# A Network Analysis of Simple Genetic Algorithms

Hiroyuki Funaya
Department of Systems Science
Kyoto University
Kyoto 606-8501 Japan

Kazushi Ikeda
Department of Systems Science
Kyoto University
Kyoto 606-8501 Japan

**Abstract**

In recent years, network analysis has revealed that some real networks have the properties of small-world and/or scale-free networks. In this study, a simple Genetic Algorithm (GA) is regarded as a network where each node and each edge respectively represent a population and the possibility of the transition between two nodes. The characteristic path length with the crossover operation, which is one of the most popular criterion in small-world networks, shows how effective the crossover operation is, compared to that with only the mutation operation.

## 1 Introduction

There have been several theoretical results on the properties of genetic algorithms (GAs), such as the schemata theorem [1,2] and the asymptotic theory [3–5]. However, GAs are not optimizers; They find a good but not optimal solution in a short time. Such a solution is termed a quasi-optimal solution. The above results do not seem to explain why GAs are good quasi-optimizers.

This study takes another approach to this problem: We regard a GA as a network, where a node is a possible set of individuals, and investigate the connectivity of the network from a network analytical point of view. Network analysis has recently attracted much attention as a new method to analyze complex phenomena in the world, where the following two properties have been found in many real networks [6–9]: One is referred to as a small-world network, which means that a network simultaneously has dense local connections and short pairwise distances. The other is a scale-free network, which means that the distribution of the orders of nodes in a network has a long tail obeying the power law.

We shed light on the former property. That is, we analytically derive the characteristic path length (CPL) $\nu$, defined as the shortest path length (SPL) between two nodes averaged over all possible pairs. Since it is expected that a GA with a smaller CPL takes a shorter time to find a solution, we see how the two basic genetic operations in GAs, crossover and mutation, affect the CPL.

## 2 Network of Genetic Algorithms

Although many variants of GAs have been proposed since the original GA was born [1,2], we analyze the simplest case as formulated below.

Each individual consists of a binary sequence of length $L$. That is, we have $2^L$ kinds of individuals. A set of individuals defines a population. We assume that each population has only two individuals at first and consider more general cases later. Then, the cardinality of the different populations becomes

$$N \equiv 2^{L-1}(2^L - 1). \tag{1}$$

When the generation proceeds, a population changes by one of the two basic genetic operations, one-point crossover or mutation. The former randomly chooses one crossover-point from $L-1$ candidates and exchanges the bits rightward from the point, while the latter randomly chooses one of $2L$ loci (or gene-positions) in the two individuals and inverts its bit from 0 to 1 or vice versa. Note that we do not treat any fitness function because we are only considering the possibility of population-transition from one in a generation to another in the next generation.

We regard a population as a node of a GA network. Hence, the network has $N$ nodes. Two nodes are linked by an edge if and only if one of the two nodes can change to the other in a one-point crossover or mutation operation (Fig. 1).

If a network consists of only the edges from mutation, it is a lattice of the $2L$-dimensional hypercube because an individual is an $L$-bit sequence and a population consists of two individuals. Therefore, the path-length of any two distinct nodes is the same as the

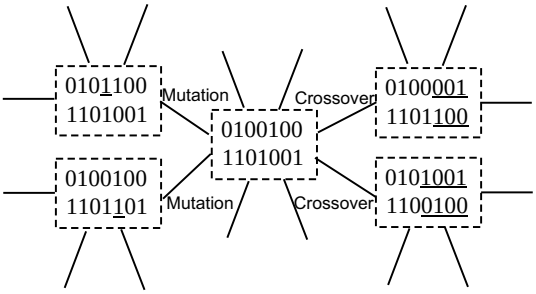


Figure 1: A part of the network of a genetic algorithm.

Manhattan distance, that is, $L$ on average. This means that we need $L$ generations to reach a quasi-optimal population from an initial one.

On the contrary, the edges from crossover are shortcuts in the network where plural bits can change at once. It is likely that these shortcuts enable GAs to find a quasi-optimal solution in a short time. The purpose of this study is to evaluate quantitatively how these shortcuts work to shorten the CPL and to clarify their effects in GAs.

## 3 Characteristic Path Length

In many cases of network analysis, the CPL is numerically calculated from the empirical data collected. However, the CPL of the GA network treated here can be derived analytically due to its simplicity, as shown below.

One of the main ideas for derivation is to classify $L$ loci into four types according to how two populations can be matched by genetic operations:

**Type 1** All four genes have the same alphabet.

**Type 2** The two genes of a population are the same but the two genes of the other population are different.

**Type 3** The two genes of each population are the same but the two populations have different alphabets.

**Type 4** Each population has two different genes. That is, the genes at the locus are 0110, 1001, 0101 or 1010. The former two are termed Type 4-1 while the others Type 4-2.

Note that the crossover operation cannot change the type of a locus and the mutation operation works bitwise. Therefore, the loci belonging to Types 1, 2 and 3 respectively contribute zero, one and two for the SPL no matter where they are located. Hence, the SPL of two nodes is the sum of the above and the SPL of the two shorter nodes consisting of only the loci of Type 4. The latter for $l$-bit individuals can be calculated using the following procedure, as shown in Fig. 2:

| | |
|---|---|
| A Gene in Population 1 $g1$ | 0 1 0 1 0 |
| A Gene in Population 2 $g2$ | 1 0 0 0 1 |
| $p$ = $g1$ XOR $g2$ | 1 1 0 1 1 |
| $p1$ = Shifted $p$ | 1 1 0 1 1 |
| $q$ = $p$ XOR $p1$ | 0 1 1 0 |

Figure 2: How to calculate the SPL of two nodes consisting of the loci of Type 4.

1. Take $l$-bit XOR bitwise between an individual in a population and one in the other population and denote it by $p$.
2. Take $(l-1)$-bit XOR bitwise between $p$ and 1-bit shifted $p$ and denote it by $q$.
3. Count the number of 1s in $q$.

The other of the main ideas for derivation is to count the number of node-pairs with the SPL $M$, instead of evaluating the SPL of each node-pair directly. Let the numbers of Type 1, 2, 3 and 4 in $L$ loci be denoted by $l_1$, $l_2$, $l_3$ and $l_4$, and the numbers of links in $M$ by $m_1$, $m_2$, $m_3$ and $m_4$, respectively. Here,

$$L = l_1 + l_2 + l_3 + l_4, \tag{2}$$

$$M = m_1 + m_2 + m_3 + m_4, \tag{3}$$

$$m_1 = 0, \quad m_2 = l_2, \quad m_3 = 2l_3 \tag{4}$$

stand by definition.

In the case of $m_4 = 0$, the number of node-pairs satisfying (3) is written as

$$\frac{L!2^{l_1}8^{l_2}2^{l_3}}{l_1!l_2!l_3!} \tag{5}$$

for fixed $l_1$, $l_2$ and $l_3$. Since they must satisfy

$$l_2 + 2l_3 = 0 \tag{6}$$

$$l_1 + l_2 + l_3 = L \tag{7}$$

$$0 \le l_3 \le \lfloor L/2 \rfloor \tag{8}$$

from (2) to (4), the ratio of the number of node-pairs to the possible pairs for $m_4 = 0$ is

$$\frac{1}{2^{4L}} \sum_{M=1}^{2L} \sum_{l_3=0}^{\lfloor L/2 \rfloor} \frac{ML!2^{L+2M-4l_3}}{(L-M+l_3)!(M-2l_3)!l_3!}, \tag{9}$$

which is denoted by $\tilde{\nu}_1$. Otherwise, for fixed $l_1$, $l_2$, $l_3$ and $l_4$, the number of combinations of positions is equal to

$$\frac{L!}{l_1!l_2!l_3!l_4!} \tag{10}$$

and the number of combinations of places where crossover occurs is ${}_{l_4-1}\mathrm{C}_{m_4}$. Taking into account the cardinality of Type 4 and the possibility that the leftmost in the Type 4 loci belongs to Type 4-1 or Type 4-2, the total number of node-pairs is written as

$$\frac{L!2^{l_1}8^{l_2}2^{l_3}2^{l_4+1}{}_{l_4-1}\mathrm{C}_{m_4}}{l_1!l_2!l_3!l_4!}. \tag{11}$$

Summing up for all possible combinations of $l_1$, $l_2$, $l_3$, $l_4$ and $m_4$ under the conditions (2) to (4) and $m_4 \leq l_4 - 1$, the ratio of the number of node-pairs to the possible pairs is written as

$$\frac{1}{2^{4L}} \sum_{M=1}^{2L} \sum_{l_1,l_2,l_3,l_4,m_4} \frac{ML!2^{l_1}8^{l_2}2^{l_3}2^{l_4+1}{}_{l_4-1}\mathrm{C}_{m_4}}{l_1!l_2!l_3!l_4!}, \tag{12}$$

which is denoted by $\tilde{\nu}_2$. In total, the complete expression for the CPL is

$$\tilde{\nu} = \tilde{\nu}_1 + \tilde{\nu}_2. \tag{13}$$

For convenience, we show $\hat{\nu}$ for some $L$'s in Table 1. Since the CPL of the network consisting of only the edges from mutation is $L$, the ratio shows how the crossover operation shortens the CPL. We see that the CPL is rather large even when the edges from crossover are added.

| $L$ | $\hat{\nu}$ | ratio |
|---|---|---|
| 3 | 2.3359 | 0.7786 |
| 8 | 6.5501 | 0.8188 |
| 13 | 10.887 | 0.8375 |
| 18 | 15.253 | 0.8474 |
| 28 | 24.000 | 0.8571 |
| 48 | 41.500 | 0.8646 |
| 68 | 59.000 | 0.8676 |

Table 1: The CPL $\hat{\nu}$ for some $L$'s.

Note that the path length is calculated in Type 4 as if the individuals in a population are ordered. However, the CPL $\tilde{\nu}$ of the network where individuals in a population are ordered is expressed as

$$\tilde{\nu} = \frac{2N\nu + L}{\tilde{N}} \tag{14}$$

where $\nu$ is the true CPL and $\tilde{N} \equiv 2^{2L}$ [10, 11]. Hence, the difference is negligible when $L$ is large.

## 4 Inductive Approach

As is expected, it is not easy to discuss more general cases such that a population consists of $K$ individuals for $K > 2$ and the exact analysis is to be done yet. Instead, we give an upper bound of the CPL for a general $K > 2$, by taking an inductive approach.

In order to make our new method clear, we first discuss the method for the case of $K = 2$. Let a pair of populations of length $l$ be expressed as a $4 \times l$ matrix denoted by $\Pi_l$, where the first and second rows correspond to one population and the third and fourth do the other. Some matrices represent a pair of populations any of whose shortest paths includes one or more crossover operations, which we term C matrices. The others are called M matrices. Let the cardinality of C matrices be denoted by $p_c(l)$ and that of M matrices by $p_m(l)$. Obviously, $p_c(l) + p_m(l) = 2^{4l}$. We denote the SPL of each of C matrices by $r_c(l, j)$, $j = 1, \ldots, p_c(l)$, and the SPL of each of M matrices by $r_m(l, i)$, $i = 1, \ldots, p_m(l)$. Then, $\tilde{\nu}$ is expressed as

$$\tilde{\nu} = \frac{q_c(l) + q_m(l)}{2^{4l}}, \tag{15}$$

where $q_c(l)$ and $q_m(l)$ are defined as

$$q_c(l) = \sum_{j=0}^{p_c(l)} r_c(l, j), \tag{16}$$

$$q_m(l) = \sum_{i=0}^{p_m(l)} r_m(l, i). \tag{17}$$

Let us consider the populations of length $l+1$. Any pair of them is expressed as one of the following 16 matrices,

$$\begin{pmatrix} 0 \\ 0 \\ \Pi_l \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ \Pi_l \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ \Pi_l \\ 1 \\ 0 \end{pmatrix}, \ldots, \begin{pmatrix} 1 \\ 1 \\ \Pi_l \\ 1 \\ 1 \end{pmatrix}. \tag{18}$$

Using this property, we can derive the update equation as

$$p_c(l+1) = 16p_c(l) + 4p_m(l), \tag{19}$$

$$p_m(l+1) = 12p_m(l), \tag{20}$$

$$q_c(l+1) = 16q_c(l) + 4q_m(l) + 14p_c(l), \tag{21}$$

$$q_m(l+1) = 12q_m(l) + 12p_m(l), \tag{22}$$

where the initial condition is

$$p_c(1) = 4, \qquad p_m(1) = 12, \tag{23}$$

$$q_c(1) = 0, \qquad q_m(1) = 12. \tag{24}$$

See [10] for detail. This result completely agrees to the result by the combinatorial method, (13).

Although (13) is exact, it is difficult to see the characteristics of this formula. A rather rough analysis below gives a simple conclusion, that is, the crossover operation reduces the CPL to its 7/8 at most. See (18) again. In four of the sixteen matrices, the added column vector is classified to Type 4. This means that the crossover operation cannot contribute to decrease the CPL in the twelve of the matrices. Moreover, a Type-4 locus belongs to either Types 4-1 or 4-2. If the SPL of one of them is the same as $\Pi_l$, the other is necessarily larger. In total, only two of the sixteen matrices have shorter SPL. This means that the crossover operation reduces the CPL to its 7/8 at most. In fact, the ratio in (13) seems to approach 7/8.

This idea still holds in the case of $K > 2$, by considering choosing two individuals from a population and the corresponding two individuals from the other population in a pair.

## 5 Conclusions

We regarded a simple GA as a network and analytically derived its exact CPL for $K = 2$ and an upper bound for a general $K$. The result shows how the crossover operation works to shorten the CPL of a network consisting of only the edges from mutation. In short, even when the crossover operation is applied, the CPL is not so small, and the same order $O(L)$ as in the case when only the mutation operation is employed.

One of the reasons is averaging: Since the CPL is the SPL averaged over any pair of populations, it is almost ignored how the information is encoded into the genes. To overcome this problem, we need to introduce the selection pressures of the GA, which were neglected here. Our future work will investigate such matters.

### Acknowledgements

This study is supported in part by a Grant-in-Aid for Scientific Research (15700130, 18300078) from the Japanese Government.

## References

[1] J. H. Holland, *Adaptation in Natural and Artificial Systems.* Univ. Michigan Press, 1975.

[2] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley Pub., 1989.

[3] J. Suzuki, "A markov chain analysis on simple genetic algorithms," *IEEE Trans. on Systems, Man and Cybernetics (TSMC)*, vol. 25, no. 4, pp. 655–659, 1995.

[4] ——, "A further result on the markov chain model of genetic algorithms and its application to a simulated annealing-like strategy," *IEEE Trans. SMC-B*, vol. 28, no. 1, pp. 95–102, 1998.

[5] H. Mühlenbein and R. Höns, "The estimation of distributions and the minimum relative entropy principle," *Evolutionary Computation*, vol. 13, no. 1, pp. 1–27, 2005.

[6] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, pp. 440–442, 1998.

[7] A.-L. Barabasi and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509–512, 1999.

[8] D. S. Callaway, M. E. J. Newmann, S. H. Strogatz, and D. J. Watts, "Network robustness and fragility: Percolation on random graphs," *Physical Review Letters*, vol. 85, pp. 5468–5471, 2000.

[9] S. H. Strogatz, "Exploring complex networks," *Nature*, vol. 410, pp. 268–276, 2001.

[10] H. Funaya, *An analysis of genetic algorithms using network science*, Bachelor's thesis, Kyoto University, 2006. In Japanese.

[11] H. Funaya and K. Ikeda, "On properties of genetic operators from a network analytical viewpoint," *Proc. ICONIP*, Part III, pp. 746–753, 2006.