a binary string search space and is considered to be a common test problem for theoretical investigations. Generally, in the onemax problem on strings of length l, the fitness value of candidate solution $x \in \{0,1\}^l$ is defined to be the number of ones in x, and is required to be maximized. The search space involves 2^l possible bit strings. Under crossoverless environments, however, we may think two bit strings are essentially equivalent if one string can be transformed to another through the operation of rearranging bit elements. Then, 2^l strings in the search space are classified into $l+1$ types by their fitness values. Suppose that the class of strings with fitness $i \in \{0, 1, \ldots, l\}$ is labeled as type-$(i+1)$. Using the above notation, it follows that $n =$ (the size of the search space) $= l + 1$, and ideally that $f_i =$ (the fitness of type-i string) $= i - 1$. To avoid expression $0/0$ that may occur in fitness-proportionate selection, however, we redefine $f_1 = 10^{-30}$. We now consider the evolutionary search processes that start with a uniform population of type-1 strings and iterate alternate application of fitness-proportionate selection and bit-flipping mutation. In the bit-flipping mutation, each bit in a parent string is separately flipped into the opposite value with some small probability p_m. Through this mutation with *mutation*



(a) The success probability.



(b) The expected population vector at generation t=20.

Figure 1: Our exact model versus 10^6 real EA runs on the onemax problem, under conditions string_length=10, mutation_rate=0.1, and population_size=2.

rate p_{m}, the probability of type-$(a+1)$ string being mutated to type-$(b+1)$ is determined as follows:

$$\Pr\{\text{type-}(a+1)\text{ string mutates to type-}(b+1)\}$$

$$= \begin{cases} \displaystyle\sum_{i=0}^{l-b} \binom{a}{i}\binom{l-a}{b-a+i} p_{\mathrm{m}}^{b-a+2i}(1-p_{\mathrm{m}})^{l+a-b-2i} & \text{if } a \le b \text{ and } l \le a+b \\ \displaystyle\sum_{i=0}^{a} \binom{a}{i}\binom{l-a}{b-a+i} p_{\mathrm{m}}^{b-a+2i}(1-p_{\mathrm{m}})^{l+a-b-2i} & \text{if } a \le b \text{ and } l \ge a+b \\ \displaystyle\sum_{i=0}^{l-a} \binom{l-a}{i}\binom{a}{a-b+i} p_{\mathrm{m}}^{a-b+2i}(1-p_{\mathrm{m}})^{l-a+b-2i} & \text{if } a \ge b \text{ and } l \le a+b \\ \displaystyle\sum_{i=0}^{b} \binom{l-a}{i}\binom{a}{a-b+i} p_{\mathrm{m}}^{a-b+2i}(1-p_{\mathrm{m}})^{l-a+b-2i} & \text{if } a \ge b \text{ and } l \ge a+b. \end{cases}$$

Note that mutating from the optimum is allowed in the EAs described at the fourth paragraph in Section 3, although it is not allowed in the mutation matrix M that is used to observe the probability distribution over virtual populations $P^{(t)}$. For the onemax problem on strings of length $l = 10$, by letting r(population size) $= 2$, letting $p_{\mathrm{m}} = 1/l = 0.1$, specifying initial distributions

$$x^{(0)} = q_i^{(0)} = \langle 1, 0, \ldots, 0\rangle^T \text{ for every } i \in \{1, 2, \ldots, n\},$$

and repetitively applying the above update rules given by Equations (14), (15), and (16), we can observe how $x^{(t)}$ and $q_i^{(t)}$ change over time as evolution proceeds. Since probability distributions $x^{(t)}$ and $q_i^{(t)}$ jointly indicate how likely it is that each virtual population will occur at generation t over all possible trajectories of the associated EA runs that start with the initial population (type-1, type-1), we can average any quantity over these trajectories. For example, to obtain the success probability, it suffices to calculate $\sum\sum_{i=n \text{ or } j=n} x_i^{(t)} q_i^{(t)}(j) = x_n^{(t)} + \sum_{i\neq n} x_i^{(t)} q_i^{(t)}(n)$. Figure 1(a) shows the graph

of this expression as a function of generation t, along with some points that indicate empirical values of the success probability measured through the corresponding 10^6 real EA runs. It should be noted that in order to obtain the empirical results for confirming the validity of our predictions, we need to implement the evolution of population $P^{(t)}$, not the evolution of $\mathcal{P}^{(t)}$. So, we iterated the following EA process:

$P^{(0)} \leftarrow$ (0000000000, 0000000000);
$t \leftarrow 0$;
repeat {
observe the current population $P^{(t)}$;
if ($P^{(t)}$ contains the optimum **or** $t \geq 500$)
break;
$Q^{(t)} \leftarrow$ an ordered-tuple of 2 strings that are selected from $P^{(t)}$ through fitness-proportionate scheme;
$P^{(t+1)} \leftarrow$ an ordered-tuple population of 2 strings that are obtained from elements in $Q^{(t)}$ by the bit-flipping mutation;
$t \leftarrow t + 1$;
}

The search process proceeds in a similar manner as in the pseudo-code at the fourth paragraph in Section 3. In order to avoid wasting computer resources, however, this EA process on $P^{(t)}$ terminates when either the optimum has been found or 500 generations have been run. We used a pseudo-random number generator, called "Mersenne Twister" (Matsumoto and Nishimura, 1998). For each $t \in \{0, 25, 50, \ldots, 500\}$, the resulting proportion of runs that find the optimum within t generations is calculated, and indicated by a cross ($\times$) in Figure 1(a). As regards this figure, we see that the empirical points lie on the model-based graph as a matter of course. To average the proportion of type-k solution in $P^{(t)}$ over all possible trajectories, it suffices to calculate $(x_k^{(t)} + \sum_i x_i^{(t)} q_i^{(t)}(k))/2$. Figure 1(b) shows the bar graph of this expression for $t = 20$ as a function of type number k, along with the empirical graph of the expected population vector at generation $t = 20$ averaged over the corresponding 10^6 real EA runs. We also see that two graphs coincide.

REMARK 3.5 (Extension to the Case When Multiple Optimums Exist): We can easily extend our discussion so that the search space can have multiple optimums. For instance, if the type-n_0 through type-n solutions are optimal and no other optimum exists, then the equation in Theorem 3.2 (a) can be extended as follows:

$$\Pr\{\text{the individual at address 1 in } P^{(t+1)} \text{ is type-}a\}$$
$$= \sum_{i<n_0} \sum_{j<n_0} x_i^{(t)} q_i^{(t)}(j) \frac{f_i m_{a,i} + f_j m_{a,j}}{f_i + f_j} + \sum_{i \geq n_0 \text{ or } j \geq n_0} \sum x_i^{(t)} q_i^{(t)}(j) m_{a,n}.$$

3.2 When the Population Size = r

We now follow and extend the notation in the previous section, and generalize the results. Let r be the population size. Concerning the virtual population $P^{(t)}$, let $x_i^{(t)}$ be the probability of the individual at address 1 exhibiting the type-i solution, and for each $k \in \{2, 3, \ldots, r\}$ and type numbers $i_1, \ldots, i_k$, let $q_{i_1,\ldots,i_{k-1}}^{(t)}(i_k)$ be the conditional probability

of the individual at address k being type-i_k given the individuals at address 1 through $(k-1)$ being type-i_1 through type-$i_{k\text{-}1}$, respectively. Then, it follows that the probability distribution over ordered-tuple populations $P^{(t)}$ is determined by

$$\Pr\{P^{(t)} = (\text{type-}i_1, \text{type-}i_2, \ldots, \text{type-}i_r)\} = x^{(t)}_{i_1} q^{(t)}_{i_1}(i_2) q^{(t)}_{i_1,i_2}(i_3) \cdots q^{(t)}_{i_1,\ldots,i_{r\text{-}1}}(i_r).$$

We will give update rules that indicate how probability distributions $x^{(t)} = \langle x^{(t)}_1, \ldots, x^{(t)}_n \rangle^T$, $q^{(t)}_{i_1} = \langle q^{(t)}_{i_1}(1), \ldots, q^{(t)}_{i_1}(n) \rangle^T$, $\ldots$, $q^{(t)}_{i_1,\ldots,i_{r\text{-}1}} = \langle q^{(t)}_{i_1,\ldots,i_{r\text{-}1}}(1), \ldots, q^{(t)}_{i_1,\ldots,i_{r\text{-}1}}(n) \rangle^T$, where $i_1, \ldots, i_{r\text{-}1} \in \{1, 2, \ldots, n\}$, change over time. As a preparation, we generalize Theorem 3.2 and Corollary 3.3.

THEOREM 3.6 (Crossoverless EA): *Consider a finite population crossoverless EA, and suppose that the population size equals r. Then, for every $k \in \{1, 2, \ldots, r\}$ and every type number $a_1, \ldots, a_k$,*

$$\begin{aligned}
&\Pr\{\textit{the individuals at address } 1, \ldots, k \textit{ in } P^{(t+1)} \textit{ are type-}a_1, \ldots, \textit{type-}a_k, \textit{ respectively}\} \\
&= \sum_{i_1 \neq n} \sum_{i_2 \neq n} \cdots \sum_{i_r \neq n} x^{(t)}_{i_1} q^{(t)}_{i_1}(i_2) \cdots q^{(t)}_{i_1,\ldots,i_{r\text{-}1}}(i_r) \\
&\qquad \prod_{h=1}^{k} \frac{f_{i_1} m_{a_h,i_1} + f_{i_2} m_{a_h,i_2} + \cdots + f_{i_r} m_{a_h,i_r}}{f_{i_1} + f_{i_2} + \cdots + f_{i_r}} \\
&\quad + \sum_{i_1,i_2,\ldots,i_r : n \in \{i_1,i_2,\ldots,i_r\}} \sum \cdots \sum x^{(t)}_{i_1} q^{(t)}_{i_1}(i_2) \cdots q^{(t)}_{i_1,\ldots,i_{r\text{-}1}}(i_r) \prod_{h=1}^{k} m_{a_h,n}.
\end{aligned}$$

PROOF: This result can be proven by an argument similar to that of Theorem 3.2. □

COROLLARY 3.7 (Crossoverless EA): *Consider a finite population crossoverless EA, and suppose that the population size equals r. For every noninitial generation t, every $k \in \{1, 2, \ldots, r\}$, every type number $a_1, a_2, \ldots, a_k$ and $b_1, b_2, \ldots, b_k$, if $\{a_1, a_2, \ldots, a_k\}$ is equal to $\{b_1, b_2, \ldots, b_k\}$ as multisets, that is, if the sequence $a_1 a_2 \cdots a_k$ can be obtained from $b_1 b_2 \cdots b_k$ by rearrangement, then*

$$x^{(t)}_{a_1} q^{(t)}_{a_1}(a_2) \cdots q^{(t)}_{a_1,\ldots,a_{k\text{-}1}}(a_k) = x^{(t)}_{b_1} q^{(t)}_{b_1}(b_2) \cdots q^{(t)}_{b_1,\ldots,b_{k\text{-}1}}(b_k).$$

PROOF: From Theorem 3.6, we see that $\Pr\{$the individuals at address $1, \ldots, k$ are type-$a_1, \ldots,$ type-a_k respectively at noninitial generation $t\}$ does not depend on the order in which $a_1, a_2, \ldots, a_{k\text{-}1}$ and a_k occur. Therefore, if $\{a_1, a_2, \ldots, a_k\} = \{b_1, b_2, \ldots, b_k\}$ as multisets, for every noninitial generation t,

$$\begin{aligned}
&x^{(t)}_{a_1} q^{(t)}_{a_1}(a_2) \cdots q^{(t)}_{a_1,\ldots,a_{k\text{-}1}}(a_k) \\
&\quad = \Pr\{\text{the individuals at address } 1, \ldots, k \text{ are type-}a_1, \ldots, \text{type-}a_k \text{ respectively at } t\} \\
&\quad = \Pr\{\text{the individuals at address } 1, \ldots, k \text{ are type-}b_1, \ldots, \text{type-}b_k \text{ respectively at } t\} \\
&\quad = x^{(t)}_{b_1} q^{(t)}_{b_1}(b_2) \cdots q^{(t)}_{b_1,\ldots,b_{k\text{-}1}}(b_k).
\end{aligned}$$

□

Based on Theorem 3.6, we now give the update rules that govern how probability distributions $x^{(t)}, q_{i_1}^{(t)}, q_{i_1,i_2}^{(t)}, \ldots, q_{i_1,\ldots,i_{r-1}}^{(t)}$ change over time.

Update Rule for $x^{(t)}=\langle x_1^{(t)}, \ldots, x_n^{(t)}\rangle^T$ to Change: From Theorem 3.6, we have a rule

$$\begin{aligned} x_a^{(t+1)} &= \Pr\{\text{the individual at address 1 in } P^{(t+1)} \text{ is type-}a\} \\ &= \sum_{i_1\neq n}\sum_{i_2\neq n}\cdots\sum_{i_r\neq n} x_{i_1}^{(t)} q_{i_1}^{(t)}(i_2)\cdots q_{i_1,\ldots,i_{r-1}}^{(t)}(i_r) \\ &\qquad \frac{f_{i_1}m_{a,i_1}+f_{i_2}m_{a,i_2}+\cdots+f_{i_r}m_{a,i_r}}{f_{i_1}+f_{i_2}+\cdots+f_{i_r}} \\ &\quad + \sum_{i_1,i_2,\ldots,i_r : n\in\{i_1,i_2,\ldots,i_r\}}\cdots\sum x_{i_1}^{(t)} q_{i_1}^{(t)}(i_2)\cdots q_{i_1,\ldots,i_{r-1}}^{(t)}(i_r) m_{a,n}. \end{aligned} \tag{17}$$

To show the specific effects of fitness-proportionate selection and mutation separately, let $y_k^{(t)}$ be the probability of the type-k solution occurring at address 1 in $Q^{(t)}$. Then, we can calculate $y_k^{(t)}$ as follows:

$$y_k^{(t)} = \begin{cases} \displaystyle\sum_{j_1\neq n}\sum_{j_2\neq n}\cdots\sum_{j_{r-1}\neq n} \frac{f_k}{f_k+f_{j_1}+f_{j_2}+\cdots+f_{j_{r-1}}} \Big(\\ \quad \Big(x_k^{(t)} q_k^{(t)}(j_1) q_{k,j_1}^{(t)}(j_2)\cdots q_{k,j_1,\ldots,j_{r-2}}^{(t)}(j_{r-1}) \\ \quad + x_{j_1}^{(t)} q_{j_1}^{(t)}(k) q_{j_1,k}^{(t)}(j_2)\cdots q_{j_1,k,j_2,\ldots,j_{r-2}}^{(t)}(j_{r-1}) \\ \quad + x_{j_1}^{(t)} q_{j_1}^{(t)}(j_2) q_{j_1,j_2}^{(t)}(k)\cdots q_{j_1,j_2,k,\ldots,j_{r-2}}^{(t)}(j_{r-1}) \\ \quad + \cdots \\ \quad + x_{j_1}^{(t)} q_{j_1}^{(t)}(j_2)\cdots q_{j_1,j_2,\ldots,j_{r-2}}^{(t)}(k) q_{j_1,j_2,\ldots,j_{r-2},k}^{(t)}(j_{r-1}) \\ \quad + x_{j_1}^{(t)} q_{j_1}^{(t)}(j_2)\cdots q_{j_1,j_2,\ldots,j_{r-2}}^{(t)}(j_{r-1}) q_{j_1,j_2,\ldots,j_{r-1}}^{(t)}(k) \Big) & \text{if } k\neq n \\ x_n^{(t)} + \displaystyle\sum_{j_1\neq n} x_{j_1}^{(t)} \Big(q_{j_1}^{(t)}(n) \\ \quad + \displaystyle\sum_{j_2\neq n} q_{j_1}^{(t)}(j_2) \Big(\cdots \Big(q_{j_1,\ldots,j_{r-3}}^{(t)}(n) \\ \quad\quad + \displaystyle\sum_{j_{r-2}\neq n} q_{j_1,\ldots,j_{r-3}}^{(t)}(j_{r-2}) \Big(q_{j_1,\ldots,j_{r-2}}^{(t)}(n) \\ \quad\quad\quad + \displaystyle\sum_{j_{r-1}\neq n} q_{j_1,\ldots,j_{r-2}}^{(t)}(j_{r-1}) q_{j_1,\ldots,j_{r-1}}^{(t)}(n) \Big) \Big) \cdots \Big) \Big) & \text{if } k=n \end{cases} \tag{18}$$

Therefore, we may calculate $y^{(t)}=\langle y_1^{(t)}, \ldots, y_n^{(t)}\rangle^T$ and $x^{(t+1)} = \langle x_1^{(t+1)}, \ldots, x_n^{(t+1)}\rangle^T$ as follows:

$$y^{(t)} = S^{(t)} x^{(t)}, \tag{19}$$

$$x^{(t+1)} = M y^{(t)}, \tag{20}$$

where $S^{(t)}=[s^{(t)}_{i,j}]$ is an $n \times n$ matrix having $(i,\ j)$th entry

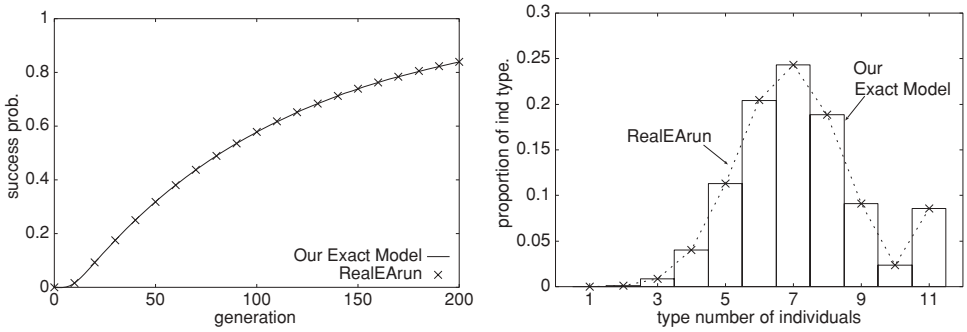
$$s^{(t)}_{i,j} = \begin{cases} \sum_{j_2 \neq n} \cdots \sum_{j_{r\text{-}1} \neq n} \frac{f_i}{f_i + f_i + f_{j_2} + \cdots + f_{j_{r\text{-}1}}} \Big(\\ \quad \times \Big(q^{(t)}_i(i) q^{(t)}_{i,i}(j_2) \cdots q^{(t)}_{i,i,j_2,\ldots,j_{r\text{-}2}}(j_{r\text{-}1}) \\ \quad + q^{(t)}_i(j_2) q^{(t)}_{i,j_2}(i) \cdots q^{(t)}_{i,j_2,i,j_3,\ldots,j_{r\text{-}2}}(j_{r\text{-}1}) + \cdots \\ \quad + q^{(t)}_i(j_2) \cdots q^{(t)}_{i,j_2,\ldots,j_{r\text{-}2}}(i) q^{(t)}_{i,j_2,\ldots,j_{r\text{-}2},i}(j_{r\text{-}1}) \\ \quad + q^{(t)}_i(j_2) \cdots q^{(t)}_{i,j_2,\ldots,j_{r\text{-}2}}(j_{r\text{-}1}) q^{(t)}_{i,j_2,\ldots,j_{r\text{-}1}}(i) \Big) \\ \quad + \sum_{j_1 \neq n} \sum_{j_2 \neq n} \cdots \sum_{j_{r\text{-}1} \neq n} \frac{f_i}{f_i + f_{j_1} + f_{j_2} + \cdots + f_{j_{r\text{-}1}}} \\ \qquad q^{(t)}_i(j_1) q^{(t)}_{i,j_1}(j_2) \cdots q^{(t)}_{i,j_1,\ldots,j_{r\text{-}2}}(j_{r\text{-}1}) & \text{if } i = j < n \\ \sum_{j_2 \neq n} \cdots \sum_{j_{r\text{-}1} \neq n} \frac{f_i}{f_i + f_j + f_{j_2} + \cdots + f_{j_{r\text{-}1}}} \Big(\\ \quad \times \Big(q^{(t)}_j(i) q^{(t)}_{j,i}(j_2) \cdots q^{(t)}_{j,i,j_2,\ldots,j_{r\text{-}2}}(j_{r\text{-}1}) \\ \quad + q^{(t)}_j(j_2) q^{(t)}_{j,j_2}(i) \cdots q^{(t)}_{j,j_2,i,\ldots,j_{r\text{-}2}}(j_{r\text{-}1}) + \cdots \\ \quad + q^{(t)}_j(j_2) \cdots q^{(t)}_{j,j_2,\ldots,j_{r\text{-}2}}(i) q^{(t)}_{j,j_2,\ldots,j_{r\text{-}2},i}(j_{r\text{-}1}) \\ \quad + q^{(t)}_j(j_2) \cdots q^{(t)}_{j,j_2,\ldots,j_{r\text{-}2}}(j_{r\text{-}1}) q^{(t)}_{j,j_2,\ldots,j_{r\text{-}1}}(i) \Big) & \text{if } i \neq j, i < n, j < n \\ q^{(t)}_j(n) + \sum_{j_2 \neq n} q^{(t)}_j(j_2) \Big(\cdots \Big(q^{(t)}_{j,j_2,\ldots,j_{r\text{-}3}}(n) \\ \quad + \sum_{j_{r\text{-}2} \neq n} q^{(t)}_{j,j_2,\ldots,j_{r\text{-}3}}(j_{r\text{-}2}) \Big(q^{(t)}_{j,j_2,\ldots,j_{r\text{-}2}}(n) \\ \quad + \sum_{j_{r\text{-}1} \neq n} q^{(t)}_{j,j_2,\ldots,j_{r\text{-}2}}(j_{r\text{-}1}) q^{(t)}_{j,j_2,\ldots,j_{r\text{-}1}}(n) \Big) \Big) \cdots \Big) & \text{if } i = n, j < n \\ 0 & \text{if } i < n, j = n \\ 1 & \text{if } i = j = n. \end{cases}$$

Update rule for $q^{(t)}_{a_1,\ldots,a_{k\text{-}1}} = \langle q^{(t)}_{a_1,\ldots,a_{k\text{-}1}}(1), \ldots, q^{(t)}_{a_1,\ldots,a_{k\text{-}1}}(n) \rangle^T$ to Change: From Theorem 3.6, we have a rule

$$q^{(t+1)}_{a_1,\ldots,a_{k\text{-}1}}(a_k) = \frac{\Pr\{P^{(t+1)}=(\text{type-}a_1, \ldots, \text{type-}a_r) \text{ for some } a_{k+1}, \ldots, a_r\}}{\Pr\{P^{(t+1)}=(\text{type-}a_1, \ldots, \text{type-}a_r) \text{ for some } a_k, a_{k+1}, \ldots, a_r\}}, \tag{21}$$

where

$$\Pr\left\{P^{(t+1)} = (\text{type-}a_1, \ldots, \text{ type-}a_r) \text{ for some } a_{k+1}, \ldots, a_r\right\} = \sum_{i_1 \neq n} \sum_{i_2 \neq n} \cdots \sum_{i_r \neq n} x^{(t)}_{i_1} q^{(t)}_{i_1}(i_2) \cdots q^{(t)}_{i_1,\ldots,i_{r\text{-}1}}(i_r)$$



(a) The success probability.

(b) The expected population vector at generation t=20.

Figure 2: Our exact model versus 10^6 real EA runs on the onemax problem, under conditions string_length=10, mutation_rate=0.1, and population_size=5.

$$\times \prod_{l=1}^{k} \frac{f_{i_1} m_{a_l,i_1} + f_{i_2} m_{a_l,i_2} + \cdots + f_{i_r} m_{a_l,i_r}}{f_{i_1} + f_{i_2} + \cdots + f_{i_r}}$$

$$+ \sum_{i_1,i_2,\ldots,i_r : n \in \{i_1,i_2,\ldots,i_r\}} \sum \cdots \sum x_{i_1}^{(t)} q_{i_1}^{(t)}(i_2) \cdots q_{i_1,\ldots,i_{r-1}}^{(t)}(i_r) \prod_{l=1}^{k} m_{a_l,n}.$$

EXPERIMENT 3.8 (Onemax Problem, String Length = 10, Population Size = 5): We now consider the onemax problem described in Experiment 3.4, and make an experiment similar to that of Experiment 3.4 for larger population size. In this experiment, by letting l(string length) = 10, letting p_{m}(mutation rate) $= 1/l = 0.1$, letting r(population size) $= 5$, making all the initial distributions $x^{(0)}$, $q_{i_1}^{(0)}$, $q_{i_1,i_2}^{(0)}$, $q_{i_1,i_2,i_3}^{(0)}$, $q_{i_1,i_2,i_3,i_4}^{(0)}$ be $\langle 1, 0, \ldots, 0 \rangle^T$, and repetitively applying the above update rules given by Equations (19), (20), and (21), we can observe how $x^{(t)}$, $q_{i_1}^{(t)}$, $q_{i_1,i_2}^{(t)}$, $q_{i_1,i_2,i_3}^{(t)}$, and $q_{i_1,i_2,i_3,i_4}^{(t)}$ change over time as evolution proceeds. To observe the success probability, we calculate $\sum\sum\sum\sum\sum_{i_1=n \text{ or } i_2=n \text{ or } i_3=n \text{ or } i_4=n \text{ or } i_5=n} x_{i_1}^{(t)} q_{i_1}^{(t)}(i_2) q_{i_1,i_2}^{(t)}(i_3) q_{i_1,i_2,i_3}^{(t)}(i_4) q_{i_1,i_2,i_3,i_4}^{(t)}(i_5)$ for each generation t. Figure 2(a) shows the graph of this expression as a function of generation t, along with some points that indicate empirical values of the success probability measured through the corresponding 10^6 real EA runs. We see that these empirical points lie on the model-based graph as a matter of course. To average the proportion of type-k solution in $P^{(t)}$ over all possible trajectories, we calculate $(x_k^{(t)} + \sum_{i_1} x_{i_1}^{(t)} q_{i_1}^{(t)}(k) + \sum_{i_1}\sum_{i_2} x_{i_1}^{(t)} q_{i_1}^{(t)}(i_2) q_{i_1,i_2}^{(t)}(k) + \sum_{i_1}\sum_{i_2}\sum_{i_3} x_{i_1}^{(t)} q_{i_1}^{(t)}(i_2) q_{i_1,i_2}^{(t)}(i_3) q_{i_1,i_2,i_3}^{(t)}(k) + \sum_{i_1}\sum_{i_2}\sum_{i_3}\sum_{i_4} x_{i_1}^{(t)} q_{i_1}^{(t)}(i_2) q_{i_1,i_2}^{(t)}(i_3) q_{i_1,i_2,i_3}^{(t)}(i_4) q_{i_1,i_2,i_3,i_4}^{(t)}(k))/5$. Figure 2(b) shows the bar graph of this expression for $t = 20$ as a function of type number k, along with the empirical graph of the expected population vector at generation $t = 20$ averaged over the corresponding 10^6 real EA runs. We also see that two graphs coincide.

REMARK 3.9 (Modification for EAs with Rank-Based Selection): We can easily modify the above discussion so as to cope with rank-based selection. Let $\mathrm{rank}(j; i_1, \ldots, i_r)$

denote the rank of type-i_j string in population (type-i_1, ...,type-i_r), where we say the individual has *rank* k if it is the kth worst one in the population. Let $\pi(k, r)$ denote the probability of the rank k individual being selected in the single selection process. Then, the equation in Theorem 3.6 can be modified as follows:

$$\begin{aligned}
&\Pr\{P^{(t+1)} = (\text{type-}a_1, \ldots, \text{type-}a_k)\} \\
&= \sum_{i_1 \neq n} \sum_{i_2 \neq n} \cdots \sum_{i_r \neq n} x^{(t)}_{i_1} q^{(t)}_{i_1}(i_2) \cdots q^{(t)}_{i_1,\ldots,i_{r-1}}(i_r) \prod_{h=1}^{k} \sum_{j=1}^{r} \pi(\text{rank}(j; i_1, \ldots, i_r), r) m_{a_h, i_j} \\
&\quad + \sum_{i_1, i_2, \ldots, i_r : n \in \{i_1, i_2, \ldots, i_r\}} \sum \cdots \sum x^{(t)}_{i_1} q^{(t)}_{i_1}(i_2) \cdots q^{(t)}_{i_1,\ldots,i_{r-1}}(i_r) \prod_{h=1}^{k} m_{a_h, n}.
\end{aligned}$$

For tournament selection with tournament size α, when every individual has a unique rank number between 1 (worst one) and r (best one), we can determine $\pi(k, r)$ by $\pi(k, r) = (k^\alpha - (k-1)^\alpha)/r^\alpha$. To proceed with the above calculation, however, we need to determine each individual's rank in each possible population (type-i_1, ...,type-i_r) every generation, and so require $O(n^r r \log r)$ additional computational steps per generation.

3.3 Approximation

While the previous section presents an exact model, this section presents approximate models for describing how the probability distribution $x^{(t)}$ changes over time.

Zeroth-Order Approximation. Consider the model described in Section 3.2. If we assume $q^{(t)}_{i_1}(i_2) \approx x^{(t)}_{i_2}$, $q^{(t)}_{i_1,i_2}(i_3) \approx x^{(t)}_{i_3}$, ..., $q^{(t)}_{i_1,\ldots,i_{r-1}}(i_r) \approx x^{(t)}_{i_r}$ for every $i_1, \ldots, i_r$ and t, then the right-hand side of Equation (18) is approximated as

$$y^{(t)}_k \approx \begin{cases} r x^{(t)}_k \displaystyle\sum_{j_1 \neq n} \sum_{j_2 \neq n} \cdots \sum_{j_{r-1} \neq n} \frac{f_k}{f_k + f_{j_1} + f_{j_2} + \cdots + f_{j_{r-1}}} x^{(t)}_{j_1} x^{(t)}_{j_2} \cdots x^{(t)}_{j_{r-1}} & \\ \qquad\qquad \text{if } k \neq n & \\ x^{(t)}_n \big(1 + \big(1 - x^{(t)}_n\big)\big(1 + \big(1 - x^{(t)}_n\big)(\cdots \big(1 + \big(1 - x^{(t)}_n\big)\big(2 - x^{(t)}_n\big)\big) \cdots)\big)\big) & \\ \qquad\qquad \text{if } k = n, & \end{cases} \tag{22}$$

and so the (i, j)th entry of the selection matrix $S^{(t)}$ in Equation (19) is approximated by

$$s^{(t)}_{i,j} \approx \begin{cases} r \displaystyle\sum_{j_1 \neq n} \sum_{j_2 \neq n} \cdots \sum_{j_{r-1} \neq n} \frac{f_i}{f_i + f_{j_1} + f_{j_2} + \cdots + f_{j_{r-1}}} x^{(t)}_{j_1} x^{(t)}_{j_2} \cdots x^{(t)}_{j_{r-1}} & \\ \qquad\qquad \text{if } i = j < n & \\ 1 + \big(1 - x^{(t)}_n\big)\big(1 + \big(1 - x^{(t)}_n\big)(\cdots \big(1 + \big(1 - x^{(t)}_n\big)\big(2 - x^{(t)}_n\big)\big) \cdots)\big) & \\ \qquad\qquad \text{if } i = j = n & \\ 0 \qquad\qquad \text{otherwise.} & \end{cases} \tag{23}$$

Note that those assumptions dispense with $q^{(t)}_{i_1}$ through $q^{(t)}_{i_1,\ldots,i_{r-1}}$ for maintaining $x^{(t)}$.

First-Order Approximation. Consider the model described in Section 3.2. If we assume $q^{(t)}_{i_1,i_2}(i_3) \approx q^{(t)}_{i_1}(i_3)$, $q^{(t)}_{i_1,i_2,i_3}(i_4) \approx q^{(t)}_{i_1}(i_4)$, ..., $q^{(t)}_{i_1,\ldots,i_{r-1}}(i_r) \approx q^{(t)}_{i_1}(i_r)$ for every $i_1, \ldots, i_r$

and t, then the right-hand side of Equation (18) is approximated as

$$y_k^{(t)} \approx \begin{cases} \displaystyle\sum_{j_1\neq n}\sum_{j_2\neq n}\cdots\sum_{j_{r\text{-}1}\neq n} \frac{f_k}{f_k+f_{j_1}+f_{j_2}+\cdots+f_{j_{r\text{-}1}}} \\ \qquad\qquad \times \left(x_k^{(t)} q_k^{(t)}(j_1) q_k^{(t)}(j_2)\cdots q_k^{(t)}(j_{r\text{-}1})\right. \\ \qquad \left. +(r-1)x_{j_1}^{(t)} q_{j_1}^{(t)}(k) q_{j_1}^{(t)}(j_2)\cdots q_{j_1}^{(t)}(j_{r\text{-}1})\right) & \text{if } k\neq n \\ x_n^{(t)} + \displaystyle\sum_{j_1\neq n} x_{j_1}^{(t)} q_{j_1}^{(t)}(n)\Bigg(1+\sum_{j_2\neq n} q_{j_1}^{(t)}(j_2) \\ \qquad \Bigg(\cdots\Bigg(1+\sum_{j_{r\text{-}2}\neq n} q_{j_1}^{(t)}(j_{r\text{-}2})\Bigg(1+\sum_{j_{r\text{-}1}\neq n} q_{j_1}^{(t)}(j_{r\text{-}1})\Bigg)\Bigg)\cdots\Bigg)\Bigg) & \text{if } k=n, \end{cases} \tag{24}$$

and so the (i,j)th entry of the selection matrix $S^{(t)}$ in Equation (19) is approximated by

$$s_{i,j}^{(t)} \approx \begin{cases} (r-1)\displaystyle\sum_{j_2\neq n}\cdots\sum_{j_{r\text{-}1}\neq n} \frac{f_i}{f_i+f_j+f_{j_2}+\cdots+f_{j_{r\text{-}1}}} q_j^{(t)}(i) q_j^{(t)}(j_2)\cdots q_j^{(t)}(j_{r\text{-}1}) \\ \quad +\displaystyle\sum_{j_1\neq n}\sum_{j_2\neq n}\cdots\sum_{j_{r\text{-}1}\neq n} \frac{f_i}{f_i+f_{j_1}+f_{j_2}+\cdots+f_{j_{r\text{-}1}}} q_i^{(t)}(j_1) q_i^{(t)}(j_2)\cdots q_i^{(t)}(j_{r\text{-}1}) \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{if } i=j<n \\ (r-1)\displaystyle\sum_{j_2\neq n}\cdots\sum_{j_{r\text{-}1}\neq n} \frac{f_i}{f_i+f_j+f_{j_2}+\cdots+f_{j_{r\text{-}1}}} q_j^{(t)}(i) q_j^{(t)}(j_2)\cdots q_j^{(t)}(j_{r\text{-}1}) \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{if } i\neq j, i<n, j<n \\ q_j^{(t)}(n)\Bigg(1+\displaystyle\sum_{j_2\neq n} q_j^{(t)}(j_2)\Bigg(\cdots\Bigg(1+\sum_{j_{r\text{-}2}\neq n} q_j^{(t)}(j_{r\text{-}2}) \\ \qquad\qquad \Bigg(1+\displaystyle\sum_{j_{r\text{-}1}\neq n} q_j^{(t)}(j_{r\text{-}1})\Bigg)\Bigg)\cdots\Bigg)\Bigg) & \text{if } i=n, j<n \\ 0 & \text{if } i<n, j=n \\ 1 & \text{if } i=j=n. \end{cases}$$

Under those assumptions, the probability $q_{a_1}^{(t+1)}(a_2)$ in Equation (21) is approximately calculated by substituting

$$\begin{aligned} &\Pr\left\{P^{(t+1)} = (\text{type-}a_1,\ldots,\ \text{type-}a_r) \text{ for some } a_{k+1},\ldots,a_r\right\} \\ &\approx \sum_{i_1\neq n}\cdots\sum_{i_r\neq n} x_{i_1}^{(t)} q_{i_1}^{(t)}(i_2)\cdots q_{i_1}^{(t)}(i_r)\prod_{l=1}^{k}\left(\frac{f_{i_1}m_{a_l,i_1}+f_{i_2}m_{a_l,i_2}+\cdots+f_{i_r}m_{a_l,i_r}}{f_{i_1}+f_{i_2}+\cdots+f_{i_r}}\right) \\ &\quad +\sum_{i_1,\ldots,i_r:n\in\{i_1,i_2,\ldots,i_r\}} x_{i_1}^{(t)} q_{i_1}^{(t)}(i_2)\cdots q_{i_1}^{(t)}(i_r)\prod_{l=1}^{k} m_{a_l,n}. \end{aligned}$$

Note that the above assumptions dispense with $q_{i_1,i_2}^{(t)}$ through $q_{i_1,\ldots,i_{r-1}}^{(t)}$ for maintaining $x^{(t)}$.

Second-Order Approximation. Consider the model described in Section 3.2. If we assume $q_{i_1,i_2,i_3}^{(t)}(i_4) \approx q_{i_1,i_2}^{(t)}(i_4), \ldots, q_{i_1,\ldots,i_{r-1}}^{(t)}(i_r) \approx q_{i_1,i_2}^{(t)}(i_r)$ for every $i_1,\ldots,i_r$ and t, then the

right-hand side of Equation (18) is approximated as

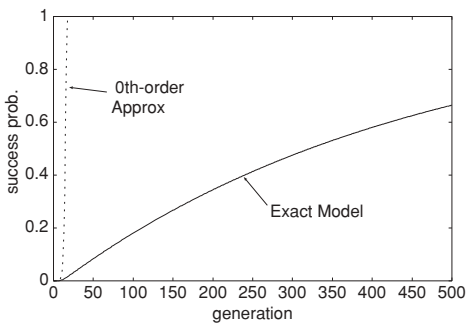
$$y_k^{(t)} \approx \begin{cases} \sum_{j_1 \neq n} \sum_{j_2 \neq n} \cdots \sum_{j_{r\text{-}1} \neq n} \frac{f_k}{f_k + f_{j_1} + f_{j_2} + \cdots + f_{j_{r\text{-}1}}} \Big(\\ \quad \times \Big(x_k^{(t)} q_k^{(t)}(j_1) q_{k,j_1}^{(t)}(j_2) \cdots q_{k,j_1}^{(t)}(j_{r\text{-}1}) \\ \quad + x_{j_1}^{(t)} q_{j_1}^{(t)}(k) q_{j_1,k}^{(t)}(j_2) \cdots q_{j_1,k}^{(t)}(j_{r\text{-}1}) \\ \quad + (r-2) x_{j_1}^{(t)} q_{j_1}^{(t)}(j_2) q_{j_1,j_2}^{(t)}(k) q_{j_1,j_2}^{(t)}(j_3) \cdots q_{j_1,j_2}^{(t)}(j_{r\text{-}1}) \Big) & \text{if } k \neq n \\ x_n^{(t)} + \sum_{j_1 \neq n} x_{j_1}^{(t)} \Bigg(q_{j_1}^{(t)}(n) + \sum_{j_2 \neq n} q_{j_1}^{(t)}(j_2) q_{j_1,j_2}^{(t)}(n) \Bigg(1 + \sum_{j_3 \neq n} q_{j_1,j_2}^{(t)}(j_3) \\ \quad \Bigg(\cdots \Bigg(1 + \sum_{j_{r\text{-}2} \neq n} q_{j_1,j_2}^{(t)}(j_{r\text{-}2}) \Bigg(1 + \sum_{j_{r\text{-}1} \neq n} q_{j_1,j_2}^{(t)}(j_{r\text{-}1}) \Bigg) \Bigg) \cdots \Bigg) \Bigg) \Bigg) & \text{if } k = n, \end{cases} \tag{25}$$

and so the (i, j)th entry of the selection matrix $S^{(t)}$ in Equation (19) is approximated by

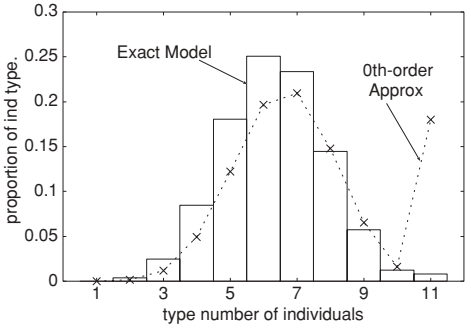
$$s_{i,j}^{(t)} \approx \begin{cases} \sum_{j_2 \neq n} \cdots \sum_{j_{r\text{-}1} \neq n} \frac{f_i}{f_i + f_j + f_{j_2} + \cdots + f_{j_{r\text{-}1}}} \big(q_j^{(t)}(i) q_{j,i}^{(t)}(j_2) \cdots q_{j,i}^{(t)}(j_{r\text{-}1}) \\ \qquad + (r-2) q_j^{(t)}(j_2) q_{j,j_2}^{(t)}(i) q_{j,j_2}^{(t)}(j_3) \cdots q_{j,j_2}^{(t)}(j_{r\text{-}1}) \big) \\ \quad + \sum_{j_1 \neq n} \sum_{j_2 \neq n} \cdots \sum_{j_{r\text{-}1} \neq n} \frac{f_i}{f_i + f_{j_1} + f_{j_2} + \cdots + f_{j_{r\text{-}1}}} q_i^{(t)}(j_1) q_{i,j_1}^{(t)}(j_2) \cdots q_{i,j_1}^{(t)}(j_{r\text{-}1}) \\ & \text{if } i = j < n \\ \sum_{j_2 \neq n} \cdots \sum_{j_{r\text{-}1} \neq n} \frac{f_i}{f_i + f_j + f_{j_2} + \cdots + f_{j_{r\text{-}1}}} \big(q_j^{(t)}(i) q_{j,i}^{(t)}(j_2) \cdots q_{j,i}^{(t)}(j_{r\text{-}1}) \\ \qquad + (r-2) q_j^{(t)}(j_2) q_{j,j_2}^{(t)}(i) q_{j,j_2}^{(t)}(j_3) \cdots q_{j,j_2}^{(t)}(j_{r\text{-}1}) \big) & \text{if } i \neq j,\ i < n,\ j < n \\ q_j^{(t)}(n) + \sum_{j_2 \neq n} q_j^{(t)}(j_2) q_{j,j_2}^{(t)}(n) \Bigg(1 + \sum_{j_3 \neq n} q_{j,j_2}^{(t)}(j_3) \\ \quad \Bigg(\cdots \Bigg(1 + \sum_{j_{r\text{-}2} \neq n} q_{j,j_2}^{(t)}(j_{r\text{-}2}) \Bigg(1 + \sum_{j_{r\text{-}1} \neq n} q_{j,j_2}^{(t)}(j_{r\text{-}1}) \Bigg) \Bigg) \cdots \Bigg) \Bigg) & \text{if } i = n,\ j < n \\ 0 & \text{if } i < n,\ j = n \\ 1 & \text{if } i = j = n. \end{cases}$$

Under those assumptions, the probabilities $q_{a_1}^{(t+1)}(a_2)$ and $q_{a_1,a_2}^{(t+1)}(a_3)$ in Equation (21) are approximately calculated by substituting

$$\begin{aligned} &\Pr\big\{ P^{(t+1)} = (\text{type-}a_1, \ldots, \text{type-}a_r) \text{ for some } a_{k+1}, \ldots, a_r \big\} \\ &\approx \sum_{i_1 \neq n} \cdots \sum_{i_r \neq n} x_{i_1}^{(t)} q_{i_1}^{(t)}(i_2) q_{i_1,i_2}^{(t)}(i_3) \cdots q_{i_1,i_2}^{(t)}(i_r) \\ &\qquad \prod_{l=1}^{k} \left(\frac{f_{i_1} m_{a_l,i_1} + f_{i_2} m_{a_l,i_2} + \cdots + f_{i_r} m_{a_l,i_r}}{f_{i_1} + f_{i_2} + \cdots + f_{i_r}} \right) \\ &\quad + \sum_{i_1,\ldots,i_r : n \in \{i_1,i_2,\ldots,i_r\}} \cdots \sum x_{i_1}^{(t)} q_{i_1}^{(t)}(i_2) q_{i_1,i_2}^{(t)}(i_3) \cdots q_{i_1,i_2}^{(t)}(i_r) \prod_{l=1}^{k} m_{a_l,n}. \end{aligned}$$



(a) The success probability.



(b) The expected population vector at generation t=14.

Figure 3: Model with zeroth-order approximation versus exact model on the one-max problem, under conditions string_length=10, mutation_rate=0.1, and population_size=2.

Note that the above assumptions dispense with $q^{(t)}_{i_1,i_2,i_3}$ through $q^{(t)}_{i_1,\ldots,i_{r-1}}$ for maintaining $x^{(t)}$.

EXPERIMENT 3.10 (Effect of Approximation; Onemax Prob., str.length $= 10$, pop.size $= 2$): Consider the onemax problem in Experiment 3.4. To see the effect of approximation, we first make an experiment similar to that of Experiment 3.4 upon the model with the zeroth-order approximation. By letting l(string length) $= 10$, letting p_{m}(mutation rate) $= 1/l = 0.1$, letting r(population size) $= 2$, making all the initial distributions be $\langle 1, 0, \ldots, 0\rangle^T$, and repetitively applying the update rules given by Equations (19) and (20) with the zeroth-order approximation given by Equation (23), we can observe how $x^{(t)}$ changes over time. To observe the success probability and the expected proportion of each type solution in $P^{(t)}$, we calculate $\sum \sum_{i=n \text{ or } j=n} x^{(t)}_i x^{(t)}_j$ and $(x^{(t)}_k + \sum_i x^{(t)}_i x^{(t)}_k)/2$ for each k, respectively. The results are shown in Figure 3(a) and 3(b), along with the graphs on the exact model (Experiment 3.4) for comparison purposes. From these figures, we see that the zeroth-order approximation leads to much faster convergence to the optimum. In order to observe the probability distribution over possible multiset populations {type-i, type-j}, we can calculate

$$\begin{cases} x^{(t)}_i q^{(t)}_i(j) + x^{(t)}_j q^{(t)}_j(i) & \text{if } i \neq j \\ x^{(t)}_i q^{(t)}_i(i) & \text{if } i = j \end{cases} \approx \begin{cases} x^{(t)}_i x^{(t)}_j + x^{(t)}_j x^{(t)}_i & \text{if } i \neq j \\ x^{(t)}_i x^{(t)}_i & \text{if } i = j \end{cases}$$

for each i and j. The result at generation 10 is shown in Figure 4, where probabilities of possible multiset populations {type-i, type-j} to occur on the zeroth-order approximate model are indicated by dashed line graphs and triangles, and those on our exact model are indicated by rod and line graphs for comparison purposes. Each point on the zeroth-order approximate model is plotted by a solid triangle (▲) when it clearly underestimates the exact probability (line graph), and is plotted by a triangle filled with oblique lines (△) when it clearly overestimates. According to Figure 4, in the approximate model, some population trajectories are wrongly attached little weight, and some others are wrongly attached great weight.

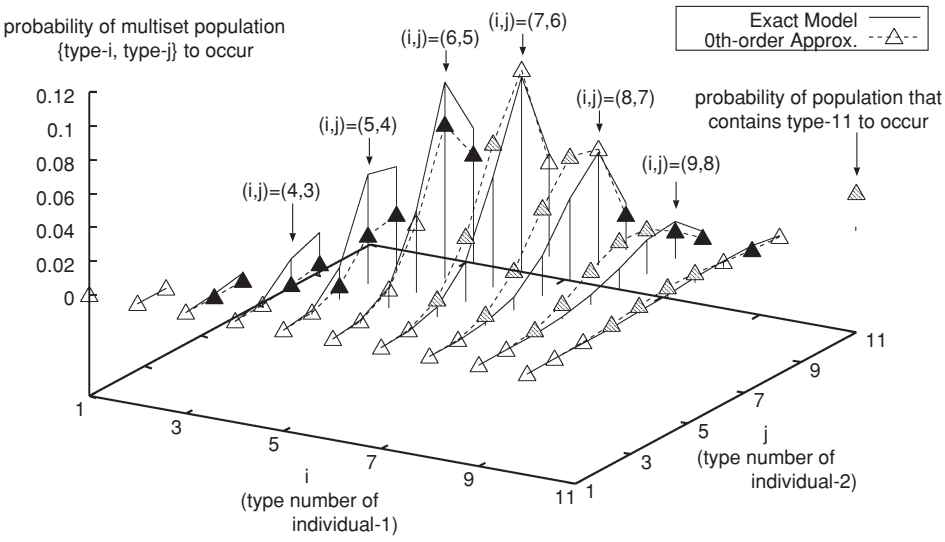
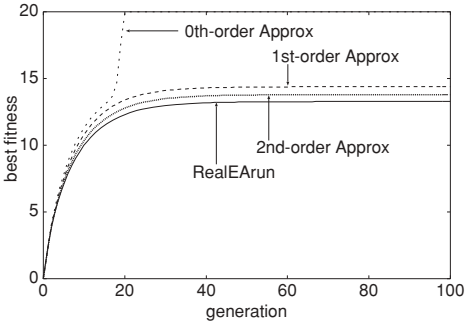
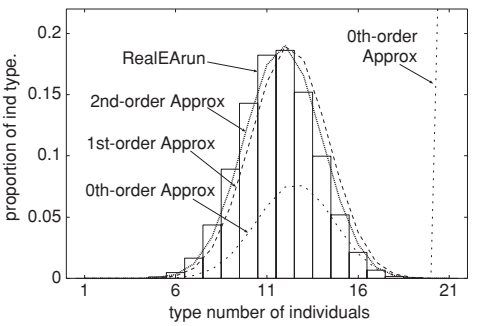


Figure 4: Comparing the zeroth-order approximate model with the exact model on the probability distribution over possible multiset populations at generation 10, when the models are applied to the onemax problem under conditions string_length=10, mutation_rate=0.1, and population_size=2.

EXPERIMENT 3.11 (Effect of Approximation; Onemax Prob., str.length $= 20$, pop.size $= 5$): In order to further study the effect of approximation, we let l(string length) $= 20$, let p_{m}(mutation rate) $= 1/l = 0.05$, let r(population size) $= 5$ and make experiments similar to that of Experiment 3.10 upon the models with zeroth- through second-order approximation. By making all the initial distributions be $\langle 1, 0, \ldots, 0\rangle^T$, and repetitively applying the update rules given by Equations (19), (20), and (21) with the zeroth-, first- or second-order approximation, we can observe how $x^{(t)}$ and the related $q^{(t)}_{i_1,\ldots,i_k}$ change over time. In order to observe the expected best fitness in $P^{(t)}$ under the zeroth-order approximation, we calculate $\sum_{i_1}\cdots\sum_{i_5} x^{(t)}_{i_1}\cdots x^{(t)}_{i_5}(\max\{i_1,\ldots,i_5\}-1)$; similarly, under the first-order approximation, we calculate $\sum_{i_1}\cdots\sum_{i_5} x^{(t)}_{i_1} q^{(t)}_{i_1}(i_2)\cdots q^{(t)}_{i_1}(i_5)\,(\max\{i_1,\ldots,i_5\}-1)$; under the second-order approximation, we calculate $\sum_{i_1}\cdots\sum_{i_5} x^{(t)}_{i_1} q^{(t)}_{i_1}(i_2) q^{(t)}_{i_1,i_2}(i_3)\cdots q^{(t)}_{i_1,i_2}(i_5)$ $(\max\{i_1,\ldots,i_5\}-1)$. In order to observe the expected proportion of each type solution in $P^{(20)}$ under the zeroth-, first-, or second-order approximation, we make calculations similar to that of Experiment 3.8 under the corresponding approximations. The results are shown in Figures 5(a) and 5(b), along with the graphs on the empirical results for comparison purposes. We see from these figures that as order number decreases, the approximation results in faster convergence to the optimum in the current instance of the onemax problem. It should be noted that the total number of possible (multiset) populations amounts to $N = \binom{5+21-1}{21-1} \approx 5.3\times 10^4$, so the Nix and Vose-style Markov chain model requires $N^2 \approx 2.8\times 10^9$ memory cells for transition matrix. On the other hand, the second-order approximation only requires $2(n+n^2+n^3)+n^2 \approx 2.0\times 10^4$ memory cells for $x^{(t)}$, $x^{(t+1)}$, $q^{(t)}_{a_1}$, $q^{(t+1)}_{a_1}$, $q^{(t)}_{a_1,a_2}$, $q^{(t+1)}_{a_1,a_2}$, and $m_{i,j}$, though it is seen to give a sufficiently good approximation.



(a) The expected best fitness.



(b) The expected population vector at generation t=20.

Figure 5: Models with zeroth- through second-order approximations versus empirical results on the onemax problem, under conditions string_length=20, mutation_rate=0.05, and population_size=5.

The above Experiments 3.10 and 3.11 demonstrate the importance of coexistence relations between individuals in the population for modeling the behavior of EAs.

4 Extension to Models for Finite Population EAs that Include Crossover

In this section, we extend our discussion so that EAs under consideration can include crossover. From now on, our discussion assumes the following EA:

$\mathcal{P}^{(0)} \leftarrow$ an initial ordered-tuple population of r individuals;
$t \leftarrow 0$;
repeat {
  $\mathcal{Q}^{(t)} \leftarrow$ an ordered-tuple (mating pool) of $2r$ individuals that are selected from $\mathcal{P}^{(t)}$ through fitness-proportionate scheme;
  $\mathcal{P}^{(t+1)} \leftarrow$ an ordered-tuple population of r individuals that are obtained by mating elements in $\mathcal{Q}^{(t)}$ with each other, and producing an offspring from each pair through a mixing operation;
  $t \leftarrow t+1$;
}

where we say "mixing" to mean any combination of crossover and mutation. Fitness-proportionate selection and a mixing operation are assumed to be alternately applied. As in the previous section, in order to observe the behavior of EAs and easily cope with the cumulative feature of the success probability, we also consider random variables $P^{(t)}$ defined as follows:

$$P^{(t)} = \begin{cases} \mathcal{P}^{(t)} & \text{if } t = 0 \text{ or } P^{(t\text{-}1)} \text{ does not contain any optimal solution} \\ \text{OPT} & \text{otherwise,} \end{cases}$$

where we use OPT to denote a population that only contains multiple copies of some distinguished optimal solution. We also see that $\Pr\{P^{(t)} \text{ contains an optimal solution}\}$ describes the success probability at generation t. Suppose that the search space has n

elements, called type-1 through type-n candidate solutions, and that the type-n solution is the unique optimum constituting OPT. Let

$$m_{i,j}(a) = \begin{cases} \Pr\{\text{type-}a \text{ solution results from the mixing operation} \\ \qquad \text{based on parent types } i \text{ and } j\} & \text{if } i \neq n,\ j \neq n \\ 0 & \text{if } (i = n \text{ or } j = n),\ a \neq n \\ 1 & \text{if } (i = n \text{ or } j = n),\ a = n \end{cases}$$

and let M_a be an $n \times n$ matrix having $(i,\ j)$th entry $m_{i,j}(a)$. We also follow other notation in the previous section.

We first give extensions of Theorem 3.6 and Corollary 3.7.

THEOREM 4.1: *Consider a finite population EA, and suppose that the population size equals r. Then, for every $k \in \{1, 2, \ldots, r\}$ and every type numbers $a_1, \ldots, a_k$,*

$$\begin{aligned} &\Pr\{\textit{the individuals at address } 1, \ldots, k \textit{ in } P^{(t+1)} \textit{ are type-}a_1, \ldots, \textit{ type-}a_k \textit{ respectively}\} \\ &= \sum_{i_1 \neq n} \sum_{i_2 \neq n} \cdots \sum_{i_r \neq n} x^{(t)}_{i_1} q^{(t)}_{i_1}(i_2) \cdots q^{(t)}_{i_1,\ldots,i_{r-1}}(i_r) \prod_{l=1}^{k} \frac{\sum_{g=1}^{r} \sum_{h=1}^{r} f_{i_g} f_{i_h} m_{i_g,i_h}(a_l)}{(f_{i_1} + f_{i_2} + \cdots + f_{i_r})^2} \\ &\quad + \sum_{i_1,i_2,\ldots,i_r : n \in \{i_1,i_2,\ldots,i_r\}} \sum \cdots \sum x^{(t)}_{i_1} q^{(t)}_{i_1}(i_2) \cdots q^{(t)}_{i_1,\ldots,i_{r-1}}(i_r) \prod_{l=1}^{k} m_{n,n}(a_l). \end{aligned}$$

PROOF: This result can be proven by an argument similar to that of Theorem 3.2. We may calculate as follows:

$$\begin{aligned} &\Pr\left\{P^{(t+1)} = (\text{type-}a_1, \ldots, \text{type-}a_r) \text{ for some } a_{k+1}, \ldots, a_r\right\} \\ &= \sum_{i_1} \sum_{i_2} \cdots \sum_{i_r} \Pr\left\{P^{(t)} = (\text{type-}i_1, \ldots, \text{ type-}i_r)\right\} \\ &\qquad \times \Pr\{P^{(t+1)} = (\text{type-}a_1,\ldots,\text{type-}a_r) \text{ for some} \\ &\qquad\quad a_{k+1}, \ldots, a_r \mid P^{(t)} = (\text{type-}i_1,\ldots,\text{type-}i_r)\} \\ &= \sum_{i_1} \sum_{i_2} \cdots \sum_{i_r} x^{(t)}_{i_1} q^{(t)}_{i_1}(i_2) \cdots q^{(t)}_{i_1,\ldots,i_{r-1}}(i_r) \\ &\qquad \times \prod_{l=1}^{k} \Pr\{P^{(t+1)} = (\text{type-}a_1,\ldots,\text{type-}a_r) \text{ for some } a_1, \ldots, \\ &\qquad\quad a_{l\text{-}1}, a_{l+1}, \ldots, a_r \mid P^{(t)} = (\text{type-}i_1,\ldots,\text{type-}i_r)\} \end{aligned} \tag{26}$$

In order to proceed with the calculation, suppose that $P^{(t)} = (\text{type-}i_1,\ldots,\text{type-}i_r)$. Now, if $n \in \{i_1, \ldots, i_r\}$, it follows from the definition of $P^{(t)}$ that $P^{(t+1)} = \text{OPT} = (\text{type-}n,\ldots,\text{type-}n)$. Therefore,

$$\begin{aligned} &\Pr\left\{P^{(t+1)} = (\text{type-}a_1,\ldots,\text{type-}a_r) \text{ for some } a_1, \ldots, a_{l-1}, a_{l+1}, \ldots, a_r \right. \\ &\qquad\qquad \left. \mid P^{(t)} = (\text{type-}i_1,\ldots,\text{type-}i_r),\ n \in \{i_1, \ldots, i_r\}\right\} \\ &= \begin{cases} 1 & \text{if } a_l = n \\ 0 & \text{otherwise} \end{cases} \\ &= m_{n,n}(a_l). \end{aligned} \tag{27}$$

On the other hand, if $i_1 \neq n, i_2 \neq n, \ldots, i_r \neq n$, then the population $P^{(t)}(= \mathcal{P}^{(t)})$ does not contain the optimum, and so fitness-proportionate selection and the mixing operation are applied to produce individuals in $\mathcal{P}^{(t+1)}(= P^{(t+1)})$. Therefore,

$$\begin{aligned}
&\Pr\{P^{(t+1)} = (\text{type-}a_1,\ldots,\text{type-}a_r) \text{ for some } a_1,\ldots,a_{l-1},a_{l+1},\ldots,a_r \\
&\qquad\qquad \mid P^{(t)} = (\text{type-}i_1,\ldots,\text{type-}i_r), i_1 \neq n,\ldots,i_r \neq n\} \\
&= \sum_{g=1}^{r}\sum_{h=1}^{r} \Pr\{\text{type-}i_g \text{ and type-}i_h \text{ parents are selected} \\
&\qquad\qquad \text{for producing the individual at address } l \text{ in } P^{(t+1)} \\
&\qquad\qquad \mid P^{(t)} = (\text{type-}i_1,\ldots,\text{type-}i_r), i_1 \neq n,\ldots,i_r \neq n\} \\
&\quad \times \Pr\{\text{the individual at address } l \text{ in } P^{(t+1)} \text{ is type-}a_l \\
&\qquad\qquad \mid P^{(t)} = (\text{type-}i_1,\ldots,\text{type-}i_r), i_1 \neq n,\ldots,i_r \neq n, \\
&\qquad\qquad \text{type-}i_g \text{ and type-}i_h \text{ parents are selected} \\
&\qquad\qquad \text{for producing the individual at address } l \text{ in } P^{(t+1)}\} \\
&= \sum_{g=1}^{r}\sum_{h=1}^{r} \left(\frac{f_{i_g}}{f_{i_1}+\cdots+f_{i_r}}\right)\left(\frac{f_{i_h}}{f_{i_1}+\cdots+f_{i_r}}\right) m_{i_g,i_h}(a_l). \qquad (28)
\end{aligned}$$

By substituting Equations (27) and (28) into Equation (26), we can obtain the required result. □

COROLLARY 4.2: *Consider a finite population EA, and suppose that the population size equals r. Then, for every noninitial generation t, every $k \in \{1, 2, \ldots, r\}$, every type numbers $a_1, a_2, \ldots, a_k$ and $b_1, b_2, \ldots, b_k$, if $\{a_1, a_2, \ldots, a_k\}$ is equal to $\{b_1, b_2, \ldots, b_k\}$ as multisets, that is, if the sequence $a_1a_2\cdots a_k$ can be obtained from $b_1b_2\cdots b_k$ by rearrangement, then*

$$x^{(t)}_{a_1} q^{(t)}_{a_1}(a_2)\cdots q^{(t)}_{a_1,\ldots,a_{k-1}}(a_k) = x^{(t)}_{b_1} q^{(t)}_{b_1}(b_2)\cdots q^{(t)}_{b_1,\ldots,b_{k-1}}(b_k).$$

PROOF: This can be proven by the same argument as in the proof of Corollary 3.7. □

Based on Theorem 4.1, we can give update rules that govern how probability distributions $x^{(t)}, q^{(t)}_{i_1}, q^{(t)}_{i_1,i_2}, \ldots, q^{(t)}_{i_1,\ldots,i_{r-1}}$ change over time.

Update Rule for $\boldsymbol{x^{(t)} = \langle x_1^{(t)}, \ldots, x_n^{(t)}\rangle^T}$ to Change. From Theorem 4.1, we have a rule

$$\begin{aligned}
x^{(t+1)}_a = &\sum_{i_1\neq n}\sum_{i_2\neq n}\cdots\sum_{i_r\neq n} x^{(t)}_{i_1} q^{(t)}_{i_1}(i_2)\cdots q^{(t)}_{i_1,\ldots,i_{r-1}}(i_r) \frac{\sum_{g=1}^{r}\sum_{h=1}^{r} f_{i_g} f_{i_h} m_{i_g,i_h}(a)}{(f_{i_1}+f_{i_2}+\cdots+f_{i_r})^2} \\
&+ \sum_{i_1,i_2,\ldots,i_r : n\in\{i_1,i_2,\ldots,i_r\}}\sum\cdots\sum x^{(t)}_{i_1} q^{(t)}_{i_1}(i_2)\cdots q^{(t)}_{i_1,\ldots,i_{r-1}}(i_r) m_{n,n}(a) \qquad (29)
\end{aligned}$$

It should be noted that unlike the crossoverless case in Section 3, we cannot separate the specific effects of selection and mixing operations. In the same way as in Section 3, we can calculate $y^{(t)} = \langle y_1^{(t)}, \ldots, y_n^{(t)} \rangle^T$ from $x^{(t)}, q_{i_1}^{(t)}, \ldots, q_{i_1,\ldots,i_{r-1}}^{(t)}$, where $y_k^{(t)}$ is defined as the probability of the type-k solution occurring at address 1 in

$$Q^{(t)} = \begin{cases} \mathcal{Q}^{(t)} & \text{if } P^{(t)} \text{ does not contain any optimal solution} \\ \mathrm{OPT} & \text{otherwise.} \end{cases}$$

However, if the EAs under consideration include crossover and maintain finite population, we cannot derive an equation $x^{(t+1)} = \langle y^{(t)T} M_1 y^{(t)}, \ldots, y^{(t)T} M_n y^{(t)} \rangle^T$, which does not correctly implement the requirement that mated parents should be selected from a common instance of $P^{(t)}$; note that $y^{(t)}$ does not represent a particular population, now.

Update Rule for $q_{a_1,\ldots,a_{k-1}}^{(t)} = \langle q_{a_1,\ldots,a_{k-1}}^{(t)}(1), \ldots, q_{a_1,\ldots,a_{k-1}}^{(t)}(n) \rangle^T$ to Change. From Theorem 4.1, we have a rule

$$q_{a_1,\ldots,a_{k-1}}^{(t+1)}(a_k) = \frac{\Pr\{ P^{(t+1)} = (\text{type-}a_1, \ldots, \text{type-}a_r) \text{ for some } a_{k+1}, \ldots, a_r \}}{\Pr\{ P^{(t+1)} = (\text{type-}a_1, \ldots, \text{type-}a_r) \text{ for some } a_k, a_{k+1}, \ldots, a_r \}}, \tag{30}$$

where

$$\begin{aligned} &\Pr\{ P^{(t+1)} = (\text{type-}a_1, \ldots, \text{type-}a_r) \text{ for some } a_{k+1}, \ldots, a_r \} \\ &= \sum_{i_1 \neq n} \sum_{i_2 \neq n} \cdots \sum_{i_r \neq n} x_{i_1}^{(t)} q_{i_1}^{(t)}(i_2) \cdots q_{i_1,\ldots,i_{r-1}}^{(t)}(i_r) \prod_{l=1}^{k} \frac{\sum_{g=1}^{r} \sum_{h=1}^{r} f_{i_g} f_{i_h} m_{i_g,i_h}(a_l)}{(f_{i_1} + f_{i_2} + \cdots + f_{i_r})^2} \\ &\quad + \sum_{i_1,i_2,\ldots,i_r : n \in \{i_1,i_2,\ldots,i_r\}} \sum \cdots \sum x_{i_1}^{(t)} q_{i_1}^{(t)}(i_2) \cdots q_{i_1,\ldots,i_{r-1}}^{(t)}(i_r) \prod_{l=1}^{k} m_{n,n}(a_l). \end{aligned}$$

Approximate Rules. We can also obtain approximate rules in the same way as in Section 3.3.

5 Discussion

5.1 Diagonalization of Selection Matrices

Remember that the effect of fitness-proportionate selection can be described by a linear transformation of the form

$$\left\langle y_1^{(t)}, \ldots, y_n^{(t)} \right\rangle^T = S^{(t)} \left\langle x_1^{(t)}, \ldots, x_n^{(t)} \right\rangle^T,$$

where $S^{(t)}$ is an $n \times n$ matrix, $y_k^{(t)}$ is defined to be the probability of the type-k solution occurring at address 1 in $Q^{(t)}$, and $x_k^{(t)}$ is defined to be the probability of the type-k solution occurring at address 1 in $P^{(t)}$. When the population size r equals 2, we have

obtained the equation

$$y_k^{(t)} = \begin{cases} \sum_{j \neq n} \frac{f_k}{f_k + f_j} \left(x_k^{(t)} q_k^{(t)}(j) + x_j^{(t)} q_j^{(t)}(k) \right) & \text{if } k \neq n \\ x_n^{(t)} + \sum_{j \neq n} x_j^{(t)} q_j^{(t)}(n) & \text{if } k = n, \end{cases} \tag{31}$$

in the process of deriving Equation (14). Now, since Corollary 3.3 says that $x_a^{(t)} q_a^{(t)}(b) = x_b^{(t)} q_b^{(t)}(a)$ for every type number a, b and noninitial generation t, when $t > 0$ Equation (31) can be rewritten as follows:

$$y_k^{(t)} = \begin{cases} \left(2 \sum_{j \neq n} \frac{f_k}{f_k + f_j} q_k^{(t)}(j) \right) x_k^{(t)} & \text{if } k \neq n \\ \left(1 + \sum_{j \neq n} q_n^{(t)}(j) \right) x_n^{(t)} & \text{if } k = n. \end{cases}$$

Thus, we can derive the following diagonalization result.

Diagonalization of the Selection Matrix in the Exact Model for Population Size 2. When the population size equals 2 and generation $t > 0$, the selection matrix $S^{(t)} = [s_{i,j}^{(t)}]$ in Equation (14) can be written as a diagonal matrix:

$$s_{i,j}^{(t)} = \begin{cases} 2 \sum_{a \neq n} \frac{f_i}{f_i + f_a} q_i^{(t)}(a) & \text{if } i = j < n \\ 1 + \sum_{a \neq n} q_n^{(t)}(a) & \text{if } i = j = n, \\ 0 & \text{otherwise.} \end{cases} \tag{32}$$

Similarly, we can derive the following diagonalization results for the general exact model in Section 3.2 and the approximate models in Section 3.3.

Diagonalization of the Selection Matrix in the General Exact Model. When the population size equals r and generation $t > 0$, by using Corollary 3.7, Equation (18) can be rewritten as

$$y_k^{(t)} = \begin{cases} r \sum_{j_1 \neq n} \sum_{j_2 \neq n} \cdots \sum_{j_{r\text{-}1} \neq n} \frac{f_k}{f_k + f_{j_1} + f_{j_2} + \cdots + f_{j_{r\text{-}1}}} \\ \qquad x_k^{(t)} q_k^{(t)}(j_1) q_{k,j_1}^{(t)}(j_2) \cdots q_{k,j_1,\ldots,j_{r\text{-}2}}^{(t)}(j_{r\text{-}1}) & \text{if } k \neq n \\ x_n^{(t)} \Bigg(1 + \sum_{j_1 \neq n} q_n^{(t)}(j_1) \Bigg(1 + \sum_{j_2 \neq 11} q_{n,j_1}^{(t)}(j_2) \\ \qquad \Bigg(\cdots \Bigg(1 + \sum_{j_{r\text{-}2} \neq n} q_{n,j_1,\ldots,j_{r\text{-}3}}^{(t)}(j_{r\text{-}2}) \\ \qquad\qquad \Bigg(1 + \sum_{j_{r\text{-}1} \neq n} q_{n,j_1,\ldots,j_{r\text{-}2}}^{(t)}(j_{r\text{-}1}) \Bigg) \Bigg) \cdots \Bigg) \Bigg) \Bigg) & \text{if } k = n, \end{cases}$$

and so the selection matrix $S^{(t)}=[s^{(t)}_{i,j}]$ in Equation (19) can be written as a diagonal matrix:

$$s^{(t)}_{i,j} = \begin{cases} r \sum_{j_1\neq n}\sum_{j_2\neq n}\cdots\sum_{j_{r\text{-}1}\neq n} \dfrac{f_i}{f_i+f_{j_1}+f_{j_2}+\cdots+f_{j_{r\text{-}1}}} \\ \qquad\qquad q^{(t)}_i(j_1)q^{(t)}_{i,j_1}(j_2)\cdots q^{(t)}_{i,j_1,\ldots,j_{r\text{-}2}}(j_{r\text{-}1}) & \text{if } i=j<n \\ 1+\sum_{j_1\neq n} q^{(t)}_n(j_1) \\ \quad \Bigg(1+\sum_{j_2\neq n} q^{(t)}_{n,j_1}(j_2) \\ \qquad \Bigg(1+\sum \cdots\cdots\cdots\cdots \\ \qquad\quad \Bigg(1+\sum_{j_{r\text{-}2}\neq n} q^{(t)}_{n,j_1,\ldots,j_{r\text{-}3}}(j_{r\text{-}2}) \\ \qquad\qquad \Bigg(1+\sum_{j_{r\text{-}1}\neq n} q^{(t)}_{n,j_1,\ldots,j_{r\text{-}2}}(j_{r\text{-}1})\Bigg)\Bigg)\cdots\Bigg)\Bigg) & \text{if } i=j=n \\ 0 & \text{otherwise.} \end{cases} \tag{33}$$

Diagonalization of the Selection Matrix under the First-Order Approximation. When the population size equals r, generation $t>0$ and the first-order approximation is assumed, by using Corollary 3.7, Equation (24) can be rewritten as

$$y^{(t)}_k \approx \begin{cases} x^{(t)}_k \sum_{j_1\neq n}\sum_{j_2\neq n}\cdots\sum_{j_{r\text{-}1}\neq n} \dfrac{f_k}{f_k+f_{j_1}+f_{j_2}+\cdots+f_{j_{r\text{-}1}}} q^{(t)}_k(j_1) \\ \qquad\qquad \times\left(q^{(t)}_k(j_2)\cdots q^{(t)}_k(j_{r\text{-}1})+(r-1)q^{(t)}_{j_1}(j_2)\cdots q^{(t)}_{j_1}(j_{r\text{-}1})\right) & \text{if } k\neq n \\ x^{(t)}_n\Bigg(1+\sum_{j_1\neq n} q^{(t)}_n(j_1)\Bigg(1+\sum_{j_2\neq n} q^{(t)}_{j_1}(j_2) \\ \qquad \Bigg(\cdots\Bigg(1+\sum_{j_{r\text{-}2}\neq n} q^{(t)}_{j_1}(j_{r\text{-}2})\Bigg(1+\sum_{j_{r\text{-}1}\neq n} q^{(t)}_{j_1}(j_{r\text{-}1})\Bigg)\Bigg)\cdots\Bigg)\Bigg)\Bigg) & \text{if } k=n, \end{cases}$$

and so the selection matrix $S^{(t)}=[s^{(t)}_{i,j}]$ in Equation (19) can be approximated by a diagonal matrix:

$$s^{(t)}_{i,j} \approx \begin{cases} \sum_{j_1\neq n}\sum_{j_2\neq n}\cdots\sum_{j_{r\text{-}1}\neq n} \dfrac{f_i}{f_i+f_{j_1}+f_{j_2}+\cdots+f_{j_{r\text{-}1}}} q^{(t)}_i(j_1) \\ \qquad\qquad \times\left(q^{(t)}_i(j_2)\cdots q^{(t)}_i(j_{r\text{-}1})+(r-1)q^{(t)}_{j_1}(j_2)\cdots q^{(t)}_{j_1}(j_{r\text{-}1})\right) & \text{if } i=j<n \\ 1+\sum_{j_1\neq n} q^{(t)}_n(j_1)\Bigg(1+\sum_{j_2\neq n} q^{(t)}_{j_1}(j_2) \\ \quad \Bigg(\cdots\Bigg(1+\sum_{j_{r\text{-}2}\neq n} q^{(t)}_{j_1}(j_{r\text{-}2})\Bigg(1+\sum_{j_{r\text{-}1}\neq n} q^{(t)}_{j_1}(j_{r\text{-}1})\Bigg)\Bigg)\cdots\Bigg)\Bigg) & \text{if } i=j=n \\ 0 & \text{otherwise.} \end{cases} \tag{34}$$

Diagonalization of the Selection Matrix under the Second-Order Approximation. When the population size equals r, generation $t > 0$ and the second-order approximation is assumed, by using Corollary 3.7, Equation (25) can be rewritten as

$$y_k^{(t)} \approx \begin{cases} x_k p^{(t)} \displaystyle\sum_{j_1\neq n}\sum_{j_2\neq n}\cdots\sum_{j_{r\text{-}1}\neq n} \frac{f_k}{f_k+f_{j_1}+f_{j_2}+\cdots+f_{j_{r\text{-}1}}} q_k^{(t)}(j_1) \\ \qquad \times \big(q_{k,j_1}^{(t)}(j_2)\cdots q_{k,j_1}^{(t)}(j_{r\text{-}1}) + q_{j_1,k}^{(t)}(j_2)\cdots q_{j_1,k}^{(t)}(j_{r\text{-}1}) \\ \qquad\quad + (r-2) q_{k,j_1}^{(t)}(j_2) q_{j_1,j_2}^{(t)}(j_3)\cdots q_{j_1,j_2}^{(t)}(j_{r\text{-}1})\big) & \text{if } k \neq n \\ x_n^{(t)}\Bigg(1+\displaystyle\sum_{j_1\neq n} q_n^{(t)}(j_1)\Bigg(1+\sum_{j_2\neq n} q_{n,j_1}^{(t)}(j_2)\Bigg(1+\sum_{j_3\neq n} q_{j_1,j_2}^{(t)}(j_3) \\ \qquad \times\Bigg(\cdots\Bigg(1+\displaystyle\sum_{j_{r-2}\neq n} q_{j_1,j_2}^{(t)}(j_{r\text{-}2})\Bigg(1+\sum_{j_{r-1}\neq n} q_{j_1,j_2}^{(t)}(j_{r\text{-}1})\Bigg)\Bigg)\cdots\Bigg)\Bigg)\Bigg)\Bigg) & \text{if } k = n, \end{cases}$$

and so the selection matrix $S^{(t)}=[s_{i,j}^{(t)}]$ in Equation (19) can be approximated by a diagonal matrix:

$$s_{i,j}^{(t)} \approx \begin{cases} \displaystyle\sum_{j_1\neq n}\sum_{j_2\neq n}\cdots\sum_{j_{r\text{-}1}\neq n} \frac{f_i}{f_i+f_{j_1}+f_{j_2}+\cdots+f_{j_{r\text{-}1}}} q_i^{(t)}(j_1) \\ \qquad \times \big(q_{i,j_1}^{(t)}(j_2)\cdots q_{i,j_1}^{(t)}(j_{r\text{-}1}) + q_{j_1,i}^{(t)}(j_2)\cdots q_{j_1,i}^{(t)}(j_{r\text{-}1}) \\ \qquad\quad + (r-2) q_{i,j_1}^{(t)}(j_2) q_{j_1,j_2}^{(t)}(j_3)\cdots q_{j_1,j_2}^{(t)}(j_{r\text{-}1})\ \big) & \text{if } i = j < n \\ 1+\sum_{j_1\neq n} q_n^{(t)}(j_1)\big(1+\sum_{j_2\neq n} q_{n,j_1}^{(t)}(j_2)\Big(1+\sum_{j_3\neq n} q_{j_1,j_2}^{(t)}(j_3) \\ \qquad \times\Bigg(\cdots\Bigg(1+\sum_{j_{r-2}\neq n} q_{j_1,j_2}^{(t)}(j_{r\text{-}2})\Bigg(1+\sum_{j_{r-1}\neq n} q_{j_1,j_2}^{(t)}(j_{r\text{-}1})\Bigg)\Bigg)\cdots\Bigg)\Bigg) & \text{if } i = j = n \\ 0 & \text{otherwise.} \end{cases} \tag{35}$$

Note that in the zeroth-order approximation, a diagonal selection matrix is already given by Equation (23). The diagonalization of selection matrices brings us to apply the update rule given by Equation (19) more efficiently. Furthermore, if a selection matrix $S^{(t)}=[s_{i,j}^{(t)}]$ is diagonalized and satisfies $y_k^{(t)} = s_{k,k}^{(t)} x_k^{(t)}$ for every k, it follows that its kth diagonal element $s_{k,k}^{(t)}$ can be considered to be an indicator of the expected pressure on the type-k individual to be selected. Through the diagonalization of a selection matrix, therefore, we can show the associated selection pressure on each candidate solution in a form of expression. This result can be utilized to theoretically observe how the selection pressure on each candidate solution varies over time, and to reconfirm well-known effects about selection, as demonstrated below.

EXPERIMENT 5.1 (Selection Pressure; onemax prob., str.length = 10, pop.size = 5): In Experiment 3.8, under conditions l(string length) = 10, p_{m}(mutation rate) = $1/l = 0.1$ and r(population size) = 5, we have repetitively applied the update rules given by Equations (19), (20), and (21) to see the expected behavior of the crossoverless EA on the onemax problem. The same expected behavior can also be approximated by repetitively

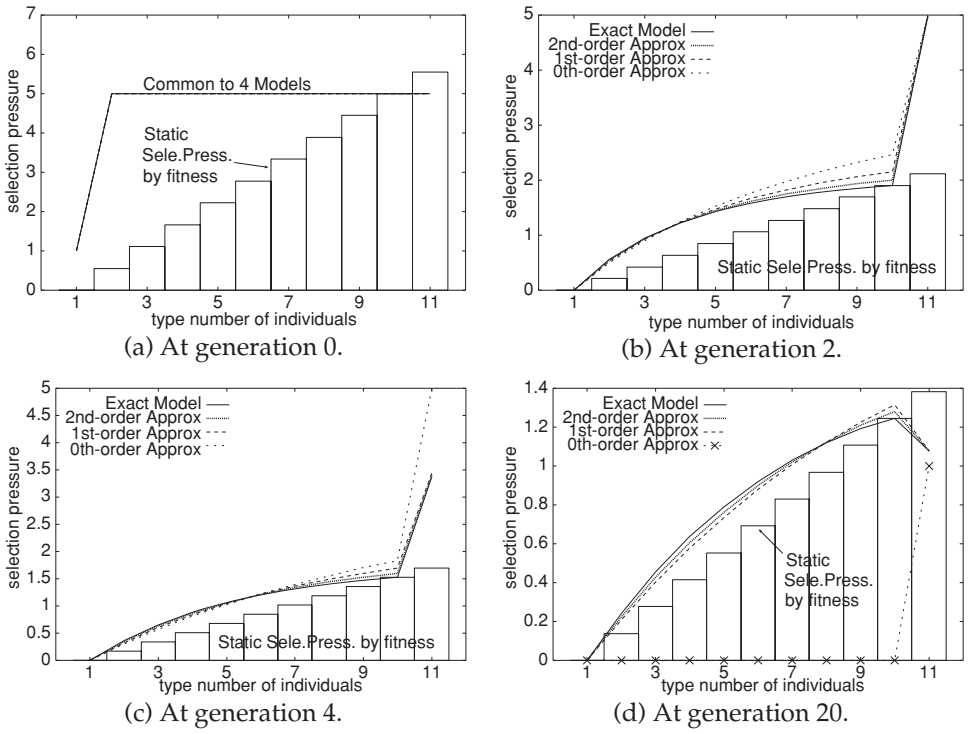


Figure 6: Distributions of the expected selection pressure obtained from the exact model and the three approximate models, on the onemax problem under conditions string length=10, mutation rate=0.1, and population size=5.

applying the update rules given by Equations (19), (20), and (21) with the zeroth-, first-, or second-order approximation, in a similar way as in Experiment 3.11. In each of these simulated evolutions, we can observe the expected selection pressure on each type individual through the diagonal selection matrices given in Equations (33), (23), (34), and (35). The results at generation 0, 2, 4, and 20 are shown in Figures 6(a), 6(b), 6(c), and 6(d) respectively. In each figure, line graphs obtained by connecting points in $\{(k, s_{kk})|k$ is a type number, s_{kk} is the kth diagonal element of a selection matrix$\}$ are depicted for the exact model and the three approximate models, together with a bar graph of fitness f_i scaled by some constant multiplier for comparison purposes. Note that the diagonal selection matrices in Equations (33), (23), (34), and (35) can also be used at generation 0, because the condition $x^{(t)}_{a_1} q^{(t)}_{a_1}(a_2) \cdots q^{(t)}_{a_1,\ldots,a_{k-1}}(a_k) = x^{(t)}_{b_1} q^{(t)}_{b_1}(b_2) \cdots q^{(t)}_{b_1,\ldots,b_{k-1}}(b_k)$ in Corollary 3.7 is also satisfied at generation 0 by the initial distributions of Experiments 3.8 and 3.11 provided that $\{a_1, a_2, \ldots, a_k\}$ is equal to $\{b_1, b_2, \ldots, b_k\}$ as multisets. In each population trajectory, fitness-proportionate selection scheme certainly exerts fitness-proportionate selection pressure on each individual in the current population. However, we see from Figure 6 that on the average over all trajectories, there is no such strong bias toward selecting the higher fitness individuals as the fitness landscape suggests. The explanation for this is that due to the finiteness of the population and the effect of selection operation, populations in possible trajectories are apt to lose some fraction of diversity and so have some undispersed distribution of fitness.

5.2 Comparison with Nix and Vose-style Markov Chain Model

The Markov chain model is a theoretical model as well. To understand the probabilistic behavior of an EA, we can theoretically run its stochastic process through the corresponding Markov chain model as well as the models described in Section 4. So, we now compare the models in Section 4 with the Nix and Vose-style Markov chain model on the basis of time and space complexities of procedures for theoretically running the stochastic search process through the models themselves. Let n be the size of the search space, and let r be the population size. Then the total number of possible multiset populations is given by $N = \binom{r+n-1}{n-1}$.

First, we determine the space complexities. Let $S_4(n, r)$, $S_{4,\mathrm{approx}}(n, r, d)$ and $S_{\mathrm{mc}}(n, r)$ be the numbers of required memory cells when we use the exact model in Section 4, the dth-order approximate model in Section 4 and the Nix and Vose-style Markov chain model respectively. In order to run the stochastic search process through the exact model in Section 4 theoretically, we need the space for maintaining the current distributions $x^{(t)}, q_{a_1}^{(t)}, \ldots, q_{a_1,\ldots,a_{r-1}}^{(t)}$, the space for storing the next distributions $x^{(t+1)}, q_{a_1}^{(t+1)}, \ldots, q_{a_1,\ldots,a_{r-1}}^{(t+1)}$, the space for constants $m_{i,j}(a)$ and f_i, and some working space for calculating the next distributions. Since each distribution has n entries, and since we must construct $\sum_{k=0}^{r-1} n^k$ distributions at each generation, it follows that the number of required memory cells for storing the distributions is $n \times \sum_{k=0}^{r-1} n^k \times 2 = 2\sum_{k=1}^{r} n^k$. Obviously, we use n^3 and n memory cells for storing $m_{i,j}(a)$ and f_i, respectively. For obtaining the next distributions, additional $O(r)$ memory cells are enough. Thus, the total number of required memory cells is given by $S_4(n, r) = 2\sum_{k=1}^{r} n^k + n^3 + n + O(r) \approx 2\frac{n^{r+1}-1}{n-1} + n^3 + n$. Likewise, to run the stochastic search process through the dth-order approximate model in Section 4, we only need the space for maintaining the current distributions $x^{(t)}, q_{a_1}^{(t)}, \ldots, q_{a_1,\ldots,a_d}^{(t)}$, the space for storing the next distributions $x^{(t+1)}, q_{a_1}^{(t+1)}, \ldots, q_{a_1,\ldots,a_d}^{(t+1)}$, the space for constants $m_{i,j}(a)$ and f_i, and some working space for calculating the next distributions. Therefore, the total number of required memory cells is given by $S_{4,\mathrm{approx}}(n, r, d) = 2\sum_{k=1}^{d+1} n^k + n^3 + n + O(r) \approx 2\frac{n^{d+2}-1}{n-1} + n^3 + n \approx S_4(n, d+1)$. On the other hand, to run the stochastic search process through the Nix and Vose-style Markov chain model theoretically, we need the space for maintaining the current distribution over possible populations, the space for storing the next distribution, the space for the transition matrix, and some working space for calculating the next distribution. Obviously, we use $2N$ and N^2 memory cells for storing the distributions and the transition matrix respectively. For obtaining the next distribution, additional $O(1)$ memory cells are enough. Thus, the total number of required memory cells is given by $S_{\mathrm{mc}}(n, r) = 2N + N^2 + O(1) \approx N^2 + 2N$.

Tables 1(a) and 1(b) show how the values of $\log_{10} S_4(n, r)$ and $\log_{10} S_{\mathrm{mc}}(n, r)$ vary over all combinations of $n \in \{10, 10^2, 10^3, 10^6, 10^9, 10^{12}\}$ and $r \in \{1, 2, \ldots, 10, 15, 20\}$ respectively. We see from these tables that if $2 \le r \ll n$, then $S_4(n, r) \ll S_{\mathrm{mc}}(n, r)$. This can roughly be seen as follows. Suppose $3 \le r \le \sqrt{n} - 1$. Then, since $N = \binom{r+n-1}{n-1} = \frac{(n+r-1)(n+r-2)\cdots n}{r!} \approx \frac{n^r}{r!}$, and since $(r+1-k)k \le ((r+1)/2)^2$ for every k, it follows that $S_{\mathrm{mc}}(n, r) \approx N^2 + 2N \approx \frac{n^{2r}}{(r!)^2} + 2\frac{n^r}{r!} = n^{2r}/(\prod_{k=1}^{r}(r+1-k)k) + 2\frac{n^r}{r!} \ge \frac{n^{2r}}{((r+1)/2)^{2r}} + 2\frac{n^r}{r!} = 4^r n^r (\frac{n}{(r+1)^2})^r + 2\frac{n^r}{r!} \ge 4^r n^r + 2\frac{n^r}{r!} \ge 2\frac{n^{r+1}-1}{n-1} + n^3 + n \approx S_4(n, r)$.

Next, we determine the time complexities. Let $T_4(n, r)$, $T_{4,\mathrm{approx}}(n, r, d)$ and $T_{\mathrm{mc}}(n, r)$ be the numbers of computational steps needed per generation when we use the exact model in Section 4, the dth-order approximate model in Section 4 and the Nix and

Table 1: (a) the number of required memory cells when we use the model in Section 4 versus (b) the number of required memory cells when we use the Markov chain model.

(a) $\log_{10} S_4(n, r)$

	n (size of the search space)					
r	10	10^2	10^3	10^6	10^9	10^{12}
1	3.0	6.0	9.0	18.0	27.0	36.0
2	**3.1**	**6.0**	**9.0**	**18.0**	**27.0**	**36.0**
3	**3.5**	**6.5**	**9.5**	**18.5**	**27.5**	**36.5**
4	**4.4**	**8.3**	**12.3**	**24.3**	**36.3**	**48.3**
5	**5.3**	**10.3**	**15.3**	**30.3**	**45.3**	**60.3**
6	**6.3**	**12.3**	**18.3**	**36.3**	**54.3**	**72.3**
7	**7.3**	**14.3**	**21.3**	**42.3**	**63.3**	**84.3**
8	**8.3**	**16.3**	**24.3**	**48.3**	**72.3**	**96.3**
9	**9.3**	**18.3**	**27.3**	**54.3**	**81.3**	**108.3**
10	10.3	**20.3**	**30.3**	**60.3**	**90.3**	**120.3**
15	15.3	**30.3**	**45.3**	**90.3**	**135.3**	**180.3**
20	20.3	**40.3**	**60.3**	**120.3**	**180.3**	**240.3**

(b) $\log_{10} S_{\mathrm{mc}}(n, r)$

	n (size of the search space)					
r	10	10^2	10^3	10^6	10^9	10^{12}
1	2.1	4.0	6.0	12.0	18.0	24.0
2	3.5	7.4	11.4	23.4	35.4	47.4
3	4.7	10.5	16.4	34.4	52.4	70.4
4	5.7	13.3	21.2	45.2	69.2	93.2
5	6.6	15.9	25.9	55.8	85.8	115.8
6	7.4	18.4	30.3	66.3	102.3	138.3
7	8.1	20.8	34.6	76.6	118.6	160.6
8	8.8	23.0	38.8	86.8	134.8	182.8
9	9.4	25.2	42.9	96.9	150.9	204.9
10	**9.9**	27.3	46.9	106.9	166.9	226.9
15	**12.2**	36.6	65.9	155.8	245.8	335.8
20	**14.0**	44.8	83.4	203.2	323.2	443.2

Vose-style Markov chain model, respectively. In order to obtain the next distributions $x^{(t+1)}, q_{a_1}^{(t+1)}, \ldots, q_{a_1,\ldots,a_{r-1}}^{(t+1)}$ with the exact model in Section 4, we need to follow the rules in Equations (29) and (30). Specifically, to obtain $x_a^{(t+1)}$, we calculate the right expression of Equation (29), so requiring $O(n^r \times r^2)$ computational steps. To obtain $q_{a_1,\ldots,a_{k-1}}^{(t+1)}(a_k)$, we calculate the right expression of Equation (30), so requiring $O(n^r \times r^2 k)$ steps. Thus, for obtaining all the next distributions, $T_4(n, r) = \sum_{k=1}^{r} O(n^r r^2 k) \times n^k = O(r^3 n^{2r})$ steps are enough. In order to proceed with the stochastic search process for one generation through the dth-order approximate model in Section 4, we only need to obtain the next (approximate) distributions $x^{(t+1)}, q_{a_1}^{(t+1)}, \ldots, q_{a_1,\ldots,a_d}^{(t+1)}$ through the approximated version of the rules in Equations (29) and (30). Since the approximated version of the rules in Equations (29) and (30) require as many computational steps as the rules in Equations (29) and (30) require, it follows that $T_{4,\mathrm{approx}}(n, r, d) = \sum_{k=1}^{d+1} O(n^r r^2 k) \times n^k = O(r^2 d n^{r+d+1})$. On the other hand, to obtain the next distribution with the Nix and Vose-style Markov chain model, we need to multiply the transition matrix by the current distribution over possible populations. Since the transition matrix is an $N \times N$ matrix and the current distribution is an N-dimensional vector, it follows that $T_{\mathrm{mc}}(n, r) = O(N^2)$. Since $\frac{n+k-1}{k} \leq n$ for each $k \in \{1, 2, \ldots, r\}$, we obtain $N = \binom{r+n-1}{n-1} = \frac{n+r-1}{r} \cdot \frac{n+r-2}{r-1} \cdot \frac{n+r-3}{r-2} \cdots \frac{n}{1} \leq n^r$, and hence $T_{\mathrm{mc}}(n, r) = O(N^2) \leq O(n^{2r}) < O(r^3 n^{2r}) = T_4(n, r)$. For sufficiently small $d \leq r - 1$, we also have $T_{4,\mathrm{approx}}(n, r, d) = O(r^2 d n^{r+d+1}) \leq O(\frac{n^{2r}}{(r!)^2}) \leq T_{\mathrm{mc}}(n, r)$.

Consequently, we can conclude that the exact model described in Section 4 uses less memory space at the expense of additional computational time in comparison with the Nix and Vose-style Markov chain model. Our exact model is not considered to be sufficiently practical as well as the Nix and Vose-style Markov chain model. However, Experiment 3.11 demonstrates the possibility of our approximate models being used to make reasonably accurate predictions of EAs in reasonable amounts of time. Furthermore, unlike the Nix and Vose-style Markov chain model, models in Section 3 and 4 do not need any effective enumeration of possible populations. If we use a model in Section 3 or 4, we can directly observe the interrelation between the event

of type-1 solution being in the current population through the event of type-n solution being in the current population.

6 Summary

We provided a review of two prevalent approaches to the understanding of EAs at microscopic level: Markov chain analysis and the dynamic systems approach. Markov chain analysis describes the behavior of a finite population EA in a probabilistically exact way, but requires enormous memory spaces even for modest problems. The dynamic systems approach mainly examines the direction and intensity of the force of evolution acting on populations, and collapses all aspects of EA behavior into a single population trajectory $p, \mathcal{G}(p), \mathcal{G}^2(p), \ldots$ which encodes the asymptotic behavior of EA. Therefore, this approach cannot be used to predict the transient behavior of finite population EAs. We have also shown that the effect of mutation on population vectors is represented as a linear transformation, while the effect of fitness-proportionate selection cannot be represented in such a simple way for finite population EAs.

Next, we presented a method for exactly modeling the behavior of finite population crossoverless EAs, and then extended our discussion so that EAs under consideration can include crossover. We saw that if the population size is greater than 1 and much less than the cardinality of the search space, our exact model requires considerably less memory space for theoretically running the stochastic search process of the original EA than the Nix and Vose-style Markov chain model. We also presented some approximate models that use still less memory space than the exact model. Furthermore, based on our models, we examined the selection pressure by fitness-proportionate selection, and observed that on the average over all population trajectories, there is no such strong bias toward selecting the higher fitness individuals as the fitness landscape suggests.

Acknowledgments

The author would like to thank the anonymous reviewers for their helpful comments.

References

Arora, S., Rabani, Y., and Vazirani, U. (1994). Simulating quadratic dynamical systems is Pspace-complete. In *Proceedings of the Twenty-sixth Annual ACM Symposium on the Theory of Computing*, pp. 459–467.

Aytug, H., and Koehler, G. J. (1996). Stopping criteria for finite length genetic algorithms. *INFORMS Journal on Computing*, 8(2):183–191.

Aytug, H., and Koehler, G. J. (2000). New stopping criterion for genetic algorithms. *European Journal of Operational Research*, 126:662–674.

Bäck, T., deGraaf, J. M., Kok, J. N., and Kosters, W. A. (1997). Theory of genetic algorithms. *Bulletin of the EATCS*, 63:161–192.

Beyer, H.-G., Schwefel, H.-P., and Wegener, I. (2002). How to analyse evolutionary algorithms. *Theoretical Computer Science*, 287:101–130.

Davis, T. E., and Principe, J. C. (1993). A Markov chain framework for the simple genetic algorithm. *Evolutionary Computation*, 1(3):269–288.

DeJong, K. A. (2006). *Evolutionary computation: A unified approach*. Cambridge, MA: MIT Press.

DeJong, K. A., Spears, W. M., and Gordon, D. F. (1995). Using Markov chains to analyse GAFOs. In *Foundations of Genetic Algorithms 3*, pp. 115–137. San Mateo, CA: Morgan Kaufmann.

Droste, S., Jansen, T., and Wegener, I. (2002). On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, 276:51–81.

Eiben, A. E., Aarts, E. H. L., and VanHee, K. M. (1991). Global convergence of genetic algorithms: A Markov chain analysis. In *Parallel problem solving from nature*, Lecture notes in computer science (pp. 4–12). Berlin: Springer-Verlag.

Eiben, A. E., and Rudolph, G. (1999). Theory of evolutionary algorithms: A bird's eye view. *Theoretical Computer Science*, 229:3–9.

Eiben, A. E., and Smith, J. E. (2003). *Introduction to evolutionary computing*. Berlin: Springer-Verlag.

Goldberg, D. E., and Segrest, P. (1987). Finite Markov chain analysis of genetic algorithms. In J. J. Grefenstette (Ed.), *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms* (pp. 1–8). Mahwah, NJ: Lawrence Erlbaum.

Matsumoto, M., and Nishimura, T. (1998). Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation*, 8:3–30.

Mitavskiy, B., and Rowe, J. (2006a). An extension of Geiringer's theorem for a wide class of evolutionary search algorithms. *Evolutionary Computation*, 14(1):87–118.

Mitavskiy, B., and Rowe, J. (2006b). Some results about the Markov chain associated to GPs and general EAs. *Theoretical Computer Science*, 361:72–110.

Nix, A. E., and Vose, M. D. (1992). Modeling genetic algorithms with Markov chains. *Annals of Mathematics and Artificial Intelligence*, 5(1):79–88.

Poli, R., McPhee, N. F., and Rowe, J. E. (2004). Exact schemata theory and Markov chain models for genetic programming and variable-length genetic algorithms with homologous crossover. *Genetic Programming and Evolvable Machines*, 5(1):31–70.

Prügel-Bennett, A. (2003). Modeling finite populations. In *Foundations of Genetic Algorithms 7* (pp. 99–114). San Mateo, CA: Morgan Kaufmann.

Rabani, Y., Ravinovich, Y., and Sinclair, A. (1995). A computational view of population genetics. In *Proceedings of the Twenty-seventh Annual ACM Symposium on the Theory of Computing* (pp. 83–92). New York: ACM.

Reeves, C. R., and Rowe, J. E. (2003). *Genetic algorithms—Principles and perspective: A guide to GA theory*. Dordrecht, The Netherlands: Kluwer Academic.

Rudolph, G. (1994). Convergence analysis of canonical genetic algorithms. *IEEE Transactions on Neural Networks*, 5(1):96–101.

Schmitt, L. M. (2001). Theory of genetic algorithms. *Theoretical Computer Science*, 259:1–61.

Schmitt, L. M. (2004). Theory of genetic algorithms II: Models for genetic operators over the string-tensor representation of populations and convergence to global optima for arbitrary fitness function under scaling. *Theoretical Computer Science*, 310:181–231.

Schmitt, L. M., Nehaniv, C. L., and Fujii, R. H. (1998). Linear analysis of genetic algorithms. *Theoretical Computer Science*, 200:101–134.

van Nimwegen, E., Crutchfield, J. P., and Mitchell, M. (1997). Finite populations induce metastability in evolutionary search. *Physics Letters A*, 229:144–150.

van Nimwegen, E., Crutchfield, J. P., and Mitchell, M. (1999). Statistical dynamics of the royal road genetic algorithms. *Theoretical Computer Science*, 229:41–102.

Vose, M. D. (1993). Modeling simple genetic algorithms. In *Foundations of Genetic Algorithms 2* (pp. 63–73). San Mateo, CA: Morgan Kaufmann.

Vose, M. D. (1996). Modeling simple genetic algorithms. *Evolutionary Computation*, 3(4):453–472.

Vose, M. D. (1999a). Random heuristic search. *Theoretical Computer Science*, 229:103–142.

Vose, M. D. (1999b). *The simple genetic algorithm: Foundations and theory*. Cambridge, MA: MIT Press.

Vose, M. D., and Liepins, G. E. (1991). Punctuated equilibria in genetic search. *Complex Systems*, 5(1):31–44.

Vose, M. D., and Wright, A. H. (1994). Simple genetic algorithms with linear fitness. *Evolutionary Computation*, 2(4):347–368.

Wegener, I. (2000). On the expected runtime and the success probability of evolutionary algorithms. In *Graph-theoretic concepts in computer science*. *Lecture notes in computer science*, Vol. 1928 (pp. 1–10). Berlin: Springer-Verlag.

Wegener, I. (2001). Theoretical aspects of evolutionary algorithms. In *The Twenty-eighth International Colloquium on Automata, Languages and Programming, ICALP2001*. *Lecture notes in computer science*, Vol. 2076 (pp. 64–78). Berlin: Springer-Verlag.

Wegener, I., and Witt, C. (2005). On the analysis of a simple evolutionary algorithm on quadratic pseudo-Boolean functions. *Journal of Discrete Algorithms*, 3:61–78.

Whitley, L. D., and Vose, M. D. (1995). Introduction. In *Foundations of Genetic Algorithms 3* (pp. 1–4). San Mateo, CA: Morgan Kaufmann.