Six Degrees of Separation in Boolean Function Networks with Neutrality

Tina Yu

ChevronTexaco Information Technology Company, San Ramon CA 94583, USA

Abstract. We analyze two Boolean function networks with different degrees of neutrality. The results show that the one with explicit neutrality is a small-world network where each pair of possible solutions has a short distance and most of the possible solutions are highly clustered. These network structural properties owe their existence to the "short cuts" introduced by redundant genes in the genotypes. We explain some important small-world network structures, such as clusters, hubs and power law link distribution. These properties have potential to be useful in designing efficient evolutionary algorithms to navigate search in the network.

1 Introduction

"Six Degrees of Separation" is a play created by John Guare in 1990 to illustrate the small-world phenomenon - most of us are linked by short chains of acquaintances. The name of the play, according to [2], came from a study by Stanley Milgram in 1967. Milgram was interested in the structure of our social networks and wanted to find out the "distance" between any two people in the United States. For this purpose, he recruited individuals in Nebraska and Kansas to try forwarding a letter to a designated target in Massachusetts through people they knew on a "first-name" basis. The starting individuals were given basic information about the target, such as the name, address, occupation, and a few other personal details. They had to choose one of their acquaintances to send the letter to, with the goal of reaching the target as quickly as possible; subsequent recipients followed the same procedure, and the chain closed in on its destination. Of the chains that were completed, the median number of steps required was six, hence "six degrees of separation" [5].

Many social and technological networks are small-world. For example, the World-Wide-Web network is small-world with nineteen degrees of separation: any web document is on average only nineteen clicks away from any other [2]. Hollywood film actors collaboration network is a small-world: each actor is 3 links from most actors [7]. Biological systems, such as food webs [9] and neural networks of the nematode worm *C. elegans* are also small-world [8]. All these networks are very large in size, yet have a small degree of separation. This is a surprising phenomenon to most people.

Although small-world networks always have a very short path between two vertices, there is not always an algorithm that can find this shortest path. In

Milgram's experiments, local information about the acquaintances are used to select the one that is most likely to know the target. Similarly, intermediate web-search results are frequently used to select the web-link that is most likely to lead to the desired document. In general, however, small-world networks are not always rich in local-information for efficient navigation. Kleinberg has showed that efficient navigation is a fundamental property of only some small-world networks [4].

This research attempts to answer two questions that are related to neutral evolution in Evolutionary Computation:

1. Are neutral Boolean function networks small-world ?
2. Can we utilize small-world network structural properties to design efficient evolutionary algorithms to navigate search in these networks ?

This paper addresses the first question by analyzing structural properties of two Boolean function networks with different degrees of neutrality. The design of efficient evolutionary navigation algorithms for these networks will be investigated in a later work.

We organize the paper as follows. Section 2 explains small-world networks and their associated properties. Section 3 describes neutrality in evolutionary search networks. In Section 4, we study structural properties of two Boolean function networks with different degrees of neutrality. The results are then analyzed and discussed in Section 5. Finally, Section 6 concludes the paper and outlines the direction of our future research.

2 Small-World Networks

The small-world network by Watts and Strogatz [8] has a large number of vertices with sparse connections. In particular, for a network with n vertices and k edges per vertex, it is required that $n \gg k \gg ln(n) \gg 1$. With $n \gg k$, the network is not fully connected; with $k \gg ln(n)$, the network is always connected [3]. They used two measurements to quantify the structural properties of small-world networks: characteristic path length L and clustering coefficient C.

Characteristic path length L is the average shortest distance between vertices pairs in a network.

Definition 1. *Let d(i,j) be the length of the shortest path between the vertices i and j, then the characteristic path length, L,is d(i,j) averaged over all* $\binom{n}{2}$ *pairs of vertices.*

Clustering coefficient C is the average local clustering coefficient C_v over all vertices in a network.

Definition 2. *The neighborhood of a vertex v,* $\Gamma_v = \{\, i : d(i,v) = 1 \,\}(v \notin, \Gamma_v)$

Definition 3. *The local clustering coefficient,* C_v*, is:* $C_v = \frac{|E(\Gamma_v)|}{\binom{m}{2}}$ *where* $|E(\Gamma_v)|$ *gives the total number of edges among the m neighbors.*

Definition 4. *The clustering coefficient, C, is C_v averaged over all vertices.*

Using social friendship networks as an example, these two measurements have intuitive meanings: L is the average number of friendships that connects two people in the network; C_v tells how many friends of v are also friends of each other. Thus C gives the cliquishness of a friendship circle. A large C indicates that everybody knows almost everybody else in the friendship network. If C is 1, everybody in the network knows everybody else.

The computation of L is straight forward. Equation (1) gives the mathematical formula, which can be easily implemented in any programming language. Calculating C, however, involves more steps. Using Figure 1 as an example network, we show step by step how to find C for vertex 0000 following Definition 2 and 3. The vertex 0000 has 8 neighbors. Among them, there are 4 edges. C_0000 is therefore 0.14285. Once C_v for all vertices in the network are obtained, C is the average of all C_v.

$\Gamma_{0000} = \{0001, 0002, 0010, 0020, 0100, 0200, 1000, 2000\}$.
$k_{0000} = 8$.
$|E(\Gamma_{0000})| = 4$.
$C_{0000} = \frac{4}{\binom{8}{2}} = 0.14285$.

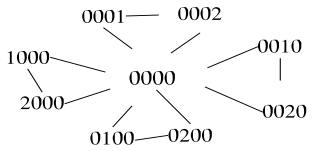

Fig. 1. A network example for calculating clustering coefficient C_v.

Watts and Strogatz used three small-size networks to study small-world network properties. In Figure 2, network (a) is a highly clustered ($C = 0.5$) large world ($L = 6.63$) where L grows linearly with the number of vertices n. In contrast, network (c) is a poorly clustered ($C = 0.09$) small world ($L = 2.87$) where L grows only logarithmically with n. By adding a few long-range edges (short cuts) to connect vertices that are farther apart in network (a), they obtained network(b), which is a highly clustered ($C = 0.45$) small world ($L = 3.99$). This network inherits the high clustering property from network (a) and also have the small distance similar to network (c). L in this small-world network grows only logarithmically with n.

With such understanding of small-world networks properties, they evaluated L and C of 3 large and sparse networks: the collaboration network of film actors, the electrical power grid network of the Western United States and the neural network of the nematode worm *C. elegans*. Their studies show that all these three networks have the small-world properties: *small* L and *large* C. They

therefore suggest that small-world phenomenon is common in most large and sparse networks.

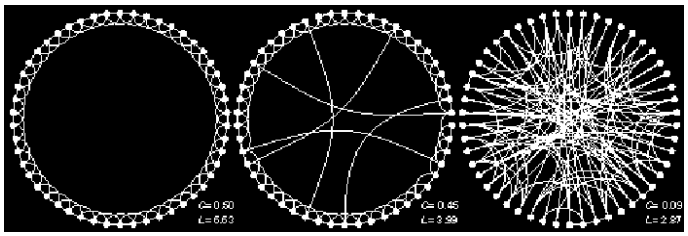

Fig. 2. Three different networks (a)(b)(c) with different L and C.

3 Neutrality in Evolutionary Search Network

Evolutionary algorithms perform searches within a space of possible solutions. The search space can be defined as a network where each vertex is a possible solution while each edge connects two solutions that can be transformed from one to the other in one operation step. The commonly used transformation operators are selection and mutation (in various forms). These operators navigate the search step by step to find the target solution.

Multiple solutions in a search network may have the same fitness. The subnetwork that connects solutions with the same fitness is called *neutral network*. Within a neutral network, evolutionary search walks randomly without the guidance of fitness. In other words, the only kind of transformation operation is mutation (called *neutral mutation*), since selection pressure has no effect under the condition of equal fitness. For the reason of simplicity, one-point mutation is the only transformation mechanism considered in this study.

A solution can have a dual-representation of genotypes and phenotypes. With this representation, mutations take place in genotypes while fitness evaluation and selection are based on corresponding phenotypes. Also, the mapping from genotypes to phenotypes and from phenotypes to fitness can be many-to-one. This means many genotypes may have the same phenotype and many phenotypes may have the same fitness. This dual-representation provides two ways to define the search networks: genotype networks and phenotype networks. Since we are interested in the characteristics of search networks with neutrality, phenotype network is a better model for this study.

4 Structural Properties of Boolean Function Networks with Neutrality

This section analyzes two Boolean function networks with neutrality. The first phenotype network is based on a one-to-one genotype-phenotype mapping rep-

resentation. Neutrality in this case is implicit: many phenotypes may have the same fitness. The second phenotype network is based on a many-to-one genotype-phenotype mapping representation. Neutrality in this instance is both implicit and explicit, through redundant genes in the genotypes. (see [10] for more discussions on implicit and explicit neutrality). Both networks have a small number of vertices to make the analysis of small-world properties easier. We will study larger and sparse Boolean function networks in a later work.

4.1 A One-to-One Genotype-Phenotype Mapping Representation

The Boolean function studied is odd-3-parity. This function takes three Boolean inputs and returns *True* if an odd number of inputs are *True*. We use only *xor* to construct this function. As analyzed in [11, 12], the fitness landscape is needle-in-haystack: a phenotype either gets every test case correct (a needle) or half of the test cases correct (a hay). With 8 test cases, a phenotype has either a fitness of 4 or a fitness of 8.

The genotype is a string of integers that are inputs to a program parse tree. Each genotype has 2 nodes, which is the minimum number of nodes required to construct a correct odd-3-parity. In each node, there are two input values to an *xor* function. The gene values range from 0 to 2, denotes inputs to the odd-3-parity (x_0, x_1, x_2). Figure 3 gives an example genotype and its corresponding phenotype. With 4 genes in each genotype, each has 3 possible values, the total number of genotypes is $3^4 = 81$, which is also the number of vertieces in the phenotype network.

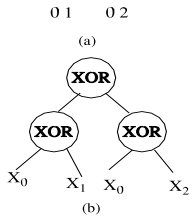

Fig. 3. A genotype (a) and its phenotype (b) under a one-to-one mapping representation.

Since the mapping between genotypes and phenotypes is one-to-one, the two are identical: each vertex in the phenotype network is a 4-integers string. Two vertices are connected if they can be transformed by an one-point mutation. The distance between each pair of vertices in the network is simply their Hamming distance.

Figure 1 gives a sub-network of the phenotype network. As shown, each vertex has 8 neighbors. Each pair of vertices has a distance between 1 and 4. The characteristic path length, L, of the network is:

$$L = \frac{\sum_{i=1}^{T} \sum_{j=i+1}^{T} d(i,j)}{\binom{T}{2}} \tag{1}$$

where *d(i,j)* is the Hamming distance between two vertices v_i and v_j; T is the total number of vertices in the network, which is 81 in this case. The characteristic path length L for this network is 2.7.

Each vertex in the network has 8 neighbors (see Figure 1). The number of edges among the 8 neighbors is 4 ($|E(\Gamma_v)| = 4$). The local clustering coefficient C_v is therefore:

$$C_v = \frac{4}{\binom{8}{2}} = 0.14285.$$

Since the overall network is regular in that every vertex has the same number of edges and the same connectivity pattern, the clustering coefficient of all vertices are the same. The clustering coefficient C of the network, which is the averaged C_v over all vertices, is therefore 0.14285.

Figure 4 gives the phenotype network. It is a poorly clustered small world, similar to network (c) in Figure 2. As the number of phenotypes n increases (through the increase of genotype length), we expect L to grow logarithmically. This will be verified in our future work.

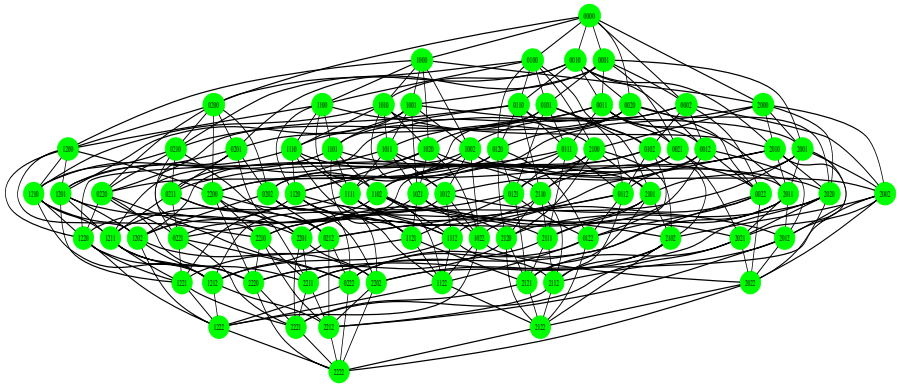

Fig. 4. The Boolean function network with a one-to-one genotype-phenotype mapping representation ($L = 2.7, C = 0.14285$).

4.2 A Many-to-One Genotype-Phenotype Mapping Representation

The many-to-one genotype-phenotype mapping representation in loosely based on the Cartesian Genetic Programming (CGP) system[6]. The genotype is a string of integers that encode an indexed graph. Each node in the genotype contains many genes; some of them are link values and some are function values.

Not all nodes in the genotype are expressed in the phenotype only those that are active. A node is active if its link value is referred by another active node. Since many genotypes may have the same active nodes, hence are mapped into the same phenotype, this representation gives a many-to-one mapping between genotypes and phenotypes.

Figure 5 gives an example genotype and its phenotype for the odd-3-parity function. Similar to the one-to-one mapping representation, the genotype has 2 nodes; each has 2 genes. Unlike the previous representation, the gene values can be either an input to the odd-3-parity function, denoted by labels 0 to 2 in the genotype, or a node output link, which has a label 3. The last node (with label 4) is the final output node.

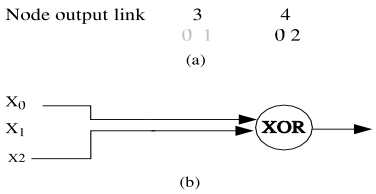

Fig. 5. A genotype (a) and its phenotype (b) under a many-to-one mapping representation.

The mapping of a genotype to its phenotype starts from the last node on the right. This is the final output node (with genes 0 2), which is active by default. This XOR node has its two inputs connected to the odd-3-parity input x_0 and x_2. The node with output link 3 is not expressed in the phenotype because it is not referenced by the only active node 4. This inactive node is grayed in the genotype.

With 2 nodes in each genotype, each can be linked to the output of a lower-number node or the inputs to the odd-3-parity, the total number of possible genotypes is $3^2 \times 4^2 = 144$. In general, for odd-N-parity with l nodes in each genotype, the total number of genotypes in the search space is:

$$\prod_{i=0}^{l-1}(N+i)^2$$

A genotype may have one or two active nodes. Since only active nodes are mapped to the phenotype, a phenotype can be either 1-node or 2-nodes. In the example given in Figure 5, the 2-node genotype `0102` is mapped to a 1-node phenotype `02`. Figure 6 shows the phenotype sub-network where each vertex is either 1-node (2-integers) or 2-nodes (4-integers).

If the phenotype is 2-integers, its genotype is `**[0..2][0..2]`, where * denotes "don't care". Hence, the number of 2-integers phenotypes is $3^2 = 9$. If the phenotype is 4-integers, the genotype is `[0..2][0..2]3[0..2]` or

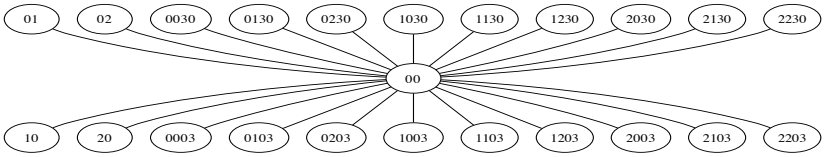

Fig. 6. The phenotype sub-network where some vertices are 2-integers and some are 4-integers.

`[0..2][0..2][0..2]3` or `[0..2][0..2]33`. The number of 4-integers phenotypes is therefore $3^3 + 3^3 + 3^2 = 63$. The total number of phenotypes is 72.

A 2-integers phenotype has its associated genotype with two "don't care" genes, each can be any of the 3 inputs to the odd-3-parity. The mapping between genotypes and phenotypes is therefore 9-to-1. In contrast, every 4-integers phenotype corresponds to one 4-integers genotype, i.e. the mapping is 1-to-1.

Since some phenotypes are 2-integers and some are 4-integers, the distance between every pair of phenotypes in the network is no longer the straight-forward Hamming distance. The following gives the algorithm that calculates $d(i, j)$ in this network:

```
if both v_i and v_j are 2-integers
   d(i,j) = Hamming(v_i,v_j);
else if both v_i and v_j are 4-integers
   d(i,j) = Hamming(v_i,v_j);
else if v_i(non-3-gene) == v_j(non-3-gene)
   d(i,j) = 1;
else
   d(i,j) = 2;
```

When both phenotypes have the same length, their Hamming distance is their distance. When one of the phenotypes has 2 integers and the other has 4 integers, their distance can be either 1 or 2, depending on their integer values. A phenotype with 4 integers means both nodes in the associated genotype are active and are linked by a gene value 3. When this link gene is mutated to a different value, the two nodes are no longer linked; only one of them remains active while the other becomes inactive. As a result, the original 4-integers phenotype becomes a 2-integers. For example, changing gene value "3" in the genotype `0031` to "2" leads to genotype `0021`, which is mapped to a 2-integers phenotype `21`. If this link is the only different gene in the active node, their distance is 1. Additionally, if the other link gene in the active node is also different, their distance is 2. For example, $d(\texttt{0231}, \texttt{20}) = 2$ and $d(\texttt{0031}, \texttt{21}) = 1$.

Using this distance function $d(i, j)$ and a T value of 72, we apply Equation (1) to compute the characteristic path length for this network. The resulting L is 2.387324.

The network clustering coefficient is calculated as follows. Each 2-integers phenotype has 22 neighbors; 4 of them have length 2 and 18 have length 4 (see Figure 6). The number of neighbors with length 2 is easy to count: each of the two integers may be mutated to one of the 2 other inputs to odd-3-parity. So, the number of neighbors is $(3-1)^2 = 4$. Counting the number of neighbors with length 4 has two parts: 1) the first integer value is mutated into 3; this leads to 3^2 possible phenotypes. 2) the second integer value is mutated into 3; this leads to another 3^2 possible phenotypes. So, the total number of neighbors with length 4 is 18. With 22 neighbors and 74 edges among them, the clustering coefficient for this type of phenotype is:

$$C_{length2} = \frac{74}{\binom{22}{2}} = 0.32034$$

Each 4-integers phenotype has 10 neighbors. There are 64 such kind of phenotypes and 54 of them have neighbors with length 2 while 9 don't. In the first group of 54, each phenotype has 3 (N) neighbors of length 2 and 7 ($(N-1)+(N-1)+9N+1-1$)) neighbors of length 4 (see Figure 7). The number of edges among these neighbors is 10. In the second group of 9, each phenotype has 10 neighbors; all of them are of length 4 (see Figure 8). The number of edges among these neighbors is 8 ($(N-1)+(N-1)+N+N$). The clustering coefficient for these two phenotype groups (a and b) are:

$$C_{length4_a} = \frac{20}{\binom{10}{2}} = 0.44444$$

$$C_{length4_b} = \frac{8}{\binom{10}{2}} = 0.17777$$

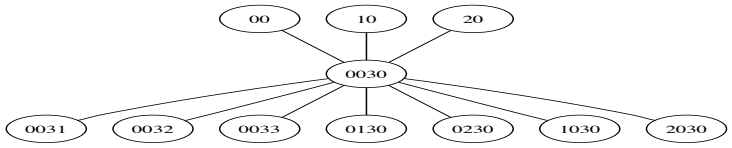

Fig. 7. A 4-integers phenotype (group a) has 10 neighbors.

The clustering coefficient of the Boolean function network is the average over the clustering coefficient of the 72 phenotypes: $C = 0.3955$.

Figure 9 gives the phenotype network. The network has similar properties as network (b) in Figure 2: small L and large C. Thus, this is a small-world network. When the number of phenotype n increases (through the increase of genotype length), we expect L to grow logarithmically. This will be verified in our future work.

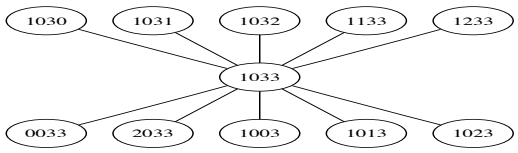

Fig. 8. A 4-integers phenotype (group b) has 10 neighbors.

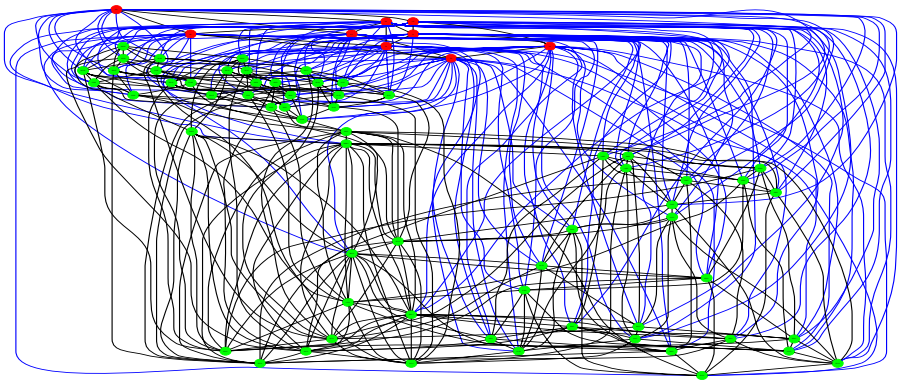

Fig. 9. The Boolean function network with a many-to-one genotype-phenotype mapping representation ($L = 2.38734, C = 0.395599$).

5 Analysis and Discussions

Table 5 summarizes structural properties of the two studied Boolean function networks. The many-one mapping representation has a higher connectivity (k) because a mutation on the link value of an active node can make inactive nodes active and vice versa. As a result, more phenotypes become accessible in one mutation step. Also, the Boolean function network based on the one-to-one mapping representation does not have the small-world structure while the Boolean function network based on the many-to-one mapping representation does. We believe one reason for such a difference is the "short cuts" introduced by redundant genes in the genotypes.

As demonstrated, this many-to-one mapping representation allows some genes in a genotype not expressed in the phenotype, hence become redundant. However, these redundant genes can be activated and become a part of a phenotype which is very different from the original phenotype. For example, the genotype `1021` is mapped to phenotype `21`; the first two genes `10` are redundant. However, when the gene value "2" is mutated to a "3", its phenotype becomes `1031`, a very different phenotype from the original `21`. Without these redundant genes, their distance would be farther in the network. In general, short cuts exist between every pair of short (2-integer) and long (4-integers) phenotypes. Such short cuts

Table 1. Structural properties of the two studied Boolean function networks.

	one-one mapping	many-one mapping
n	81	72
k	8	10 or 22
$ln(n)$	4.394	4.2766
L	2.7	2.38734
C	0.14285	0.395599

increase the network clustering coefficient C and reduce (slightly) the characteristic path length L.

Shorter phenotypes have twice more links than the longer phenotypes have. In other words, shorter phenotypes are hubs in the network and are visited more frequently during the evolutionary search. As shown on the left top corner of Figure 9, these hubs (9 of them) have a higher connectivity. Although small in number, hubs play an important role in small-world networks. In addition to providing short cuts, they also make the whole network connected. Removing some of the hubs, according to [2], a small-world network is no longer connected.

Although 2 data points are statistically meaningless, the link distribution in the Boolean function network suggests that it might follow power law. Recall that power law distribution does not have a peak as that of a Gaussian distribution. Instead, the distribution has a continuously decreasing curve. With this link distribution, there are a few hubs and many vertices having a small number of edges. Every power law is characterized by a unique exponent telling how many hubs are there in the network relative to the non-hub vertices. This exponent gives the search bias in the network.

The purpose of placing evolutionary search space in a small-world network framework is to help us design efficient evolutionary algorithms to navigate search. We have made the first step of identifying a small-size Boolean function network to be small-world. We have also explained some important small-world structures, such as hubs and power law link distribution. Our next goal is to investigate if we can use these properties to design evolutionary algorithms that finds the shortest path in the small-world networks.

At the meantime, we have to address questions on how our findings can be applied to networks with a larger number of vertices. For example:

- Would the network still have small-world properties ?
- Would the characteristic path length L grow logarithmically?
- Would the link distribution follow the power law as that observed in some small-world networks, such as World-Wide-Web [1]?
- What is the search bias under such link distribution ?

6 Concluding Remarks

Modeling search space based on the small-world network structure is a new approach to study evolutionary search in fitness landscapes with neutrality. We

have demonstrated that a Boolean function network based on a many-to-one genotype-phenotype mapping representation is small-world. This opens the possibility of applying small-world networks research to the general field of Evolutionary Computation.

Small-world networks are very rich in structures. We have discussed some of them such as clusters, hubs and power law link distribution. Such properties are valuable assets to the design of effective evolutionary algorithms in navigating the search of solutions.

The investigated networks, however, are very small. There remains many questions on how our findings can be applied to typical evolutionary search networks which are much larger in size. We acknowledge the gap and continue our efforts to address those open issues.

Acknowledgments

I would like to thank Center for the Study of Complex Systems at University of Michigan for providing me the environment to conduct this research.

References

1. Albert, R., Jeong, H. and Barabási, A-L: Diameter of the World Wide Web. Nature **401** (1999) 130-131
2. Barabási, A-L: Linked, The New Science of Networks. Perseus Publishing, 2002.
3. Bollábas, B: Random Graphs. Academic London, 1985.
4. Kleinberg, J. M.: Navigation in a small world. Nature **406** (2000) 845
5. Milgram, S.: The small world problem. Psychology Today **2** (1967) 60–67
6. Miller, J. F. and Thomson, P.: Cartesian Genetic Programming. Proceedings of the Third European Conference on Genetic Programming. Spring Verlag, (2002) 121–132
7. Tjaden, B. and Wasson, G.: The "Oracle of Bacon" website. www.cs.virginia.edu/bct7m/bacon.html.
8. Watts, D.,J., Strogatz, S. H.: Collective dynamics of 'small-world' networks. Nature **393** (1998) 440–442.
9. Williams, R. J., E. L. Berlow, J. A. Dunne, A-L Barabsi. and N. D. Martinez: Two degrees of separation in complex food webs. Proceedings of the National Academy of Sciences (2002) 99:12913-12916.
10. Yu, T., Miller, J.: Neutrality and the evolvability of Boolean function landscape. Proceedings of the Fourth European Conference on Genetic Programming. Spring Verlag, (2001) 204–211.
11. Yu, T., Miller, J.: Finding needles in haystack is not hard with neutrality. Proceedings of the Fifth European Conference on Genetic Programming. Spring Verlag, (2002) 13–25.
12. Yu, T., Miller, J.: Through the interaction of neutral and adaptive mutations, evolutionary search finds a way. Submitted to Genetic Programming and Evolvable Machines.