Fractals, Vol. 14, No. 2 (2006) 133–142

COMBINATORIAL FRACTAL GEOMETRY WITH A BIOLOGICAL APPLICATION

JOHN KONVALINA,* IGOR KONFISAKHAR, JACK HEIDEL and JIM ROGERS
Department of Mathematics, University of Nebraska at Omaha
Omaha, NE 68182-0243, USA
**johnkon@unomaha.edu*

Received June 29, 2005
Accepted October 3, 2005

Abstract

The solution to a deceptively simple combinatorial problem on bit strings results in the emergence of a fractal related to the Sierpinski Gasket. The result is generalized to higher dimensions and applied to the study of global dynamics in Boolean network models of complex biological systems.

Keywords: Fractal Geometry; Combinatorics; Sierpinski Gasket; Box-Counting Method; Stirling Numbers; Boolean Networks; Entropy.

1. INTRODUCTION

The relationship between combinatorics and fractal geometry[1] is beautifully illustrated by construction of the Sierpinski Gasket from Pascal's Triangle by reducing the binomial coefficients modulo two.[2] In this paper, we show how a relatively simple combinatorial problem on comparing two bit strings of length m leads to the construction of a fractal that is an interesting variant of the Sierpinski Gasket. Moreover, by generalizing the combinatorial problem to comparing k bit strings of length m, we construct a sequence of higher-dimensional fractals. An unexpected consequence of this generalized fractal construction is a direct link to the global dynamical behaviors arising in the study of certain classes of Boolean network models of complex biological systems.

Consider the following simple combinatorial question: Given two bit strings of length, say 4, written as rows of a 2×4 matrix, how many distinct columns are there? The columns are bit strings of length 2, and there are 4 possible column types, namely, 00, 01, 10, and 11. For example, the two bit strings, 0011 and 1011, written as rows in the following 2×4 matrix: $\begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$ contain 3 distinct columns, denoted by **01**, **00**, and **11** (the last column **11** is a duplicate). Now define the *pairing number* of any two bit strings of length m as the number of *distinct* columns in the matrix representation of the bit strings. Thus, the pairing number ranges from 1 to 4, inclusive. Denote the pairing number of two bit strings, A and B, as $p(A, B)$. For example, we have $p(0011, 1011) = 3$. Suppose we have two arbitrary bit strings of length m and we compute the pairing number for all possible pairs of bit strings. There are 2^m bit strings of length m and, consequently, 2^{2m} pairing numbers. Intuitively, given the simplicity of the definition one would expect the pairing number function to be reasonably well-behaved and, perhaps, computable by known combinatorial functions, such as the binomial coefficients. However, we will show there is a fractal nature to this deceptively simple combinatorial problem that seems to defy our intuition, but at the same time contains far-reaching consequences.

2. CONSTRUCTION OF THE K2 FRACTAL

Before defining and generating the fractal, we note the pairing number function has a corresponding complementary function that will be used in the application (Sec. 4 below). We define the *complementary pairing number*: $\tilde{p}(A, B) = 4 - p(A, B)$. This function ranges from 0 to 3 and counts the number of missing column types from the 4 possible types, **00**, **01**, **10**, and **11**. For example, $\tilde{p}(0011, 1010) = 0$, since the pairing number is 4 and there are no missing types. Figure 1 depicts a 16×16 square array consisting of 256 unit squares containing all possible complementary pairing numbers with two input bit strings A and B of length 4. This square array is known as the degrees of freedom space for $m = 4$ and $K = 2$. Observe that all pairs of bit strings having the maximum pairing number of 4 are shown in Fig. 1 with the complementary number **0**. We will call the set of unit squares with complementary number 0 the *control region*. In the subsequent application we will show that the control region can be interpreted as a measure of the amount of "order" in a system.

m=4; Input A \ Input B	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3
0001	2	2	1	1	1	1	1	1	1	1	1	1	1	1	2	2
0010	2	1	2	1	1	1	1	1	1	1	1	1	1	2	1	2
0011	2	1	1	2	1	**0**	**0**	1	1	**0**	**0**	1	2	1	1	2
0100	2	1	1	1	2	1	1	1	1	1	1	2	1	1	1	2
0101	2	1	1	**0**	1	2	**0**	1	1	**0**	2	1	**0**	1	1	2
0110	2	1	1	**0**	1	**0**	2	1	1	2	**0**	1	**0**	1	1	2
0111	2	1	1	1	1	1	1	2	2	1	1	1	1	1	1	2
1000	2	1	1	1	1	1	1	2	2	1	1	1	1	1	1	2
1001	2	1	1	**0**	1	**0**	2	1	1	2	**0**	1	**0**	1	1	2
1010	2	1	1	**0**	1	2	**0**	1	1	**0**	2	1	**0**	1	1	2
1011	2	1	1	1	2	1	1	1	1	1	1	2	1	1	1	2
1100	2	1	1	2	1	**0**	**0**	1	1	**0**	**0**	1	2	1	1	2
1101	2	1	2	1	1	1	1	1	1	1	1	1	1	2	1	2
1110	2	2	1	1	1	1	1	1	1	1	1	1	1	1	2	2
1111	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3

Fig. 1 Degrees of freedom space for $m = 4$ and $K = 2$. The row and column labels are all possible bit strings of length 4. The entries are the complementary pairing numbers with the **0**'s indicating bit string pairs having maximum pairing number 4.

The complementary pairing numbers can be interpreted as colors, denoted by 0, 1, 2, or 3 representing *degrees of freedom* in our system application. The 4 corners of Fig. 1 have pairing number 1, so color 3 is used. The pairing number of any bit string with itself (except the bit string of all 0's) is 2, so the color $4 - 2 = 2$ is used. A similar argument applies to a bit string and its complement. Thus, the two diagonals in Fig. 1 will have color 2. Observe that if B is any nontrivial bit string (not all 0's or all 1's), then

$$p(0000, B) = 2, \quad p(B, 0000) = 2,$$
$$p(B, 1111) = 2, \quad p(1111, B) = 2.$$

So, the 4 sides of Fig. 1 except for the four corners will have color 2 as well. The remaining unit squares will be colored 1 or 0, depending on whether the pairing number is 3 or 4, respectively. We will call the set of unit squares with a nonzero coloring number the *freedom region*. Accordingly, in our

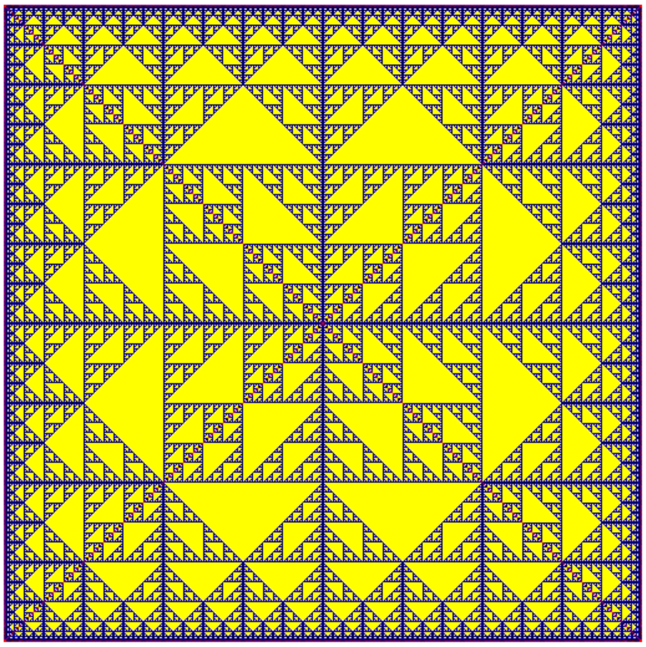

Fig. 2 Degrees of freedom space for $m = 9$ and $K = 2$. Yellow = zero degrees of freedom, blue = one degree of freedom, red = two degrees of freedom. The four corner points represent the only points with three degrees of freedom. Continued iteration results in the "K2" fractal as m approaches infinity.

application below we will show that the freedom region can be interpreted as a measure of the amount of "disorder" in a system.

We can repeat the combinatorial procedure for bit strings of any length m and construct a $2^m \times 2^m$ square array, compute the pairing numbers, and color the unit squares accordingly. Figure 2 depicts the resulting construction with bit strings of length $m = 9$ (known as the degrees of freedom space for $m = 9$ and $K = 2$). This complex geometrical structure exhibits symmetry, self-similarity, and a fractal structure reminiscent of the Sierpinski Gasket. Since our construction can be iterated as the length m of the two bit strings approaches infinity, what is the fractal dimension of the freedom region (non-triangular region) in the limiting geometrical structure? We will call the resulting fractal the "K2 fractal." Surprisingly, we now show the fractal dimension of the K2 fractal is identical to the Sierpinski Gasket, namely, $\ln 3/\ln 2$.

We can compute the fractal dimension of the freedom region by first computing the number of unit squares making up the control region (triangular components), and then subtracting from the total number of unit squares (2^{2m}). Suppose we have two input bit strings of length m with pairing number 4, or color 0. This means that at least all 4 column types (**00**, **01**, **10**, **11**) must occur. How many ways can this happen? This problem is equivalent to the combinatorial question: How many ways are there to distribute m labeled balls into 4 labeled boxes (box labels are **00**, **01**, **10**, **11**) with no box empty? The answer is $4!S(m,4)$, where $S(m,4)$ is the number of partitions of a set with m elements into 4 non-empty subsets. The general combinatorial numbers $S(m,n)$ are called *Stirling numbers of the second kind*, and there are exact formulas expressed in terms of binomial coefficients.[3] In fact,

$$n!\,S(m,n) = \sum_{j=0}^{n} (-1)^j \binom{n}{j} (n-j)^m, \qquad (1)$$

and, in our particular case:

$$4!S(m,4) = 4^m - \binom{4}{1} 3^m + \binom{4}{2} 2^m - \binom{4}{3} 1^m.$$

Using the box counting method[4] and our exact formula for $4!S(m,4)$, we compute the fractal dimension D of the freedom region:

$$D = \lim_{m\to\infty} \frac{\log(2^{2m} - 4!S(m,4))}{\log(2^m)} = \frac{\log 3}{\log 2}. \qquad (2)$$

3. GENERALIZATION

Instead of just two bit strings of length m, we can generalize the combinatorial construction and fractal generation in K-dimensional space for K bit strings, where K is any positive integer. Thus, we have K input strings each of length m. The number of possible column types is 2^K. There are 2^{Km} pairing numbers with the coloring numbers ranging in value from 0 to $2^K - 1$. The construction is applied in K-dimensional space with a hypercube of side length 2^m consisting of 2^{Km} unit hypercubes. The freedom region consists of the unit hypercubes with a nonzero coloring number. By the same combinatorial argument used for $K = 2$, the number of unit hypercubes making up the control region (coloring number 0) is $2^K!S(m, 2^K)$. Let D_K denote the fractal dimension of the freedom region in the limiting K-dimensional geometric structure. Then applying (1) with $n = 2^K$, we obtain

$$\begin{aligned} D_K &= \lim_{m\to\infty} \frac{\log(2^{Km} - (2^K!)S(m,2^K))}{\log(2^m)} \\ &= \frac{\log(2^K - 1)}{\log 2}. \end{aligned} \qquad (3)$$

Table 1 Approximate fractal dimensions (entropies) for $K = 1$ to 7.

K	Dimension
1	0
2	1.58
3	2.81
4	3.91
5	4.95
6	5.98
7	6.99

Our combinatorial construction generates a sequence of higher-dimensional fractals with fractal dimension $\ln(2^K - 1)/\ln 2$ for K bit strings. The first seven values are computed in Table 1. When $K = 2$, the fractal dimension reduces to $\ln 3/\ln 2$ for the K2 fractal discussed previously.

4. APPLICATION TO BOOLEAN NETWORKS

Boolean networks have been extensively used to model diverse discrete dynamical systems. The networks were originally studied by Kauffman[5] for biological applications. During the past few decades, Boolean nets have been widely applied in the physical and biological sciences, including in genetic regulatory networks, neural networks and signal transduction.[6–14]

Boolean NK nets are discrete network models consisting of N binary variables denoting the nodes of the network. The nodes can represent objects such as cells, genes, or automata. Each node is regulated by exactly K nodes in the network acting as inputs. Also, the output of each node is determined by a Boolean $(0, 1)$ function of the K input variables. Since each node can be represented as 0 or 1 and there are N nodes, the system can be in any one of 2^N possible states (known as the *state space*) at any given discrete time interval. Given any state of the system, the next state is determined by applying the Boolean output functions in parallel to the current state of the nodes (see the Appendix for more detailed information on Boolean nets). Despite their simplicity Boolean NK nets can exhibit very complex dynamics. For example, when $K = 1$ and each node depends on exactly one node, the dynamics tend to be ordered with relatively short cycles, but when $N = K$ the dynamics tend to be highly disordered with extremely long cycles. As the connectivity K decreases from N to 3, we still observe a large amount of disorder and long cycles. However, the dynamics dramatically change when $K = 2$. Suddenly, we have moderate disorder and moderate cycle lengths. Thus, there appears to be a phase transition when $K = 2$. Why there is a phase transition when $K = 2$ is not clearly understood.[6] We show how the K2 fractal and its generalization provide some interesting insights into this mystery. Also, we show the fractals and their corresponding fractal dimensions in K-dimensional space correlate nicely with the dynamical behaviors for NK Boolean networks as the connectivity K varies.

Consider a generic Boolean net with $N = 3$ and $K = 3$ (Fig. 3a). Since $K = N = 3$, each output function depends on the three input variables A, B, C and takes on some values, say

K=N

A	B	C	f_A	f_B	f_C
0	0	0	x_0	y_0	z_0
0	0	1	x_1	y_1	z_1
0	1	0	x_2	y_2	z_2
0	1	1	x_3	y_3	z_3
1	0	0	x_4	y_4	z_4
1	0	1	x_5	y_5	z_5
1	1	0	x_6	y_6	z_6
1	1	1	x_7	y_7	z_7

(a)

K=1

A	B	C	f_A	f_B	f_C
0	0	0	x_0	y_0	z_0
0	0	1	x_0	y_1	z_0
0	1	0	x_0	y_0	z_1
0	1	1	x_0	y_1	z_1
1	0	0	x_1	y_0	z_0
1	0	1	x_1	y_1	z_0
1	1	0	x_1	y_0	z_1
1	1	1	x_1	y_1	z_1

(b)

K=2

A	B	C	f_A	f_B	f_C
0	0	0	x_0	y_0	z_0
0	0	1	x_0	y_1	z_1
0	1	0	x_1	y_0	z_2
0	1	1	x_1	y_1	z_3
1	0	0	x_2	y_2	z_0
1	0	1	x_2	y_3	z_1
1	1	0	x_3	y_2	z_2
1	1	1	x_3	y_3	z_3

(c)

Fig. 3 State space for Boolean networks with $N = 3$ and K from 1 to N. All variables are equal to either 0 or 1 (e.g. $x_0 = 0$ or 1, etc.). (**a**) When $K = N = 3$, each node is dependent on the values of all nodes in the network. With $2^3 = 8$ possible initial configurations of the three nodes, there are eight possible values for each output function (degrees of freedom, labeled 0 to 7 for each function). Each step through state space defines exactly one variable in each column, reducing the number of degrees of freedom by one. Since the number of initial states equals the number of degrees of freedom in the output function, there are no restrictions on the movement of the network through state space, so it is maximally disordered. (**b**) When $K = 1$, each node is dependent on the value of only one node. (In this case, A is dependent on A, B on C, and C on B.) Because each node is dependent on only one node, there are only two possible values for each node (labeled 0 and 1 for each variable). With only two degrees of freedom, each output function f_A, f_B, and f_C is completely determined by any two steps in state space that define its two variables. Thus there is little freedom in choosing the logic for a particular cycle structure (i.e. the logic is "forced"). (**c**) In between the extremes of $K = N$ and $K = 1$, $K = 2$ has 4 degrees of freedom for each function. Thus, $K = 2$ networks have a balance between freedom and force (control).

$x_0, x_1, \ldots, x_7$. We are free to assign a 0 or 1 to each of these output variables. In assigning values to the output variables we have a certain amount of freedom that depends on the connectivity K. We will refer to the number of free variables as *degrees of freedom* in the Boolean output function. Thus in Fig. 3a, initially, each output function has 8 degrees of freedom. Once an output variable is specified as a 0 or 1, the degrees of freedom will decrease by 1. Therefore, in the case of $K = N$, we have total freedom (maximum disorder) to go from any given state to any other state since the output variables are completely independent of one another. Now consider a generic network with $N = 3$ and $K = 1$ (Fig. 3b). Since $K = 1$, each output function depends on only one of the input variables and takes on some values, say x_0, x_1. In this case each output function has only 2 degrees of freedom. If, for example, we assign $x_0 = 0$, $x_1 = 1$, then the output function has zero degrees of freedom, since the remainder of the state space for that function has been completely determined or "forced." The same argument for $K = 1$ applies independent of the number of nodes N. Thus, there is very little freedom in the output function. In fact once an output variable is specified, half of the state space for that function has been forced. Between the two extremes of $K = 1$ and $K = N$ we find a different kind of behavior for $K = 2$ (Fig. 3c). In this case each output function depends on two of the input variables and takes on values, say x_0, x_1, x_2, x_3. Initially, we start with 4 degrees of freedom, and as we assign values to the output variables the degrees of freedom will decrease until they reach 0 and then force the remainder of the state space. In this case, we have more of a balanced interaction of freedom and control.

Our main objective is to define a complexity measure for each connectivity K that can discriminate between the different degrees of freedom and control for arbitrary subsets of the state space. A consequence of this process is the emergence of the K2 fractal and its generalization discussed previously. Consider $K = 2$ nets for large N. Initially, there are 4 degrees of freedom for each output function, say x_0, x_1, x_2, x_3. If we randomly choose one state from the state space and assign a value of 0 or 1 to one of these variables, the output function will then have 3 degrees of freedom. However, if we randomly pick two states, the corresponding output function will have 2 or 3 degrees of freedom depending on whether or not the output variables were different or the same, respectively. For example, in Fig. 3c if we choose states $(0, 0, 0)$ and $(0, 0, 1)$ and assign x_0 a value of 0 or 1, the output function f_A will have 3 degrees of freedom since x_1, x_2, and x_3 are free, but if we choose states $(0, 0, 0)$ and $(0, 1, 0)$ and assign values to x_0 and x_1, then f_A will have 2 degrees of freedom. The same argument applies to each output function. So after arbitrarily choosing, say m states, and assigning output values, each output function independently will have either 3, 2, 1, or 0 degrees of freedom.

Now consider the ensemble of all possible degrees of freedom for an output function when choosing m states. For example, Fig. 1 depicts the ensemble of the degrees of freedom space for $K = 2$ and $m = 4$. The row and column labels are all possible values of any 2 input variables taken from the 4 states. The entries in the table are the corresponding degrees of freedom. They are computed as follows: for $K = 2$, there are 4 possible bit pairs 00, 01, 10, and 11. These pairings correspond to the output variables x_0, x_1, x_2, x_3, respectively. So if the input strings are 0000 and 0000 (row 1 and column 1 of Fig. 1), and we compare these strings bit by bit, then we essentially have one distinct pair 00 (or just x_0), and the output function will have 3 degrees of freedom. However, if the input strings are 0011 and 0101 and we compare them bit by bit (left to right) we obtain the pairs 00, 01, 10, 11, corresponding to the output variables x_0, x_1, x_2, x_3. Thus, in this case all the variables have been assigned values and we have zero degrees of freedom as indicated in Fig. 1. In the degrees of freedom space, the nonzero degrees represent the freedom region and the zero degrees represent the control region. This ensemble depicts the various degrees of disorder and order possible when $m = 4$. Our next step is to quantify this observation and define an entropy measure for any connectivity K and any number of states m. The basis for this procedure is the underlying emergent fractal.

If for $K = 2$ we construct the degrees of freedom space for each positive integer m, we obtain a $2^m \times 2^m$ square (consisting of 2^{2m} unit squares) partitioned into two major regions, the freedom region (dark area) and the control region (light area). Figure 2 depicts the square with $m = 9$. The light triangular areas correspond to the zero degrees of freedom (control region).

The fractal dimension computed in (2) can be interpreted as a global entropy measure of the amount of disorder in $K = 2$ nets. Also, the numerical value of the mth iterate can be interpreted as a local ensemble entropy measure for each m. For example, the dimension (D) of the K2 fractal is $\log 3/\log 2$, or about 1.58, and this number can be interpreted as a global entropy measure for Boolean nets with $K = 2$, and is independent of the number of nodes N. Also, for each m the ratio of the above logarithms measures the local disorder for the freedom region with m states. We call this the *ensemble entropy* for the degrees of freedom space with m states. The ensemble entropy for $m = 4$ in Fig. 1 is $\log(256-24)/\log(16)$, or about 1.96, indicating very high entropy (freedom), since the maximum entropy is 2. Similarly, the fractal dimensions computed in (3) can be interpreted as a global entropy measure of the amount of disorder in Boolean networks with connectivity K.

The fractal dimensions for Boolean networks with connectivity K from 1 to 7 are shown in Table 1. Observe when $K = 1$ the dimension (entropy) is 0, implying negligible disorder and almost total order. Also, as K increases from 2, the entropy increases and approaches the maximum (total disorder) as K increases. However, for $K = 2$ we have a fractal dimension of $\log 3/\log 2$, or about 1.58, indicating moderate disorder. Thus, $K = 2$ (and, to a lesser extent, $K = 3$) networks are unique in that they are not near total order or total disorder. This highly fractal nature indicates at $K = 2$ a phase transition exists.[6] The phase transition between order and chaos at $K = 2$ is illustrated in Fig. 4.

Note that as K increases the fractal dimensions are very close to the maximum entropy indicating potential for large-scale disorder but, importantly, not *total* disorder. Even with high K it is still possible to construct complex networks that are neither trivial nor chaotic, but the logic must be selected carefully in order for this to be achieved (see Appendix A.4).

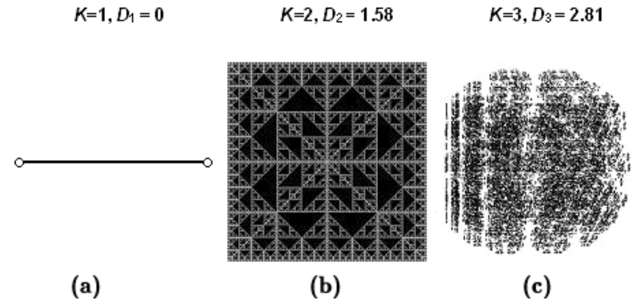

Fig. 4 The phase transition from order to chaos at $K = 2$. The figures for each K were generated by calculating the degrees of freedom space for $m = 9$. Regions of the freedom space with zero degrees of freedom ("forced") are colored black, while regions with one or more degrees of freedom ("free") are white. The dimensions shown (D_1, D_2, and D_3) are of the white areas at the limit as m approaches infinity. The dimensions of the black regions are one, two, and three, respectively, as m approaches infinity. (**a**) At $K = 1$ the free region (the two points at the ends of the forced region line) has zero dimension (D_1) and thus is not fractal. With such a low freedom/forcing ratio, $K = 1$ systems would be expected to be highly ordered. (**b**) At $K = 2$ there is near balance between the order and the free regions as indicated by the highly fractal dimension of the free area (D_2). This fractal is neither highly ordered nor highly disordered, indicating that the system has potential for nontrivial behavior. (**c**) At $K = 3$ there is more free region as indicated by the fact that the fractal dimension of the free area is closer to integer (D_3). This fractal indicates that $K = 3$ systems are likely to have chaotic dynamics. As K increases above three, the freedom spaces cannot be pictured because the dimension of the black area is more than three. However, the fractal dimension moves closer to integer as K increases (Table 1), indicating more and more freedom in the system. Since the freedom space of $K = 2$ networks is the most fractal (balanced between freedom and forcing) of any K, it indicates that there is a phase transition from order to chaos at $K = 2$.

5. CONCLUSIONS

In this paper, we have shown how "combinatorial fractal geometry" can help us gain a deeper understanding of the dynamical behaviors observed in NK Boolean network models of complex systems. Despite the application to a restricted class of discrete models, the results seem to accentuate the significance of the fractal nature of complex network dynamics. One of the major challenges of contemporary science is to understand the nature of complex systems ranging from social networks to molecular networks. Extensive research on real complex networks has focused on static properties including degree distributions, scaling, clustering, and other structural features.[15] The hope is that the architectural properties will yield some clues regarding the dynamics of the networks. Many of these networks are scale-free with the degrees of the nodes having a power law distribution. Recently, it was shown that some of these networks possess self-similar structures.[16] Thus, the existence of a power law and self-similarity suggests an underlying fractal structure may exist and, perhaps, provide some insights on the dynamics. The relationship between complex networks and fractals remains an enigma.[17] In many complex systems the competing forces of freedom and control tend to shape the dynamics. Too much freedom in a system can lead to chaotic, disordered behavior, while too much control can result in totally ordered dynamics. Complex systems seem to function effectively on the border of order and disorder where phase transitions occur and fractals emerge.

In conclusion, we have analyzed an extensively studied class of discrete networks and shown that the dynamics of these networks expressed in terms of their given logical functions are intimately linked to emergent fractals. Applying a combinatorial approach to the state space, we defined a complexity (entropy) measure that can discriminate between the different degrees of freedom and control for arbitrary subsets of the state space. We demonstrated the existence of a global phase transition between order and disorder of these networks that provided some mathematical evidence for the fractal nature of complex network dynamics. The challenge for future research is to extend these results to real complex networks. The presence of power law distributions in real networks suggests the existence of underlying fractals that can provide useful global information for understanding the observed network dynamics.

ACKNOWLEDGMENTS

We thank Deborah Eveleth for generating the graphic of Fig. 4c. This work was supported by NIH/NIGMS (Grant No. 5-R01-GM067272-03).

REFERENCES

1. B. B. Mandelbrot, *The Fractal Geometry of Nature* (W. H. Freeman, New York, 1982).
2. H. O. Peitgen, H. Jürgens and D. Saupe, *Chaos and Fractals* (Springer, New York, 2004).
3. L. Comtet, *Advanced Combinatorics* (Springer, New York, 1974).
4. K. J. Falconer, *Fractal Geometry* (Wiley, Chichester, UK, 2003).
5. S. Kauffman, Metabolic stability and epigenesis in randomly constructed genetic nets, *J. Theor. Biol.* **22** (1969) 437–467.
6. S. Kauffman, *The Origins of Order* (Oxford, New York, 1993).
7. R. Solé and B. Goodwin, *Signs of Life*: *How Complexity Pervades Biology* (Basic Books, New York, 2000).
8. S. Huang, Genomics, complexity and drug discovery: insights from Boolean network models of cellular regulation, *Pharmacogenomics* **2**(3) (2001) 203–222.
9. C. Farrow, J. Heidel, J. Maloney and J. Rogers, Scalar equations for synchronous Boolean networks with biological applications, *IEEE Trans. Neural Netw.* **15** (2004) 348–354.
10. J. Heidel, J. Maloney, C. Farrow and J. Rogers, Finding cycles in synchronous Boolean networks with applications to biochemical systems, *Int. J. Bifurcat. Chaos* **13**(3) (2003) 535–552.
11. M. Aldana, S. Coppersmith and L. Kadanoff, Boolean dynamics with random couplings, in *Perspectives and Problems in Nonlinear Science*, eds. E. Kaplan, J. Marsden and K. Sreenivasan (Springer, New York, 2002), pp. 23–89.
12. W. Just, I. Shmulevich and J. Konvalina, The number and probability of canalizing functions, *Physica D* **197** (2004) 211–221.
13. R. J. Bagley and L. Glass, Counting and classifying attractors in high dimensional dynamical systems, *J. Theor. Biol.* **183** (1996) 269–284.
14. S. E. Harris, B. Sawhill, A. Wuensche and S. Kauffman, A model of transcriptional regulatory networks based on biases in the observed regulation rules, *Complexity* **7**(4) (2002) 23–40.
15. A.-L. Barabasi and R. Albert, Emergence of scaling in random networks, *Science* **286** (1999) 509–512.
16. C. Song, S. Havlin and H. Makse, Self-similarity of complex networks, *Nature* **433** (2005) 392–395.
17. S. Strogatz, Romanesque networks, *Nature* **433** (2005) 365–366.

APPENDIX: BACKGROUND ON *NK* BOOLEAN NETWORKS AND THEIR DYNAMICS

NK Boolean (Kauffman) networks[6] are a commonly used modeling system for the study of spontaneous emergence of nontrivial behavior. Although relatively simple, *NK* Boolean networks are able to capture the dynamics systems ranging from trivial to exceedingly complex, including those of living systems.[8]

A.1. The Anatomy of *NK* Boolean Networks

Consider the simple network shown in Fig. A1(a). There are three elements in this network and each element is connected to each of the others. Therefore, the parameter N (the number of elements in the network) is 3 and the parameter K (the number of connections feeding into each element) is 2. The logical connections, shown in the tables in Panel B, are OR and AND. These tables give the on/off state (shown as a 1 for on and a 0 for off) of each element as a function of the on/off state of the other two elements connecting to it. Thus the third table shows that element 3 will be on if either element 1 or 2 (or both) are on. The network exists at time T in some

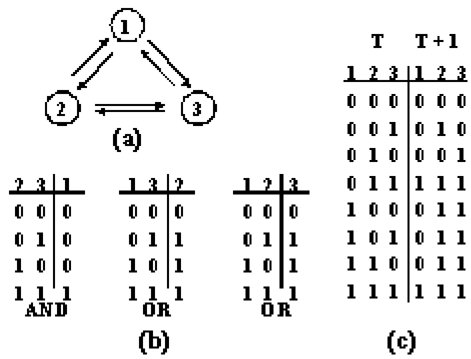

Fig. A1 A simple network and its logical connections.

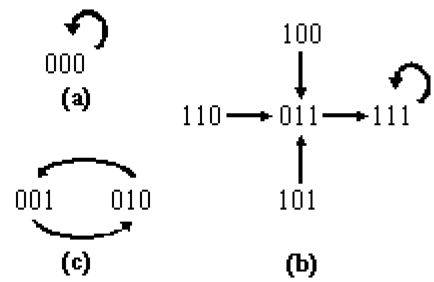

Fig. A2 All possible trajectories and attractors for the network in Fig. A1.

initial state, with each separate element either on or off. At the next time ($T+1$), the states of all three elements will change according to the tables shown. The evaluation of the entire system from time T to time $T+1$ can be represented in a single table [shown in Fig. A1(c)] where the column T contains all the possible initial states of the system and column $T+1$ shows the result of application of the logic set to each initial condition. Continued iteration by the same method results in a trajectory of the system as the states change over time. The trajectory that a given initial condition follows depends on all of the variables and parameters described in this section, and variation of the parameters can radically alter the types of trajectories obtained as will be discussed below.

A.2. Attractors and Basins of Attraction

The network introduced in Fig. A1 is simple enough to view all of the possible trajectories, which are shown in Fig. A2. In panel (a), for example, the system is shown to be at an initial state of element 1 = 0, element 2 = 0 and element 3 = 0, or 000. According to the logic tables in Fig. A1(b) (or, equivalently, the map shown in Fig. A1(c)), at the next time point the system will remain at 000. This trajectory is indicated by the arrow in Fig. A2(a). Similarly, Figs. A2(b) and A2(c) show the trajectories for the other possible starting combinations.

Because there are a finite number of elements in the system (N), there are a finite number (2^N) of possible states of the system. Thus, as the system travels in time, it must (regardless of trajectory) re-enter a state previously encountered. As shown in Figs. A2(a) and A2(b), when the system is at state 000 or 111, it remains there (encountering itself over and over), thus those two states are referred to as steady states. Figure A2(c) shows that if the system is at state 001 or 010, it cycles between those two states, a trajectory that is referred to as a period 2 cycle. Finally, Fig. A2(b) shows that there are four other states of the system (110, 100, 011, and 101) that follow trajectories to the steady-state 111. In summary, Fig. A2 shows that the network described in Fig. A1 has three conditions (namely the two steady-states 000 and 111, and a period 2 cycle containing 001 and 010) called attractors into which the trajectories of all initial states eventually settle.

Basins of attraction are those states whose trajectories lead to a given attractor. For example, the basin of attraction for the steady-state attractor 000 consists only of the condition 000. The basin of attraction for the steady-state attractor 111 is larger, consisting of 110, 100, 011, and 101. The basin for the period 2 cycle consists only of its two states, 001 and 010.

The sizes and stability of attractors and basins of attraction are characteristic features of networks,[13] and allow characterization of networks as ordered, chaotic, or complex as explained in the next section.

A.3. Order, Chaos, and Complexity in *NK* Boolean Networks

Boolean networks can have a wide variety of dynamics. For example, the network in Fig. A3 has an attractor of period six [Fig. A3(d)]. This attractor is considered to be relatively large since the largest possible attractor for this network is period eight. When networks have attractors that are large (relative to the size of the network), they are considered to be minimally ordered . In other words, they are structured in such a way as to have a minimal number of re-encounters with previous states. Furthermore, any network with such a structure would

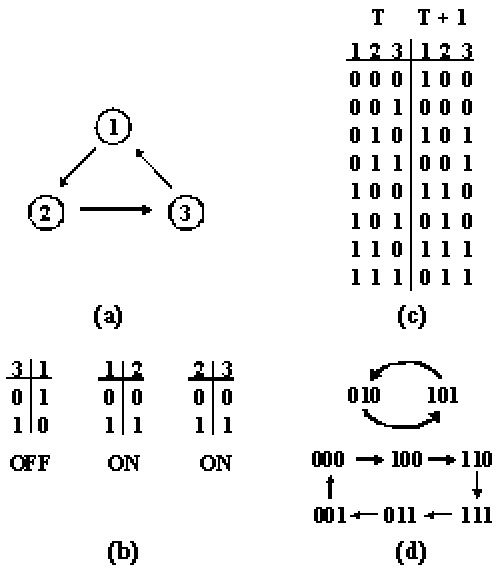

Fig. A3 An NK Boolean network with relatively chaotic dynamics.

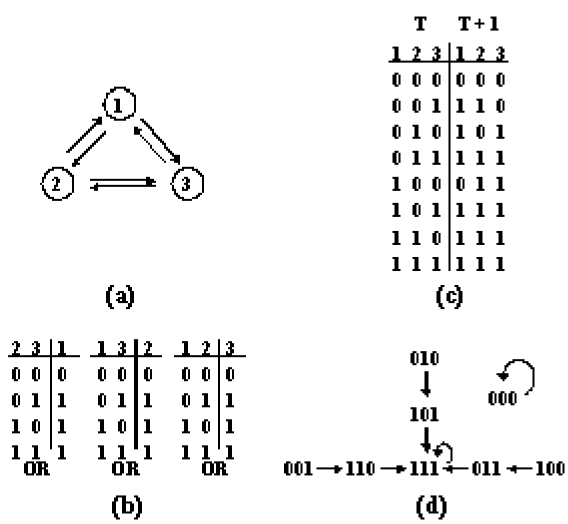

Fig. A4 An NK Boolean network with highly ordered dynamics.

have attractors whose sizes scaled exponentially (by a factor of 2^N) as the number of components in the system (N) is increased. Thus a minimally ordered system with $N = 200$ components could have an attractor of size 2^{200}, or approximately 1.6×10^{60}! An attractor of this magnitude means that the system would never repeat itself in any relevant time scale and could be considered effectively infinite. By definition, a trajectory of this type that is not periodic over (effectively) infinite iteration is called chaotic.

On the other hand, networks can be structured in such a way that the system always moves to small attractors, an example of which is shown in Fig. A4. In this network, all states (except 000) move to the period one (fixed point) attractor 111. Since this network always becomes frozen with all of the nodes on (111) or off (000), this network is considered to be highly ordered.

Between the extremes of order and chaos, some networks have dynamics that are neither fully ordered or fully chaotic. For example, the network shown in Fig. A1 has two fixed point attractors as well as a period two attractor (as shown in Fig. A2). Thus, this network is not clearly classified as ordered or chaotic and networks of this type are termed "complex." Complex networks tend to have a moderate number of moderately-sized attractors. This is strong evidence that complex, "edge of chaos" networks are where nontrivial behavior is expected because they are not frozen on small cycles or chaotically wandering on enormous, effectively infinite cycles.

The $K = 2$ (Kauffman) conjecture states that networks with connectivity $K = 2$ naturally tend to be complex, while $K = 1$ networks tend to be highly ordered and $K > 4$ networks tend to be highly chaotic. $K = 3$ and $K = 4$ networks are also considered to be chaotic and tend toward trivial dynamics, but they are less so than $K > 4$ networks which appear to be very close to maximal disorder. According to this conjecture, by simple virtue of $K = 2$ connectivity, a random network is likely to display nontrivial dynamics that would not be expected in random networks created with higher or lower K. The theoretical foundation for this phenomenon is the increase in the ratio of canalizing functions (see Sec. A.4 below) as K decreases, therefore increasing the likelihood of orderly behavior in a "low-K" network when logic is randomly chosen for each node. Note that in this theory no connectivity is *guaranteed* to have any particular dynamics. For example, the network shown in Fig. A3 is relatively chaotic despite $K = 1$, while the network in Fig. A4 is highly ordered despite the fact that $K = 2$. This is because, regardless of the connectivity, logic can almost always be selected that can create any desired dynamics. However, while it is possible to contrive a network that has any connectivity and any dynamics, the choices of logic in such contrived networks are severely limited. So, while many different choices of logic are available to make a highly connected network chaotic, there are few logical choices that will make a low connected network chaotic. The amount of freedom to choose logic based on the connectivity is precisely the freedom/forcing ratio that is represented by the fractals in this paper.

A.4. Logic Necessary for Nontrivial Dynamics in $K > 2$ Networks

Even though there is much freedom (disorder) in $K > 2$ networks, one can obtain ordered dynamics by introducing more forcing or canalizing functions. Canalizing functions are Boolean functions that include at least one input variable that can determine the output regardless of the values of the other variables. Recently, exact formulas for the number of canalizing functions have been obtained.[12] If the number of input variables K is greater than 4, then there are relatively few canalizing functions. However, biological networks are dominated by canalizing (forcing) functions.[14] Thus, the connectivity in biological networks can be greater than two, but with a correspondingly high number of canalizing functions, there can still be non-chaotic and nontrivial dynamics.