

# A New Approach to Dynamics Analysis of Genetic Algorithms without Selection

**Tatsuya Okabe**

Honda R&D Co., Ltd.,  
Wako Research Center.  
1-4-1 Chuo, Wako-shi, Saitama,  
351-0193, Japan  
tatsuya\_okabe@n.w.rd.honda.co.jp

**Yaochu Jin**

Honda Research Institute Europe  
Carl-Legien Strasse 30  
63073 Offenbach am Main  
Germany  
yaochu.jin@honda-ri.de

**Bernhard Sendhoff**

Honda Research Institute Europe  
Carl-Legien Strasse 30  
63073 Offenbach am Main  
Germany  
bernhard.sendhoff@honda-ri.de

**Abstract-** Theoretical analysis of the dynamics of evolutionary algorithms is believed to be very important to understand the search behavior of evolutionary algorithms and to develop more efficient algorithms. We investigate the dynamics of a canonical genetic algorithm with one-point crossover and mutation theoretically. To this end, a new theoretical framework has been suggested in which the probability of each chromosome in the offspring population can be calculated from the probability distribution of the parent population after crossover and mutation. Empirical studies are conducted to verify the theoretical analysis. The finite population effect is also discussed. Compared to existing approaches to dynamics analysis, our theoretical framework is able to provide richer information on population dynamics and is computationally more efficient.

## 1 Introduction

Theoretical analysis of evolutionary algorithms has received increasing attention in the recent years [Ree03]. A few examples of interesting topics are, among many others, convergence analysis [Rud94, Bae94], dynamics of evolution strategies [Bey99], genetic algorithms [Vos99a, Pru01b], and analysis of average computational time [He02].

However, the dynamics of EAs during optimization and the roles of each genetic operator are still unclear. In our opinion, the analysis of dynamics of EAs is very helpful not only to understand working mechanism of EAs [Oka02] but also to improve performance of EAs and to propose new algorithms [Oka03] because the solution of an optimizer is the result of the dynamics of EAs.

In this paper, we investigate the dynamics of crossover and mutation of genetic algorithms (GAs), both theoretically and empirically. This paper will not discuss the dynamics of selection due to the restriction of pages. The selection will be discussed elsewhere [Oka05]. For this purpose, we propose a new theoretical framework, which is particularly suited for analyzing the dynamics of GAs compared to the existing ones [Gol89, Vos99a].

Section 2 introduces in greater detail the related work on dynamics analysis of GAs. The new framework for analyzing dynamics of genetic algorithms is described in Section 3. The dynamics of GAs is investigated theoretically and empirically in Section 4. More discussions of the proposed framework are presented in Section 5. A summary of the paper is given in Section 6.

## 2 Related Work

### 2.1 Analysis of Dynamics with Cumulants

Several approaches to the dynamics analysis of single objective evolutionary algorithms have been reported, e.g. [Pru97, Rat95]. For single objective EAs, the fitness space is one-dimensional, and therefore the cumulants, e.g. average, deviation, skewness, kurtosis etc., can be used to describe the main dynamics of EAs. Prügel-Bennett and Shapiro [Pru97] have shown how to derive a set of equations describing the dynamics of a GA. They have shown the influence of genetic operators on the first four cumulants. Prügel-Bennett has also studied selection and ranking [Pru00], two point crossover [Pru01a] and recombination [Pru01b] with cumulants. Rogers and Prügel-Bennett [Rog97] have studied the roulette wheel and stochastic universal sampling, where the finite population effect has been fully considered. Generational selection and the steady state selection are analyzed in [Rog99]. Based on their analysis, they suggested that mutation tends to increase the variance of the final population equilibrium distribution but also move the mean of the distribution away from the global minimum back toward the maximum entropy state. Ratray [Rat95] has also investigated the GA dynamics with cumulants. He concluded that higher cumulants improve convergence as they increase the accumulation of correlations under selection. The role of crossover seems to be to distribute the correlations more evenly in order to increase diversity, reducing the magnitude of the higher cumulants. Van Nimwegen et al. [Nim97] have analyzed the dynamics of the Royal Road Genetic Algorithm with cumulants, where GA dynamics is considered as a flow in the fitness space.

### 2.2 Analysis of Dynamics by Modeling GAs

Although the cumulants are more tractable than the population distribution itself, much information will be lost. One of the first models of GAs was introduced in [Gol89]. Goldberg built the model for a canonical GA with two-bit strings for solving the minimum deceptive problem, where proportional selection is used. Vose [Vos93] extended Goldberg's model to an arbitrary number of strings, which is often termed *Vose's Model*. To store the information of the population distribution, he used a probability vector of which each component indicates the probability of a certain chromosome. The usage of the probability vector implicitly as-

sumes an infinite population size in its definition [Vos93].

Suzuki [Suz98] explained how to model the GA with Markov chain. Using Markov chain, Fogel[Fog92] and Rudolph [Rud94] have analyzed the convergence of canonical GA. Since an infinite population size is not realistic, the effect of finite population size in population dynamics has been studied in [Pru97, Pru00]. A more detailed review of research work on this topic can be found in [Aga99, Whi93, Whi95a].

One main drawback of Vose's Model is its huge time complexity, i.e.  $o(2^{3l})$ , where  $l$  is the length of strings. Vose and Wright [Vos98a, Vos98b] have employed the Walsh transformation to reduce the complexity. With the help of the Walsh transformation, the mixing matrix becomes a triangular matrix. In the triangular matrix, only  $3^l$  components are non-zero. In this way, the complexity can be reduced to  $o(3^l)$ . In [Vos98b], they derived Geiringer's theorem (also known as Robbins' proportions) [Wri02] using a GA model. Wright et al. [Wri02] have analyzed a gene pool GA with the Walsh transformation.

### 3 A New Theoretical Framework

For any chromosome  $Z$  of length  $l$ ,  $Z \in \{0, 1\}^l$ . A chromosome index and the occurrence probability are denoted by  $\zeta$  and  $P(\zeta)$ , respectively. Notice that in a population, we have  $\sum_{\zeta} P(\zeta) = 1$ . In this paper, a chromosome index  $\zeta$  and a chromosome  $Z$  will be used synonymously. For a function  $f(\cdot)$ , we also use both representation of  $f(\zeta)$  and  $f(Z)$  to show the value of the function  $f$  at  $Z$ . To facilitate our analysis, we first define the following notations:

**Definition 1 (Don't Care Symbol):** If the allele of bit, i.e., 1 or 0, has no influence or is not considered, this allele is notated with the symbol '\*', which is known as *Don't Care Symbol*.

**Definition 2:** If the allele is fixed, i.e., 1 or 0, we notate this allele with the symbol '#'. This means that the actual value of the allele does not matter, but it has to be fixed.

**Definition 3 (Chromosome Index  $\zeta$ ):** The index of chromosome  $Z$  is defined by  $\zeta = \sum_{i=1}^l B(Z, i) \times 2^{l-i}$ . Here,  $B(Z, i)$  means the  $i$ -th component of the chromosome  $Z$ . This equation is the same function which decodes  $Z$  to the integer value in the range  $[0, 2^l - 1]$  using binary coding.

#### 3.1 One-point Crossover

We first analyze the change of population distribution resulting from one-point crossover theoretically. Assume that the probability of applying crossover is  $P_c$ , which is also called crossover rate.

As we know, the one-point crossover in GAs is carried out by exchanging part of their chromosome. To facilitate the theoretical analysis, we can consider that the crossover is implemented in two steps, i.e., offspring 1 is generated from parent 1 assisted by parent 2, and similarly, offspring 2 is generated from parent 2 assisted by parent 1, refer to Fig. 1. Since no selection is considered in this work, we assume that the assisting parent is chosen from the population randomly.

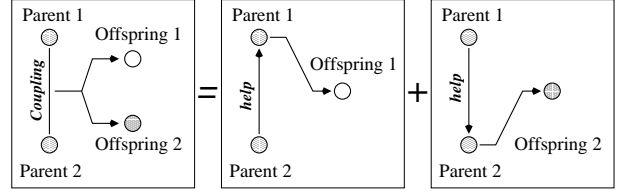


Figure 1: Illustration of a model for analyzing crossover dynamics.

Let us now consider the probability of a particular type of building blocks in which the allele of the final  $i$  bits is \*. For simplicity, a chromosome whose alleles of its last  $i$  bits are \*'s is notated as  $\zeta_i^b$  ( $b$  stands for *backwards*). For example,  $\zeta_1^b$  denotes the chromosome  $Z = \{\#\#\dots\#\#\}$ ,  $\zeta_2^b$  for chromosome  $Z = \{\#\#\dots\#\#\ast\}$  as  $\zeta_2^b$ , and so on. Thus, the chromosome index of  $\zeta_i^b$  can be defined as  $\zeta_i^b = \sum_{j=1}^{l-i} B(Z, j) \times 2^{l-j-i}$ . The probability of  $\zeta_i^b = j$ , denoted by  $P_i^b(j)$ , can be calculated from the probabilities of  $\zeta_{i-1}^b$  as follows:

$$P_i^b(j) = \begin{cases} P(2j) + P(2j+1) & \text{if } i = 1 \\ P_{i-1}^b(2j) + P_{i-1}^b(2j+1) & \text{if } i \neq 1. \end{cases} \quad (1)$$

To help understand Equation (1), an example with  $l = 2$  is given in Table 1. As Table 1 shows, the probability of  $P_i^b(\zeta_i^b)$  can be calculated from the probability of two individuals  $P_{i-1}^b(\zeta_{i-1}^b)$  having one bit fewer of *Don't Care Symbol*.

Table 1: Example of Equation (1). Here, 2 bits are assumed. The abbreviations are as follows: Chrom. = Chromosome, Prob. = Probability and B.B. = Building block.

$\zeta$	Chrom.	Prob.	$\zeta_1^b$	B.B.	Prob.
0	[00]	$P(0)$	0	[0*]	$P_1^b(0)$
1	[01]	$P(1)$		$= P(0) + P(1)$	
2	[10]	$P(2)$	1	[1*]	$P_1^b(1)$
3	[11]	$P(3)$		$= P(2) + P(3)$	

Next, we consider chromosomes whose first  $i$ -th bits are \*'s. Similarly, a chromosome index for  $Z = \{\ast\ast\dots\ast\ast\#\#\dots\#\#\}$  (the first  $i$  bits are the *Don't Care symbol*) is defined as  $\zeta_i^f = \sum_{j=1+i}^l B(Z, j) \times 2^{l-j}$  ( $f$  stands for *forwards*).

The probability of  $\zeta_i^f = j$ , denoted by  $P_i^f(j)$ , can be easily calculated by the probability of  $\zeta_{i-1}^f$  as follows:

$$P_i^f(j) = \begin{cases} P(j) + P(j+2^{l-i}) & \text{if } i = 1 \\ P_{i-1}^f(j) + P_{i-1}^f(j+2^{l-i}) & \text{if } i \neq 1. \end{cases} \quad (2)$$

Again, an example with  $l = 2$  is given in Table 2 to illustrate that the probability of  $P_i^f(\zeta_i^f)$  can be calculated by using the probabilities of  $P_{i-1}^f(\zeta_{i-1}^f)$ .

With Equation (1) and (2), the generative probability of  $Z$ , that is, the probability of  $Z$  after crossover (denoted by  $\xi(Z)$ ) can be calculated as follows:

$$\xi(Z) = P(Z) \times (1.0 - P_c) + \sum_{k=1}^{l-1} P_{l-k}^b(\zeta_{l-k}^b) \times P_k^f(\zeta_k^f) \times P_c \times \frac{1}{l-1}, \quad (3)$$

Table 2: Example of Equation (2). Here, 2 bits are assumed. The abbreviations are the same as in Table 1.

$\zeta$	Chrom.	Prob.	$\zeta_1^f$	B.B.	Prob.
0	[00]	$P(0)$	0	[*0]	$P_1^f(0)$
1	[01]	$P(1)$			$= P(0) + P(2)$
2	[10]	$P(2)$	1	[*1]	$P_1^f(1)$
3	[11]	$P(3)$			$= P(1) + P(3)$

where,  $P_c$  is the crossover rate. There are two cases in which a certain chromosome  $Z$  will be generated. The first case is that a parent with chromosome  $Z$  is chosen and no crossover occurs, which corresponds to the first term in Equation (3). The second case is that two parents are chosen and crossover occurs, corresponding to the second term in Equation (3). With Equation (3), the generative probability of all possible chromosome resulting from one-crossover can be calculated.

Now, the computational complexity to calculate all  $\xi(Z)$  is considered. It is noticed that  $3 \cdot 2^l - 6$  arithmetic operations are needed for calculating all  $P_i^b(\cdot)$ ,  $4 \cdot 2^l - 8$  for calculating  $P_i^f(\cdot)$ , 1 for calculating  $1.0 - P_c$ , 2 for calculating  $P_c/(l-1)$  and  $l \cdot (7l-3) \cdot 2^{l-1}$  for calculating  $\xi(Z)$  because  $3l+k+2$  operations are necessary for calculating  $P_{l-k}^b(\zeta_{l-k}^b) \times P_k^f(\zeta_k^f) \times P_c/(l-1)$ . Thus, the total computational complexity is  $2^{l-1} \cdot (7l^2 - 3l + 14) - 11$ . Since the part of  $2^{l-1}$  has more influence than other parts when  $l$  is large, the dominant complexity can be said to be  $o(2^l)$ .

### 3.2 Bit-flipping Mutation

If the Hamming distance [Gol89] between two chromosome  $Z$  and  $Z'$  is given by  $|Z \oplus Z'|$ , the probability that the chromosome  $Z'$  becomes  $Z$ ,  $\Xi(Z' \rightarrow Z)$ , can be calculated as:

$$\Xi(Z' \rightarrow Z) = (1 - P_m)^{l-|Z \oplus Z'|} P_m^{|Z \oplus Z'|}, \quad (4)$$

where,  $l$  and  $P_m$  are the length of chromosome and the mutation rate, respectively. The operator  $\oplus$  is bitwise exclusive-or. Since the Hamming distance indicates the number of different alleles between  $Z$  and  $Z'$ , the mutation should occur at  $|Z \oplus Z'|$  alleles and no mutation should occur at  $(l - |Z \oplus Z'|)$  alleles. Since all chromosomes have a chance to become  $Z$ , the total generative probability of  $Z$ ,  $\Xi(Z)$ , can be calculated by summarizing over all possibilities:

$$\Xi(Z) = \sum_{Z'} (1 - P_m)^{l-|Z \oplus Z'|} P_m^{|Z \oplus Z'|} \xi(Z'). \quad (5)$$

For all  $\Xi(Z)$ , denoted by  $\vec{\Xi}$ , the above equation can be rewritten in the following matrix form:

$$\vec{\Xi} = \mathcal{M} \vec{\xi}, \quad (6)$$

here,  $\vec{\Xi} = \{\Xi(0), \Xi(1), \Xi(2), \dots, \Xi(2^l - 1)\}^T$  and  $\vec{\xi} = \{\xi(0), \xi(1), \xi(2), \dots, \xi(2^l - 1)\}^T$ . The matrix  $\mathcal{M}$  is called *Mutation Matrix*. As an example, the mutation matrix for 2 bit chromosome ( $2^2 \times 2^2$ ), is calculated as:

$$\mathcal{M} = \begin{bmatrix} a & c & c & b \\ c & a & b & c \\ c & b & a & c \\ b & c & c & a \end{bmatrix},$$

where,  $a = (1 - P_m)^2$ ,  $b = P_m(1 - P_m)$  and  $c = P_m^2$ .

The dominant complexity of Equation (6) can be easily calculated to be  $o(2^{2l})$ . However, this complexity can be reduced by the Walsh transformation [Vos98a, Vos98b]. Equation (6) can be written as:

$$\widehat{\Xi} = \widehat{\mathcal{M}} \widehat{\xi}, \quad (7)$$

where,  $\widehat{\Xi} = \mathcal{W} \vec{\Xi}$ ,  $\widehat{\mathcal{M}} = \mathcal{W} \mathcal{M} \mathcal{W}$  and  $\widehat{\xi} = \mathcal{W} \vec{\xi}$ . The matrix  $\mathcal{W}$  is the Walsh matrix defined in [Vos98a, Vos98b]. With the Walsh transformation, the mutation matrix becomes a diagonal matrix where only  $2^l$  components are non-zero.

**Theorem (Walsh Transformed Mutation Matrix):** The mutation matrix can be simplified to the diagonal matrix using the Walsh transformation.

A proof of the above theorem is given in Appendix A.

The Walsh transformed mutation matrix can be calculated as:

$$\begin{aligned} \widehat{\mathcal{M}}_0 &= 1 \\ \widehat{\mathcal{M}}_n &= \begin{bmatrix} \widehat{\mathcal{M}}_{n-1} & 0 \\ 0 & (1 - 2P_m) \widehat{\mathcal{M}}_{n-1} \end{bmatrix}. \end{aligned} \quad (8)$$

With this recursive equation, the complexity to obtain Walsh transformed mutation matrix,  $\widehat{\mathcal{M}}_n$ , is only  $o(2^l)$ .

The Walsh transformed vector  $\widehat{\xi}$  can be calculated in  $o(2^l)$  by the fast Walsh transformation [Vos99a]. Then, the Walsh transformed vector  $\widehat{\Xi}$  can be calculated from  $\widehat{\mathcal{M}}$  and  $\widehat{\xi}$  in  $o(2^l(l+1))$ . By conducting the fast inverse Walsh transformation toward  $\widehat{\Xi}$ , the vector  $\Xi$  can be calculated in  $o(2^l)$ . Thus, the dominant complexity of the proposed theory in this work is still  $o(2^l)$ .

## 4 Dynamics of Crossover and Mutation

### 4.1 Theoretical Results on Crossover Dynamics

With Equation (3), we can calculate the probability of offspring after crossover without mutation ( $P_m = 0$ ),  $\xi(Z)$  in GA, given a parent population with a certain probability distribution. No selection is taken into account, i.e., the parents are selected randomly and all offspring become the next parents. The new probability of parent  $Z$  will be  $P(Z) = \xi(Z)$ . We can repeat this procedure to obtain the transition of the probability. This transition of existent probability for any possible chromosome is termed *population dynamics*, or dynamics for short in this paper. Consider an initial population whose individuals are composed of a chromosome of length 3. An example probability of each chromosome of the initial population is given in Table 3. Then, we can calculate the transition of the probability resulted from one-point crossover. The results for two crossover rates ( $P_c = 1.0$  and  $P_c = 0.5$ ) are shown in Figure 2(a) and (b), respectively.

Table 3: Example of the probability with 3 bits.

P(0) = 0.150	P(1) = 0.100	P(2) = 0.125	P(3) = 0.200
P(4) = 0.175	P(5) = 0.175	P(6) = 0.025	P(7) = 0.050

Figure 2 indicates that the probability of all chromosome converges to a certain value, which is independent of the crossover rate. In addition, if a larger crossover rate is

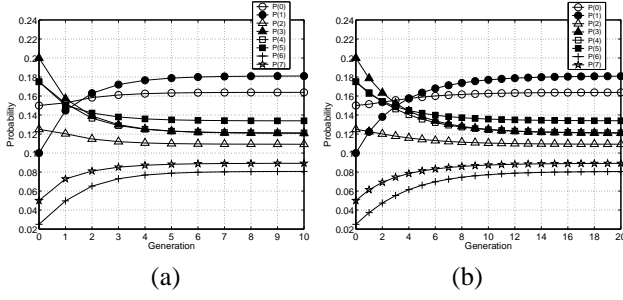


Figure 2: Theoretical results of crossover dynamics: (a)  $P_c = 1.0$ , (b)  $P_c = 0.5$ .

adopted, the convergence speed is faster. The derivative of the curves in Figure 2 is larger if the difference between initial probability and the final probability is large.

One question that arises is how to calculate the final converged probability of each chromosome. We find that the converged probability can be calculated from the product of the initial probability at which a certain allele appears at all loci. In the above example, the probability of an allele “0” appearing in the first locus is 0.575 (accordingly, the probability of an allele “1” is 0.425), the probability of an allele “0” in the second locus is 0.600, and the probability of an allele “0” in the third locus is 0.0475. Thus, the converged probability of chromosome [000] equals  $0.575 \times 0.600 \times 0.475 = 0.163875$ , refer also to Figure 2.

The correctness of above result obtained from our framework can be confirmed by *Geiringer’s Theorem II* [Boo93, Vos98b].

**Geiringer’s Theorem II [Boo93]** : If  $l$  loci are arbitrarily linked, with the one exception of “complete linkage”, the distribution of transmitted alleles “converges toward independence”. The limit distribution is given by

$$\lim_{t \rightarrow \infty} P_t(Z) = \prod_{i=1}^l P_0^i(Z), \quad (9)$$

which is the product of the  $l$  marginal distributions of alleles from the initial population.

A population in this state is said to be in *Linkage Equilibrium* or *Robbin’s Equilibrium* [Boo93].

## 4.2 Empirical Results on Crossover Dynamics

To verify the theoretical results on the crossover dynamics achieved in our framework, empirical calculations have been conducted. Figure 3 shows a generic procedure for empirical verifications. Of course, mutation and selection are skipped at this stage.

Note that the genetic algorithm is executed  $r_{max}$  times to reduce the randomness. In each generation, the number of individuals with a certain chromosome is calculated. Finally, the probability of a certain chromosome is calculated.

The empirical results are shown in Figure 4. The parameters used in the calculations are as follows: number of runs  $r_{max} = 100000$ ; crossover rate  $P_c = 1.0, 0.5$ ; the number of individuals  $|P| = 10$ , and 1000; the initial probabilities are given in Table 3. The dotted lines in Figure 4 denote the

```

begin
  counter(i, t) := 0, ∀i ∈ PI and ∀t ∈ [0, t_max];
  for r := 1 to r_max do
    initialize population P based on given probabilities;
    ∀i ∈ PI do counter(i, 0) += (number of individuals with i);
    for t := 1 to t_max do
      Crossover, P → P_cros;
      Mutation, P_cros → P_muta;
      Selection, P_muta → P_sele;
      Copy, P = P_sele;
      counter(i, t) += (number of individuals with i in P), ∀i ∈ PI;
    endfor;
  endfor;
  probability(i, t) := counter(i, t) / r_max / |P|;
end

```

Figure 3: A generic procedure for empirical verifications, where  $PI$  is the possible individual,  $t_{max}$  is the maximum number of generations,  $r_{max}$  is the number of runs,  $P_{cros}$  is the population after crossover,  $P_{muta}$  is the population after mutation,  $P_{sele}$  is population after selection, and finally,  $|P|$  is the number of individuals in population  $P$ .

theoretical results shown in Figure 2. A good agreement between the theoretical and empirical results can be observed when the population size is sufficiently large. However, a discrepancy between the theoretical and empirical results can be observed when the population size is small, which is known as *finite population effect*. This finite population effect will be discussed further in Section 5.

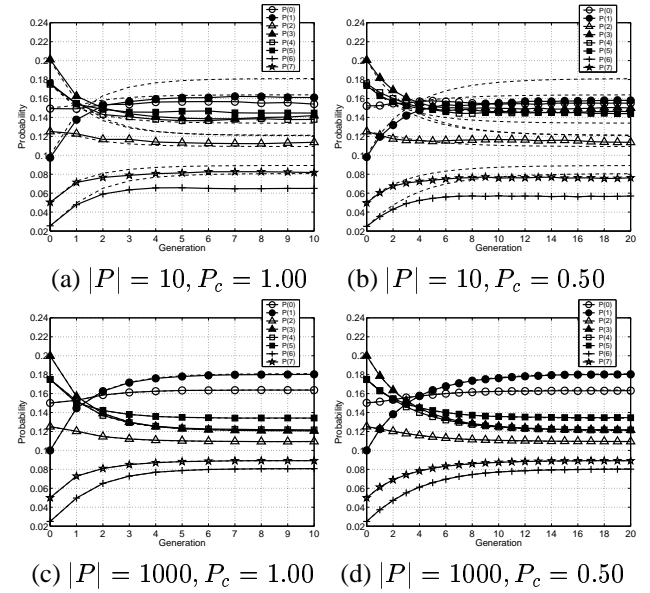


Figure 4: Empirical results on crossover dynamics. Two crossover rates and two population sizes are considered.

## 4.3 Theoretical Results on Mutation Dynamics

To investigate the dynamics of mutation, we can observe the transition of the probabilities resulting from mutation. Again the probability distribution in Table 3 has been used for the initial population and the results are shown in Figure 5, for two mutation rates  $P_m = 0.10$  and  $P_m = 0.01$ .

The dynamics resulting from mutation is clearly different from that from crossover mutation. It can be seen from

Figure 5 (a) and (b) that the probability for all chromosome converges to the same value of  $1/2^l$ . The only difference between Figure 5(a) and (b) is the speed of convergence. The larger the mutation rate, the faster the convergence speed. From Figure 5, one can also see that mutation increases the entropy of the population. It is also interesting to note that a different approach used in [Rog01] for analyzing mutation dynamics yielded the same result. In Appendix B, we show that the result obtained using our method and that in [Rog01] are equivalent using the Perron-Frobenius Theorem.

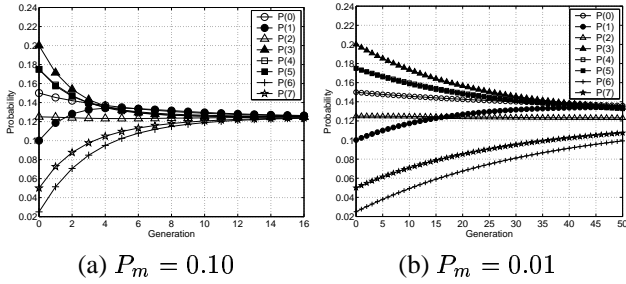


Figure 5: Theoretical results on mutation dynamics. (a)  $P_m = 0.10$  and (b)  $P_m = 0.01$ .

#### 4.4 Empirical Results on Mutation Dynamics

To verify the theoretical result on mutation dynamics, we investigate empirically the population dynamics resulting from mutation using the generic procedure in Figure 3, where crossover and selection are not considered. The population size is 10 and two different mutation rates,  $P_m = 0.10$  and  $P_m = 0.01$  are used.

The results are shown in Figure 6, which are completely the same as those obtained from theoretical analysis shown in Figure 5. Unlike the crossover dynamics, the finite population effect has no influence on the mutation dynamics since mutation is carried out independent of any other individuals.

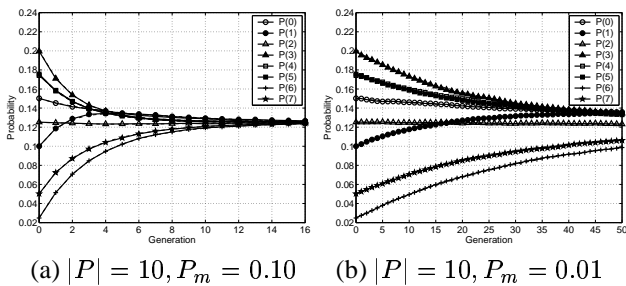


Figure 6: Empirical results on mutation dynamics. (a)  $P_m = 0.10$  and (b)  $P_m = 0.01$ . 10 individuals are used. The dotted curves are used to denote the theoretical results. Since the theoretical and empirical results are the same, we cannot see the dotted curves.

#### 4.5 Theoretical Results on Combined Dynamics of Crossover and Mutation

Now, we investigate the population dynamics when both crossover and mutation are applied. Starting from the ini-

tial probabilities given in Table 3, the transition of probabilities is calculated theoretically. The results with various crossover and mutation rates are shown in Figure 7.

Figure 7 shows that the dynamics is similar to the crossover dynamics in the early generations. However, after 2 to 10 generations, the dynamics is governed by the mutation dynamics. Comparing Figures 7(a) and (c), we find that the influence of the crossover rate is minor when the mutation rate is large. Even when the mutation rate is small (e.g., 0.01), the influence of different crossover rates diminishes in later generations.

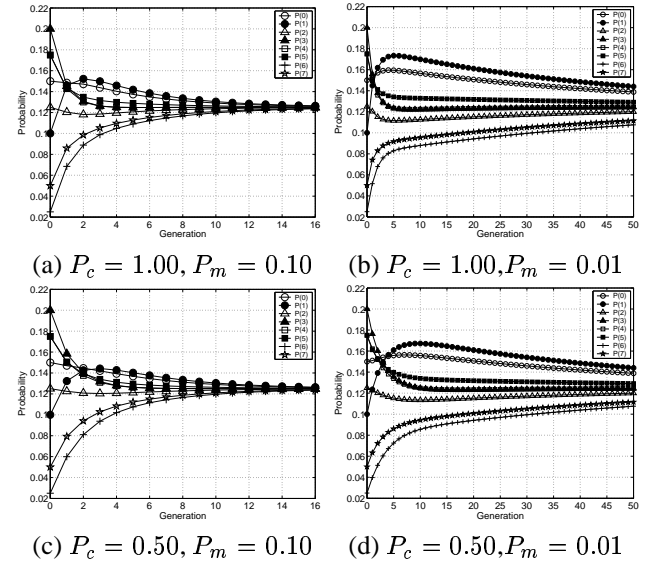


Figure 7: The theoretical results on dynamics of crossover and mutation.

#### 4.6 Empirical Results on Combined Dynamics of Crossover and Mutation

The dynamics of crossover and mutation is investigated empirically in this section. The population size is 10, the crossover rates are  $P_c = 0.50, 1.0$ , and the mutation rates are  $P_m = 0.01, 0.10$ . The results are shown in Figure 8.

Figure 8 shows that the empirical results agree well with the theoretical results. Additionally, Figure 8 shows that the finite population effect observed on crossover becomes much less significant when the mutation rate is high, e.g., in Figure 8(a) and (c) where  $P_m = 0.10$ , due to the fact that the population size does not play any role in mutation dynamics. This indicates that the population dynamics can be predicted correctly using our theoretical framework when the mutation rate is high. If the mutation rate is low, the finite population effect becomes noticeable.

## 5 Discussions

### 5.1 Difference to Vose's Theory

It is interesting to discuss the advantages and disadvantages of Vose's Model [Vos99a] and the framework proposed in this work. In our opinion, Vose's Model is particularly well suited for convergence analysis of genetic algorithms. In

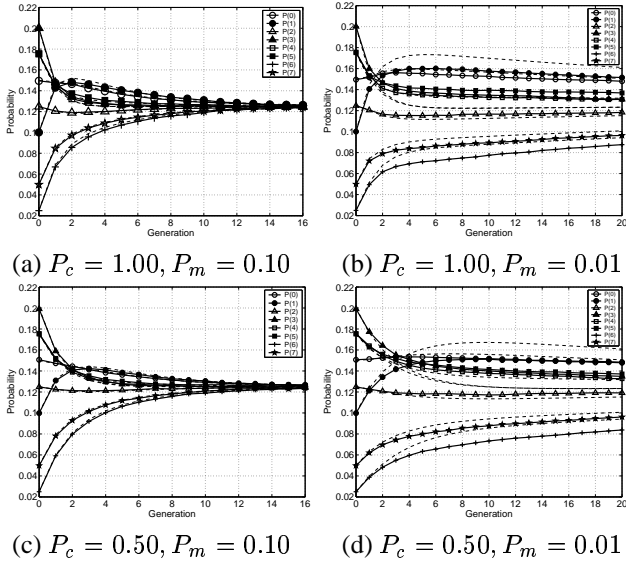


Figure 8: Empirical results on the combined dynamics of crossover and mutation. 10 individuals are used. The dotted curves show the theoretical results. (a)  $(P_c, P_m) = (1.00, 0.10)$ , (b)  $(P_c, P_m) = (1.00, 0.01)$ , (c)  $(P_c, P_m) = (0.50, 0.10)$  and (d)  $(P_c, P_m) = (0.50, 0.01)$ .

contrast, our framework is particularly effective to investigate the transient dynamics of genetic algorithms, however not very efficient for convergence analysis. The reason is as follows. In Vose's model, the computational complexity for calculating the mixing matrix is very expensive ( $o(3^l)$ ,  $l$  is the length of chromosome). Of course, this matrix needs to be calculated only once. In our theory, the computational complexity to calculate the dynamics of crossover and mutation is  $o(2^l)$ , though a re-calculation is needed for each generation. Another drawback of our framework is that it works only for one-point crossover.

## 5.2 Finite Population Effect

The discrepancy between the theoretical and empirical results when the population is finite is known as the *Finite Population Effect*. Let us denote a certain chromosome, the initial probability, the theoretically converged probability, and the converged probability obtained empirically from a finite population as  $P_{initial}$ ,  $P_{theory}$ , and  $P_{empirical}$ , respectively. Now, we define a ratio  $\alpha$  as follows:

$$\alpha = \frac{P_{empirical} - P_{initial}}{P_{theory} - P_{initial}}. \quad (10)$$

It is worth noting that  $\alpha = 1$  means an empirically converged probability is equal to a theoretically converged probability. Note that  $\alpha$  cannot be defined if a theoretically converged probability is equal to an initial probability.

We study the change of ratio  $\alpha$  with respect to the population size. We also use the probabilities in Table 3 as the initial probabilities, we set  $P_c = 1.00$  for crossover. We take the probabilities at generation 50 as the empirically converged probability. The results are shown in Figure 9 (a) when the population size is varied between 2 and 1000. Since we use 3 bits in this study,  $2^3 = 8$  possible chromo-

some exist, whose probabilities are shown in the figure. It can be seen from Figure 9 (a) that all 8 curves show the same tendency, although a few oscillations can be observed. More precisely,  $\alpha$  increases as the population size increases, and converges to 1 when the population size is larger than 120. This means that if the population size is larger than 120, the difference between an empirical result and the theoretical result is minor.

A further question that could arise is whether the convergence of  $\alpha$  depends on the initial probabilities or on the length of the chromosome. To answer this question partially, we observe  $\alpha$  using different initial probabilities and different chromosome lengths. Nine cases are investigated, i.e., 3 cases for 2 bits, 2 cases for 3 bits, 4 bits and 5 bits. Due to space limit, the different initial probabilities used will not be given here. The results are shown in Figure 9 (b). In the figure, the values of each  $\alpha$  are averaged over different chromosomes. We show only one curve for each case.

Figure 9 (b) indicates that the curves are nearly the same even if we change initial probabilities and the number of bits. However, we still have to investigate the value of  $\alpha$  for more than 5 bits to draw a more general and concrete conclusion.

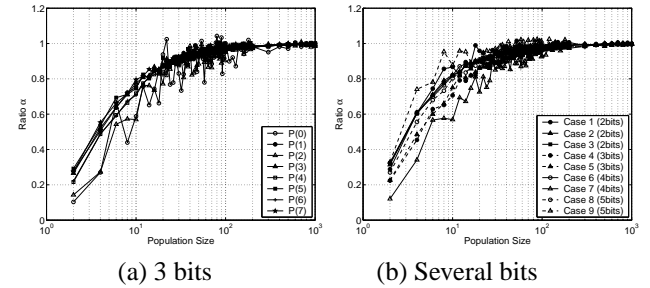


Figure 9: Change of the ratio  $\alpha$  with the number of individuals.  $\alpha = 1.00$  indicates that an empirically converged probability equals a theoretically converged probability. Here,  $P_c = 1.00$ .

## 5.3 Influence of the Number of Bits

The benefit of our theory proposed here is its low complexity<sup>1</sup>. This reduction of the complexity enables us to use more bits than the existing theory could do. To exploit this benefit, the influence of the number of bits is investigated. As an example, crossover is investigated here. The number of bits used here are 2, 4, 6, 8, 10, 12 and 14 bits. The crossover rate,  $P_c$ , is assumed to be 1.0. Due to the page limit, the initial probabilities are not given here. The representative history of each bit is shown in Figure 10. To compare all results quantitatively, the normalized probability,  $P_{norm}$  is used as:  $P_{norm} = (P - P_{init}) / (P_{link} - P_{init})$ . Here,  $P$ ,  $P_{init}$  and  $P_{link}$  are the obtained probability, the initial probability and the probability under linkage equilibrium, respectively. Note that in Figure 10, only the probability of the chromosome whose alleles are all zero is shown. Figure 10 shows that the number of bits changes the dynamics.

<sup>1</sup> Although the reduction of the complexity was successfully conducted, the complexity is still exponential. Further investigation should be done.

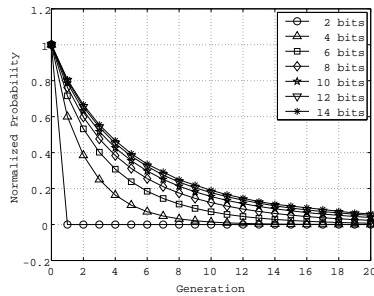


Figure 10: The influence of the number of bits.

## 6 Summary

In this paper, we have proposed a new theoretical framework for analyzing the dynamics of crossover and mutation of genetic algorithms. Compared to existing model-based approaches to dynamics analyses of GAs, this framework is computationally efficient, which makes it possible to analyze GA dynamics with a longer bit length. Besides, it enables us to examine the transient dynamics of GAs generation by generation, which might be more inspiring for designing new algorithms.

As expected, our framework confirms the main findings about the roles of crossover and mutation that were achieved by existing frameworks, though a very different approach has been adopted. For example, crossover moves a population to the so-called *Linkage Equilibrium*, which can be calculated with the initial probability distribution of the population. Besides, crossover shows the finite population effect, which can be greatly reduced with an increasing population size. In contrast, mutation moves a population to a uniform distribution, and shows no finite population effect.

Since our new framework is able to examine the transient dynamics, we also observed some more detailed behavior of crossover and mutation. For example, crossover rate has a big influence on the speed of the convergence to the linkage equilibrium. Besides, we noticed that the combined dynamics of crossover and mutation is dominated by that of crossover in the early generations. However, the dynamics of mutation begins to dominate in the later generations. If the mutation rate is sufficiently large, the effect of crossover will disappear. Contrary to that, the effect of crossover will disappear gradually when the evolution proceeds, only if the mutation rate is sufficiently small.

We studied the finite population effect more quantitatively by introducing the ratio  $\alpha$  reflecting the discrepancy between the theoretic and empirical results. With the help of this ratio, we show with our framework that the discrepancy is minor when the chromosome length is small.

## Acknowledgment

The authors would like to thank E. Körner and A. Richter for their supports and M. Olhofer for his discussion. The first author also thanks T. Arima and J. Takado.

## Bibliography

- [Aga99] Agapie, A., Adaptive Genetic Algorithms - Modeling and Convergence, *Proceedings of Congress on Evolutionary Computation (CEC-1999)*, pages 729–735, 1999.
- [Bae94] Th. Baeck. Order Statistics for Convergence Velocity Analysis of Simplified Evolutionary Algorithms. *Foundations of Genetic Algorithms*, pages 91–102, 1994.
- [Bey99] Beyer, H.-G., On the Dynamics of EAs without Selection, *Proceedings of Foundations of Genetic Algorithms 5 (FOGA-5)*, pages 5–26, 1999.
- [Boo93] Booker, L.B., Recombination Distribution for Genetic Algorithms, *Proceedings of Foundations of Genetic Algorithms 2 (FOGA-2)*, pages 29–44, 1993.
- [He02] He, J. and Yao, X., Towards an analytic framework for analysing the computation time of evolutionary algorithms. *Artificial Intelligence*, 145(1-2), pages 59–97, 2003
- [Fog92] Fogel, D. B., *Evolving Artificial Intelligence*, Ph.D. Dissertation, Univ. of California, 1992.
- [Gol89] Goldberg, D. E., *Genetic Algorithms in Search, Optimization and machine Learning*, Addison Wesley, 1989.
- [Nim97] Van Nimwegen, E., Crutchfield, J. P. and Mitchell, M., Statistical Dynamics of the Royal Road Genetic Algorithm, SFI Working Paper 97-04-035, Santa Fe Institute, 1997.
- [Oka02] Okabe, T., Jin, Y. and Sendhoff, B., On the Dynamics of Evolutionary Multi-Objective Optimisation, *Proceedings of Genetic and Evolutionary Computation Conference (GECCO-2002)*, pages 247–255, 2002.
- [Oka03] Okabe, T., Jin, Y. and Sendhoff, B., Evolutionary Multi-Objective Optimisation with a Hybrid Representation, *Proceedings of Congress on Evolutionary Computation (CEC-2003)*, pages 2262–2269, 2003.
- [Oka05] Okabe, T., Jin, Y. and Sendhoff, B., A New Theoretical Approach to Population Dynamics of Single Objective Genetic Algorithms, (In Preparation).
- [Pru97] Pruegel-Bennett, A. and Shapiro, J. L., The Dynamics of a Genetic Algorithm for Simple Random Ising Systems, *Physica D*, 104, pages 75–114, 1997.
- [Pru00] Pruegel-Bennett, A., Finite Population Effects for Ranking and Tournament Selection, *Complex Systems*, 12(2), pages 183–205, 2000.
- [Pru01a] Pruegel-Bennett, A., Modelling Crossover-Induced Linkage in Genetic Algorithms, *IEEE Transactions on Evolutionary Computation*, 5(4), pages 376–387, 2001.
- [Pru01b] Pruegel-Bennett, A., Modelling Genetic Algorithm Dynamics, *Theoretical Aspects of Evolutionary Computing*, pages 59–85, 2001.
- [Rat95] Rattray, L. M., The Dynamics of a Genetic Algorithm under Stabilizing Selection, *Complex Systems*, 9, pages 213–234, 1995.
- [Ree03] Reeves, C. R. and Rowe, J. E., *Genetic Algorithms - Principles and Perspectives: A Guide to GA Theory*, Kluwer Academic Publishers, 2003.
- [Rog97] Rogers, A. and Pruegel-Bennett, A., The Dynamics of a Genetic Algorithm on a Model Hard Optimization Problem, *Complex Systems*, 11(6), pages 437–464, 1997.
- [Rog99] Rogers, A. and Pruegel-Bennett, A., Modelling the Dynamics of a Steady State Genetic Algorithm, *Proceedings of Foundations of Genetic Algorithms 5 (FOGA-5)*, pages 57–68, 1999.
- [Rog01] Rogers, A. and Pruegel-Bennett, A., A Solvable Model of a Hard Optimisation Problem, *Theoretical Aspects of Evolutionary Computing*, pages 207–221, 2001.
- [Rud94] Rudolph, G., Convergence Analysis of Canonical Genetic Algorithms, *IEEE Transactions on Neural Networks*, 5(1), pages 96–101, 1994.

- [Suz98] Suzuki, J., A Further Result on the Markov Chain Model of Genetic Algorithms and Its Application to a Simulated Annealing-Like Strategy, *IEEE Transactions Systems, Man, Cybernetics*, pages 95–102, 1998.
- [Whi93] Whitley, D., An Executable Model of a Simple Genetic Algorithm, *Proceedings of Foundations of Genetic Algorithms 2 (FOGA-2)*, pages 45–62, 1993.
- [Whi95a] Whitley, D., A Review of Models for Simple Genetic Algorithms and Cellular Genetic Algorithms, *Applications of Modern Heuristic Methods*, pages 55–67, 1995.
- [Wri02] Wright, A. H., Rowe, J. E., Poli, R. and Stephens, C. R., A Fixed Point Analysis of a Gene Pool GA with Mutation, *Proceedings of Genetic and Evolutionary Computation Conference (GECCO-2002)*, pages 642–649, 2002.
- [Vos93] Vose, M. D., Modeling Simple Genetic Algorithms, *Proceedings of Foundations of Genetic Algorithms 2 (FOGA-2)*, pages 63–73, 1993.
- [Vos98a] Vose, M. D. and Wright, A. H., The Simple Genetic Algorithm and the Walsh Transform: Part I, Theory, *Evolutionary Computation*, 6(3), pages 253–274, 1998.
- [Vos98b] Vose, M. D. and Wright, A. H., The Simple Genetic Algorithm and the Walsh Transform: Part II, The Inverse, *Evolutionary Computation*, 6(3), pages 275–289, 1998.
- [Vos99a] Vose, M. D., *The Simple Genetic Algorithm: Foundations and Theory*, The MIT Press, 1999.

## A Walsh Transformed Mutation Matrix

**Proof:** Mathematical induction is used in the proof.

(1) The Walsh transformed mutation matrix  $\widehat{\mathcal{M}}_1$  for 1 bit is as follows:

$$\widehat{\mathcal{M}}_1 = \mathcal{W}_1 \mathcal{M}_1 \mathcal{W}_1 = \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 - 2P_m \end{pmatrix}. \quad (11)$$

Thus,  $\mathcal{M}_1$  is a diagonal matrix.

(2) We assume that the Walsh transformed mutation matrix for  $l = n - 1$ ,  $\widehat{\mathcal{M}}_{n-1}$ , is a diagonal matrix.

Now, we will take  $l = n$  into consideration. The Walsh transformation matrix  $\mathcal{W}_n$  and the mutation matrix  $\mathcal{M}_n$  can be expressed with the submatrix  $\mathcal{W}_{n-1}$  and  $\mathcal{M}_{n-1}$  as follows:

$$\mathcal{W}_n = \frac{1}{\sqrt{2}} \begin{pmatrix} \mathcal{W}_{n-1} & \mathcal{W}_{n-1} \\ \mathcal{W}_{n-1} & -\mathcal{W}_{n-1} \end{pmatrix}, \quad (12)$$

$$\mathcal{M}_n = \begin{pmatrix} (1 - P_m)\mathcal{M}_{n-1} & P_m\mathcal{M}_{n-1} \\ P_m\mathcal{M}_{n-1} & (1 - P_m)\mathcal{M}_{n-1} \end{pmatrix}. \quad (13)$$

The Walsh transformed mutation matrix  $\widehat{\mathcal{M}}_n$  can be calculated as:

$$\widehat{\mathcal{M}}_n = \mathcal{W}_{n-1} \mathcal{M}_{n-1} \mathcal{W}_{n-1} \begin{pmatrix} 1 & 0 \\ 0 & 1 - 2P_m \end{pmatrix}. \quad (14)$$

Then, the Walsh transformed mutation matrix  $\widehat{\mathcal{M}}_n$  is also a diagonal matrix.

With (1) and (2), the Walsh transformed mutation matrix becomes a diagonal matrix for any number of bits.

(Q.E.D)

## B Role of Mutation

**Proof:** The mutation matrix is assumed to be  $\mathcal{M}_n$  for  $n$ -bit problem. The number of components in  $\mathcal{M}_n$  is  $2^n \times 2^n$ . We denote the identity matrix with  $2^n \times 2^n$  as  $I_n$ .

The mutation matrix  $\mathcal{M}_n$  can be written as:

$$\mathcal{M}_0 = (1), \quad \mathcal{M}_1 = \begin{pmatrix} 1 - P_m & P_m \\ P_m & 1 - P_m \end{pmatrix}, \quad \dots \quad (15)$$

Here,  $P_m$  ( $0 < P_m \leq 1$ ) is the mutation rate. For general cases, the mutation matrix  $\mathcal{M}_n$  can be given by:

$$\mathcal{M}_n = \begin{pmatrix} (1 - P_m)\mathcal{M}_{n-1} & P_m\mathcal{M}_{n-1} \\ P_m\mathcal{M}_{n-1} & (1 - P_m)\mathcal{M}_{n-1} \end{pmatrix}. \quad (16)$$

Since  $0 < P_m \leq 1$ , all components of  $\mathcal{M}_n$  are non-negative. Thus, Perron-Frobenius theorem says that  $\lim_{t \rightarrow \infty} (\mathcal{M}_n)^t \vec{P}$  will converge  $\vec{x}$  which satisfies  $\mathcal{M}_n \vec{x} = \lambda_{\max} \vec{x}$ . Here, the sum of all components in  $\vec{P}$  and  $\vec{x}$  are 1. The variable of  $\lambda_{\max}$  is given by  $\lambda_{\max} = \max(|\lambda|)$ , where,  $\lambda$  are eigenvalues obtained from  $\mathcal{M}_n \vec{x} = \lambda \vec{x}$ . Note that the number of components of  $\vec{P}$  and  $\vec{x}$  are  $2^n$ .

To obtain  $\lambda_{\max}$ , first, we will calculate the eigenvalues of  $\mathcal{M}_n$ . Since  $\mathcal{M}_n \vec{x} = \lambda \vec{x} \iff \mathcal{W}_n \mathcal{M}_n \mathcal{W}_n \mathcal{W}_n \vec{x} = \lambda \mathcal{W}_n \vec{x} \iff \widehat{\mathcal{M}}_n \vec{y} = \lambda \vec{y}$ , we can calculate the eigenvalues by  $\widehat{\mathcal{M}}_n$ . Here,  $\mathcal{W}_n$  is the Walsh transformation matrix for  $n$  bits,  $\widehat{\mathcal{M}}_n = \mathcal{W}_n \mathcal{M}_n \mathcal{W}_n$  and  $\vec{y} = \mathcal{W}_n \vec{x}$ .

We denote  $d_n$  as  $d_n = \det(\widehat{\mathcal{M}}_n - \lambda I_n)$ . Since the Walsh transformed mutation matrix,  $\widehat{\mathcal{M}}_n$ , can be given by:

$$\widehat{\mathcal{M}}_n = \begin{pmatrix} \mathcal{M}_{n-1} & 0 \\ 0 & (1 - 2P_m)\mathcal{M}_{n-1} \end{pmatrix}, \quad (17)$$

we can calculate  $d_n$  as follows:

- (1)  $n = 0$   
 $d_0 = \det(\widehat{\mathcal{M}}_0 - \lambda I_0) = 1 - \lambda$
- (2)  $n = 1$   
 $d_1 = \det(\widehat{\mathcal{M}}_1 - \lambda I_1) = (1 - \lambda)((1 - 2P_m) - \lambda)$
- (3)  $n = 2$   
 $d_2 = \det(\widehat{\mathcal{M}}_2 - \lambda I_2)$   
 $= (1 - \lambda)((1 - 2P_m) - \lambda)^2((1 - 2P_m)^2 - \lambda)$ .

From the above examples, one can easily find out the solutions which satisfy  $d_n = \det(\widehat{\mathcal{M}}_n - \lambda I_n) = 0$ . The solutions for  $n$  bits are as follows:

$$\lambda = 1, \quad 1 - 2P_m, \quad (1 - 2P_m)^2, \quad \dots, \quad (1 - 2P_m)^n. \quad (18)$$

Since  $0 < P_m \leq 1$ , the variable of  $\lambda_{\max}$  is  $\lambda_{\max} = 1$ .

Now, the eigenvector of  $\vec{y}$  for  $\lambda = 1$  will be calculated by  $\widehat{\mathcal{M}}_n \vec{y} = \lambda \vec{y} = \vec{y}$ . This equation can be also written with each component as:

$$\begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 - 2P_m & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & (1 - 2P_m)^n \end{pmatrix} \vec{y} = \vec{y}. \quad (19)$$

Since  $0 < P_m \leq 1$ ,  $(1 - 2P_m)^n \neq 1$ . the vector  $\vec{y}$  can be written as  $\vec{y} = (y_0, 0, 0, \dots, 0)^T$ .

With  $\vec{y} = \mathcal{W}_n \vec{x}$ , the eigenvector of  $\vec{x}$  will be calculated. Since  $\mathcal{W}_n^{-1} = \mathcal{W}_n$ ,  $\vec{x}$  can be calculated as:

$$\vec{x} = \mathcal{W}_n \vec{y} = (y_0, y_0, \dots, y_0). \quad (20)$$

Since  $\vec{x}$  should satisfy  $\sum_{i=0}^{2^n-1} x_i = 1$ , one can obtain  $\vec{x}$  that  $\vec{P}$  converges as:

$$\vec{x} = (1/2^l, 1/2^l, \dots, 1/2^l)^T. \quad (21)$$

Therefore, mutation increases the entropy of the population, and finally it leads population with a uniform distribution.

(Q.E.D)