Fearless-Evolution on Boolean Landscapes: Boolean Phase Portraits Reveal a New Navigation Strategy Based on Fearful Symmetry

Thomas E. Malloy, Jonathan Butner, Chase Dickerson & Joel M. Cooper
Department of Psychology, University of Utah, USA

We present two important developments for understanding *nk* Boolean landscapes: First, a discrete analogue to the continuous phase portrait for analyzing Boolean landscapes and, second, an evolutionary alternative to mutation and recombination. Applications include evolution in biological systems, social networks, electrical flow on power grids, business organizations. The Curie Principle states that symmetry in causes will be seen, transformed, in effects. Using our discrete phase portrait we found that ring symmetry in network structure can be found as symmetry groups among the attractors on a Boolean landscape. The transforms which define these symmetry groups constitute a new way to move among groups of attractors (fitness peaks). A system, using only information from its current peak and a locally-generated symmetry transform, can move directly to another peak. Among the implications for theories of adaptation is our proposal of a cogent definition of fitness that goes beyond the 0 to 1 fitness metric.

Introduction

The economic crash of 2008 stressed, if not destroyed, the fitness of many businesses' ability to compete in the market place. Such catastrophes not only challenge organizations but challenge also how we think about the evolution of organizations under stress. In this paper we propose new ways of thinking about important aspects of landscape theory. Using an evolutionary framework for adaptive landscapes, we propose an alternative to the two most prominent evolutionary strategies, mutation and recombination, both of which might lack appeal to organizations at times of crisis: Mutation is a fearful random leap that may well land the organization in a valley of low fitness and recombination is known to keep a system close to its current fitness peak, the very peak that is failing in the extant crisis.

This paper, using an *nk* Boolean network model, will offer a proof of concept that entities that can be construed as networks (such as genetic regulatory networks, cerebral neurology, power transmission grids, and organizations) generate behavioral landscapes whose attractors (stability points, fitness peaks) can fall into symmetry groups. The transforms that define these symmetry groups constitute a new strategy for an entity, when stressed, to evolve its behavior. Moreover, this new strategy will not require the system to have any information about the landscape except that which is local, that is, located in its current position on the landscape. For our derivation, the network that defines the entity must have some degree of symmetry. For our proof of concept we will start with networks that have a high degree of symmetry.

Two additional considerations will arise during our exploration of symmetry groups. First, to demonstrate the symmetry transforms that move a system's behavior from one stability point to another, we must develop a new tool to see the inherent symmetries: a discrete analogue to the continuous phase portrait. Second, this new adaptive strategy will elucidate an alternative to the standard definitions of fitness—the assignment, usually pseudo-randomly, of a 0 to 1 fitness metric to stability points (peaks) on a landscape. Instead, we will propose a cogent definition of fitness as the ability to move nimbly between many stability points under the presumption that stability points will have differential fitness for varying environmental factors and it is the ability to move among stability points that is central to fitness.

Boolean Networks

An *nk* Boolean system consists of a set of *n* binary-valued nodes each of which has a Boolean truth table that expresses its relations to *k* other such nodes. These *n* nodes form a network through their complex couplings, constituting a set of constrained generating procedures (Holland, 1998). Taking an organizational perspective, a Boolean network can be thought of as the analog for the many parts of an organization all working together through rules of operation.

From this network emerges a Boolean landscape usually consisting of many attractors (stability points) along with tributaries leading into attractors. The Boolean landscape is distinct from the network from which it arises and can be thought of as what the network *does* as opposed to what it *is*. Thus, the Boolean landscape is the analog for the varying behaviors of an organization that emerge as a function of the organization's rules of operation. Mitchell (2009: 255ff) makes the distinction between the "*structure* of networks—their static degrees of separation—[as opposed to the] *dynamics* of spreading information in a network." We would soften Mitchell's point; most real world networks are not entirely static but they do evolve much more slowly than does the behavior of the network.

The structure of networks (random graphs, rings, small worlds, scale free) is a discipline in itself (e.g., Barabasi, 2005; Watts, 1999). In contrast, for many mission critical applications the dynamics of whatever spreads across a network is the primary focus of interest—electricity flowing across a network of power lines can produce waves that are responsive to demand. But the electrical flow can be perturbed into a cascade of increasing amplitude that "blacks out" a whole section of a country. It is this dynamic behavior of the network in which we are particularly interested since this array of behaviors dictates organizational success.

The dynamics of the information spreading across a network, to use Mitchell's (2009) terms, can be captured by the idea of a landscape where the flow across the network settles into one of many attractors but can be perturbed into other attractors. The dynamical evolution of systemic behavior on landscapes has been useful in framing theoretical issues in many content areas including biological evolution (Gravner *et al*., 2007), ecological relations such as predator-prey (Brown *et al*., 2007), information (Yossi & Poli, 2005), evolution of the wear on rocket propulsion plates (Farnell & Williams, 2010), and organizational behavior (e.g., Levinthal, 1997; Haslett, 2005; Rhodes & Donnelly-Cox, 2008; Rivkin & Siggelkow, 2002; Winter *et al*., 2007).

The Dynamic Behavior of Networks as an Analog of Organizational Behavior

Systems that can be construed as networks generate an emergent landscape filled with attractors that represent the stable behaviors of those systems. How does such a system change its behavior? The landscape metaphor is expressed in language evoking geographical terrain in two converse forms either visualizing the landscape's attractors as valleys (potential wells) or peaks (points of high fitness). If we use the language of peaks, a peak is a local optimum that "traps" an adaptive walk across a landscape (Kauffman, 1993) because small movements in any direction down off the peak mean lowering fitness. A fundamental puzzle arises: How does a system navigate a landscape from one peak (attractor) to another attractor (possibly with higher fitness) given that valleys of low fitness act as barriers to movement?

The two preeminent theories for how a system (successfully) evolves are mutation and crossover recombination, including sexual reproduction, both of which have serious disadvantages. Mutation is a random leap off the peak that guarantees nothing about the fitness of the landing point; indeed the unknown result of this leap is nontrivial because fitness peaks are hard to find (Skellet *et al*., 2005). Crossover recombination tends to become trapped at local optima because crossover recombination converges on a local fitness maximum and thus is not an ideal way to escape from a local maximum upon which a system is residing, since, if it moves off the maximum it is likely to return to the same maximum.

Kauffman (1993) points out that mutation and crossover recombination acting together can overcome the limitations of each alone. But assuming that organizations do not intentionally choose the random leap of mutation they are left with recombination and must find some way of splicing their "genetic code" by mergers, consolidations of work groups, and so on. Indeed organizations can become stuck at points that are near but not even on a local maximum (Rivkin & Siggelkow, 2002). The stakes are high and leaps off a local fitness maximum can be fearful; organizations are inclined to stay on or close to their current peak.

We will offer a proof of concept for a new method of navigating landscapes. What if it were possible to use information from the current peak to specify what changes in the current behavioral pattern need to be made to arrive directly at another peak? This is possible if peaks fall into symmetry groups. Mulvey, Amazeen, and Riley (2005: 308) note that symmetry relations among symmetry groups code information "that otherwise would need to be specified individually." Thus, our alternative method of navigating landscapes in search of peaks will be based in symmetry group theory. But before we can demonstrate how symmetry group theory applies to adaptive landscapes we must first derive a discrete phase portrait for Boolean systems so that we can analyze the dynamics on a landscape.

nk Boolean Methodology

N*k* Boolean networks (described in detail by Kauffman 1993: 188ff; Kauffman, 1995: 75ff; and Malloy *et al*., 2005) consist of an arbitrary number *n* of abstract entities called nodes. The nodes have two states: ON (state = 1) and OFF (state = 0). Each node takes input (0 or 1) from *k* nodes. Time flows in discrete iterations. On iteration *t*, each node, uses the input (0 or 1) it has received from its *k* input nodes and, based on a Boolean (logical) truth table, calculates what its own value will be (either 0 or 1) on the next iteration, *t*+1. That is, the nodes are coupled; the output of one is input to others and visa versa. *nk* Boolean networks are quite general and can thus be wired to produce cellular automata (where the network structure follows a direct spatial representation) and to produce limited forms of neural networks (excluding, for example, those that allow for continuous underlying values for nodes).

Table 1 fully specifies a randomly generated *n* (nodes) =4, *k* (inputs per node) =2 Boolean system which we will call "System 1" as a simple example. The four nodes are arbitrarily labeled *A* through *D*. Under each of the Nodes, *A* to *D*, in Table 1 is a sub-table specifying the truth table that the particular node uses to generate its future output from current input. For example, in Table 1 note that Node *A* takes input from Nodes *B* and *C* at time *t*; then, based on the states of *B* and *C* at time *t* (leftmost two columns under Node *A*), *A* will either generate a 0 value or a 1 value at time *t*+1 (third column under Node *A*). To continue the example, if *B*=0 and *C*=1 at time *t*, *A*=1 at time *t*+1.

As explained in greater detail in Malloy *et al*. (2005) the node sub-tables in Table 1 each specify a logical operator that relates a node's two inputs at time *t* and calculates that node's value at time *t*+1. The column vector for each node for *t*+1 is called its truth vector and is indicated in bold. It is the truth vector that defines a node's relationship to its inputs. In the case of Node *A* in Table 1 the logical relation is the XOR operator that detects difference. Notice in Table 1 if the two inputs to *A* are the same (both 0 or both 1) at *t*, *A* will be 0 at *t*+1; in contrast if the two inputs are different at t, *A* will be 1 at *t*+1. Node *C* also uses XOR as a relation between inputs at *t* to determine its state at *t*+1. In contrast Node *B* uses an operator that ignores its input from *A* and takes on the same value at *t*+1 as *D* had at *t*. Similarly Node *D* ignores its input from *B* and mimics its input from *C*. There are sixteen possible logical operators (truth vectors) for $k = 2$ inputs and the possibilities increase exponentially as k increases

Malloy *et al*. (2005) have built an open code *nk* Boolean simulation program written in Java and named E42. System 1 was generated pseudo-randomly by E42. That is, the two inputs to each node and logical operators for each node were determined by a randomization algorithm. The network size, density, inputs and operators can also be manually specified and the resulting networks can be examined with a series of traditional and specialized visual interfaces. All of the methods discussed here are currently accessible in E42 publicly.

Node A			**Node B**			**Node C**			**Node D**		
B at *t*	C at *t*	**A at *t*+1**	D at *t*	A at *t*	**B at *t*+1**	D at *t*	A at *t*	**C *t*+1**	B at *t*	C at *t*	**D *t*+1**
0	0	**0**	0	0	**0**	0	0	**0**	0	0	**0**
0	1	**1**	0	1	**0**	0	1	**1**	0	1	**1**
1	0	**1**	1	0	**1**	1	0	**1**	1	0	**0**
1	1	**0**	1	1	**1**	1	1	**0**	1	1	**1**

Table 1. *Logical relations among the nodes of* System 1, *a randomly-generated n=4, k=2 Boolean system. Each of the four nodes takes input from two other nodes. The truth vectors for each node are indicated in bold (below t+1).*

State Vectors and Basins of Attraction

The on/off state (Boolean values) of all the nodes can be described at any moment in time (i.e., on any iteration) by a *state vector*. The state vector is ordered from the first to the last node (in this case from Node *A* to Node *D*). Because Malloy *et al*. (2005) describe in detail the derivation of a Boolean landscape, we will skip some of the steps for the formal derivations of a landscape for System 1. But for the reader's convenience Table 2 specifies all of the state vectors of System 1 (left hand four columns) as well as the deterministic transition of each state vector at *t* to a state vector at *t*+1.

The logical relations among nodes in Table 1 determine the transitions from *t* to *t*+1 in Table 2 (see Malloy *et al*., 2005). This table defines the state space of this network in its entirety. Unfortunately, this representation of state space fails to show the inherent order that emerges from the Boolean network (this will become clearer as we show that this example generates a landscape of attractor cycles). Also, it is reasonable to create this state space for small systems such as this one, but loses its inherent value when *n* is large (too many possibilities). Even though we do not derive, step by step, the state transitions in Table 2, Table 1 gives enough information to do so.

Time *t*	>>	**Time *t*+1**
0 0 0 0	>>	0 0 0 0
0 0 0 1	>>	0 1 1 0
0 0 1 0	>>	1 0 0 1
0 0 1 1	>>	1 1 1 1
0 1 0 0	>>	1 0 0 0
0 1 0 1	>>	1 1 1 0
0 1 1 0	>>	0 0 0 1
0 1 1 1	>>	0 1 1 1
1 0 0 0	>>	0 0 1 0
1 0 0 1	>>	0 1 0 0
1 0 1 0	>>	1 0 1 1
1 0 1 1	>>	1 1 0 1
1 1 0 0	>>	1 0 1 0
1 1 0 1	>>	1 1 0 0
1 1 1 0	>>	0 0 1 1
1 1 1 1	>>	0 1 0 1

Table 2 *State Transition Table. All possible 16 state row vectors are shown in Time t column; this constitutes the state space. The vectors shown under t+1 are derived from the logic of Table 1 and show the transition of every vector in the state space into some vector in the state space.*

The Emergence of a Boolean Landscape

The emergence of a landscape is important for our arguments and we will detail this process by using the transitions between state vectors in Table 2 to find basins and attractors that comprise a landscape. Suppose on iteration *t*=1 the ON-OFF pattern for the four nodes from *A* to *D* is OFF, OFF, ON, OFF. This can be written as a state vector: **S**(1) ={0010}. Using Table 2 we can look up the state vector {0010} at time *t* and see that it becomes {1001} at time *t*+1. Following that procedure we derive the deterministic sequence from one state vector to another: **S**(1) = {0010} → **S**(2) = {1001} → **S**(3) = {0100} → **S**(4) = {1000} → **S**(5) = {0010} → **S**(6) = {1001} → **S**(7) = {0100} → etc… This sequence therefore describes the flow of Boolean values (or, alternatively, the behavior) of the system across time. We can do this for any initial state vector. For written text it is convenient to write state vectors as row vectors as we have just done but in the figures and tables below we will transpose them so they are column vectors; this places nodes on the vertical axis and time on its traditional, horizontal, axis.

Because the system is deterministic, a given state vector **S**(*t*) will always be followed by the same state vector **S**(*t*+1). Therefore the fact that in the above example **S**(5) is identical to **S**(1) means that **S**(6) must be identical **S**(2) and **S**(7) must be identical to **S**(3), and so on. This indicates that the system is in an attractor cycle of four distinct state vectors, **S**(1) through **S**(4), repeating endlessly. This is the attractor cycle (which we will label A3 on the landscape shown in Figure 1, below.

The length of an attractor cycle, measured in the number of state vectors (or iterations) before it repeats, is the fundamental frequency or length, *l*, of the cycle. Due to complete determinism in the system, it cannot escape this attractor cycle unless the system is perturbed. (Perturbation amounts to changing the state of one or more nodes.) The important points here are that the nodes of *nk* Boolean dynamic systems flow from state to state by a deterministic relational logic, that at any moment the entire system can be characterized by a state vector, and that the flow from state vector to state vector across time can fall into cyclic attractor cycles within basins of attraction. All of these basins from the same underlying network comprise a landscape.

Attractor Matrix

An attractor can be coded as a matrix of Boolean values (0s and 1s); the *n* nodes of the system are arrayed along the vertical dimension of the matrix (we place the first node, *A* in System 1, at the top of the vertical dimension of the matrix and the last node, *D*, at the bottom) while the horizontal dimension of the matrix counts the discrete units of time (iterations) required by the length, *l*, of the attractor. The attractor matrix is therefore *n* x *l* (e.g., the attractor we are discussing is a 4 x 4 matrix—four nodes that have four unique state vectors across time).

Since the attractor cycle will repeat without limit, this matrix fully describes the possible state vectors seen at all future iterations. Our convention in writing an attractor matrix is to begin with the state vector (column vector) that has the lowest binary value (defined by starting with the value of the first node in the system which is at the top of the columnar state vectors and reading down to the value of the last node). Starting an attractor matrix with the lowest valued vector has many virtues including the ease of finding and identifying a specific attractor in a database of attractor matrices. We call this process rotating the attractor matrix to its normalized form.

Full Landscape

A Boolean landscape summarizes the activity that emerges from the Boolean network. If we follow the deterministic path for vectors in the state space of Table 2 we will eventually derive the landscape of System 1 shown in Figure 1. That is, Table 2 provides the reader with the full information for deriving the landscape in Figure 1. In Holland's (1998) terms, the node truth tables in Table 1 are the constrained generating procedures from which this landscape emerges. Notice that the dynamics of this small Boolean system fall into six basins of attraction; each basin has an attractor cycle—a series of state vectors that loop (as they did in the example above where **S**(5) was identical to **S**(1)). State vectors can also flow into attractors but not be a part of the looping vectors that define an attractor, these are known as *tributaries* (see Malloy *et al.*, 2005 for an example with tributaries). Tributaries are highly related to transience within continuous systems in that they are temporary as a function of the initial state vector or as a function of some perturbation to the system.

System 1 is unusual in that, in the general case, a Boolean landscape is comprised of basins that have both tributaries and attractors. In the case of System 1, the Attractors, A1 through A6, are isomorphic with the basins because there are no tributary state vectors. While unusual in not having tributaries, System 1 is an excellent example for the conceptual points we want to develop here. We will use A1, A2, A3,… as symbols for the attractors in Basin 1, Basin 2, Basin 3,… The basin numbering system in E42 is rational in that E42 archives basins first rotating each attractor matrix so that its first column has the lowest binary value among the state vectors in the attractor and then, using the values of the first column vector, naming them with the counting numbers (A1, A2,…) based on the binary value of each attractors first column. Thus notice that the Attractor (A1) for Basin 1 in Figure 1 begins with the lowest Boolean value of vectors in the state space, A2 begins with the second lowest Boolean value, etc.

Basins 3, 4 and 6 have attractor cycles (A3, A4, A6) that repeat every four iterations, that is the cycle length is $l = 4$. Basins 1 and 5 have cycle lengths of $l = 1$. Basin 2 has an attractor, A2, with $l = 2$. In which basin does a system currently reside? That depends on the initial state vector in which the system starts and on perturbations to the system. We will return to the issue of how to traverse a landscape within the *nk* Boolean system.

Visualizing and Simplifying Boolean System Dynamics

We now switch focus from the methodology of how *nk* Boolean systems generate an emergent landscape to the ways under which we can visualize the dy-

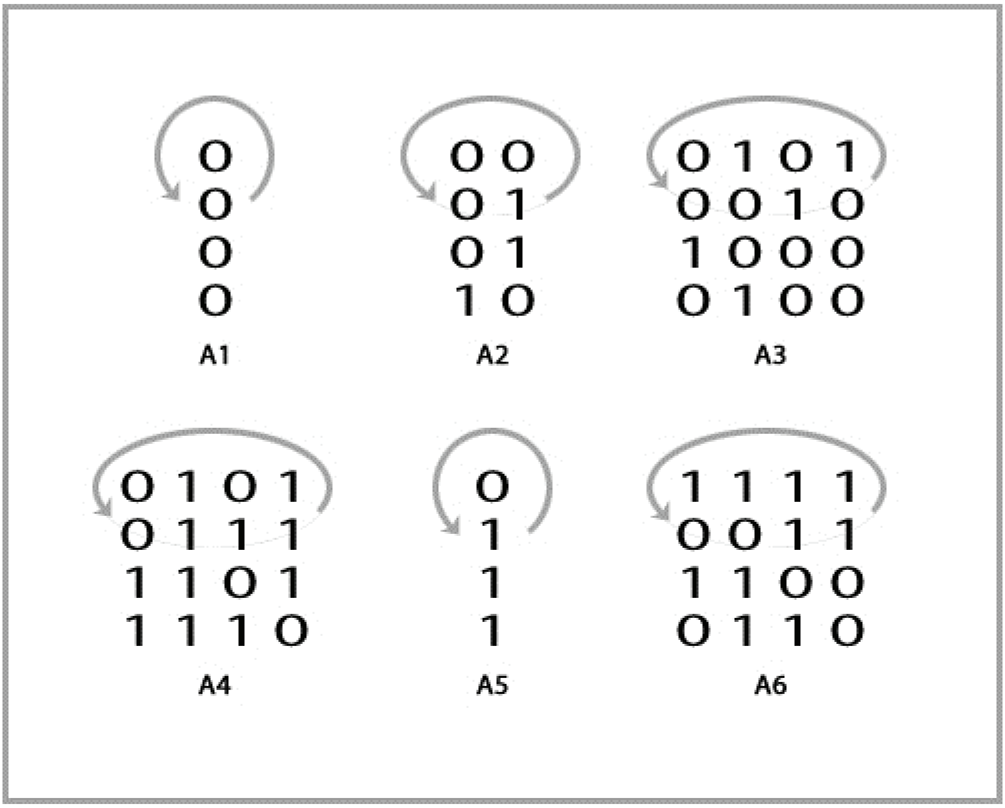


Figure 1 *Attractor Landscape for System 1. The basin landscape of a small 4 node system named System 1 expressed as column vectors. The landscape includes six basins, each with an attractor cycle. Each attractor is coded as a matrix of zeros and ones; the vertical dimension of the matrix contains the nodes of a system and the horizontal axis is discrete units of time (iterations). In general, a landscape has tributaries (state vectors) that run into attractors but are not part of the attractor loop; System 1 is unusual in that all of its state vectors exist in attractors and none of them are tributaries. All attractor cycles have been normalized, that is, rotated to begin with the state vector that has the lowest Boolean value.*

namics of the system. The advantage to the methodology laid out so far is that we can fully determine the landscape of any system we generate—though they quickly become quite large and complex (the example shown here is intentionally small and simple, though the principles hold for the more complex case). The goal of visualizing the underlying dynamics is to generate a simplification. We will first generate a visual representation of the state vectors in time, called a historical trace. This will then be combined with the discrete form of derivatives, seeking a simplification of the pattern. Finally, we will seek a visual representation of the relationship between the discrete derivatives, yielding the phase portrait equivalent.

Historical Trace

As noted earlier, the columns of 0s and 1s in Figure 1 can be construed as *n* x *l* attractor matrices. To visualize these state vectors in a perceptually useful way we place these matrices on a grid with 0s represented as white cells and 1s represented as black cells. The vertical axis will then represent individual nodes from 1 (on top) to *n* (on the bottom) and the horizontal axis will represent iterations from 1 to *l*. Figure 2 shows the A3, A4, A6 attractor matrices from Figure 1 represented in this way. We call this a historical trace because it shows the pattern of states vectors of the system over some period of time (four iterations in Figure 2).

Different attractor cycles generate different visual patterns; Figure 2a shows the form generated by the A3 attractor cycle, Figure 2b shows A4, and Figure 2c shows A6. A fuller discussion of the historical trace can be found in Malloy *et al.* (2005). For the methodology we are presenting it is important that we be able to see, with perceptual immediacy, the patterns found both in the system's dynamics and in the analyses of those dynamics we are about to develop. Because the historical traces are a recoding of the attractor matrix, like the attractor matrix we generally rotate the historical trace to normalized form.

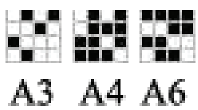


Figure 2 *Visualization of Column Vectors for Three Attractors. Visual patterns formed by the state column vectors of attractors A3, A4, and A6 from System 1. Nodes are on the horizontal axis and time (iteration) is on the horizontal axis. All attractors have been rotated to normal form.*

Derivative Simplification: TAO

As with continuous systems, we embrace derivatives as a way of simplifying the description of a system. We therefore now turn to the discrete derivative and its visual representation developed by Malloy *et al.* (2005) and Malloy and Jensen (2008) known as the TAO function. As noted in Table 1, XOR compares two inputs and if the states of the inputs are the same it returns a 0 and if the states of the inputs are different it returns a 1. Therefore we say that the XOR operator detects difference. Since the definition of a derivative is change in time, we apply the XOR operator to compare two successive state vectors to determine if there was change or no change.

To understand TAO, consider two state vectors, **S**(1) and **S**(2), from the discussion of A3 above (see text and Figure 1) and which are shown as column vectors in Table 3; these column vectors represent the states of the four nodes at t =1 and at t = 2. The top value in a vector represents the state of Node *A*, the second value represents the state of Node *B*, and so on. The reader merely has to compare the Boolean values, node by node, moving down **S**(1) and **S**(2) confirming that a 0 appears in the TAO column when the values for a node are the same across time and 1 when the values for a node are different across time. Because it tracks the absolute change (or lack of change) in the flow of Boolean values across time TAO is the basis for the discrete analogue to the first derivative (The difference between TAO and XOR is that TAO is a matrix operator that compares, node by node, one column vector in the matrix with a succeeding vector in time). Since all of the TAO values are also merely 0 (same) or 1 (different), it can also be visualized as white and black cells in place of zeros and ones, having no effect on the logic.

Node	S(1)	S(2)	TAO
A	0	1	1
B	0	0	0
C	1	0	1
D	0	1	1

Table 3 *Two state column vectors from the A3 attractor cycle with the resulting TAO column vector indicating change versus no-change for each node from T1 to T2.*

We find it useful to label the attractor matrix TAO-0; this indicates a null application of the TAO operator to the attractor's dynamics. Since TAO-0 is a matrix representation of the state vectors, in temporal order, the first derivative can also be expressed in matrix form. To complete the logic of TAO, examine the relation between the TAO-0 (attractor matrix) pattern and the TAO-1 pattern in Figure 3. Begin with Node *A* in the TAO-0 matrix; notice that Node *A* changes from white to black between iteration 1 and iteration 2. When a node changes its value between two successive iterations the XOR operator indicates this difference by returning a 1 (black cell). Now notice in the top (Node *A*) row of the TAO-1 matrix that the first column is black; this black cell corresponds to the change in the value of Node *A* between iteration 1 and 2.

In fact, notice in the TAO-0 attractor matrix that Node *A* changes from black to white between each successive pair of iterations; therefore the first row of the TAO-1 matrix is all black indicating that Node *A* changed on each iteration. To what do we compare the last column in the attractor matrix? To state the obvious, the attractor is a cycle that repeats its first value after its last value no matter where we start in the cycle; therefore the fourth column in the TAO-0 matrix is compared with the first column.

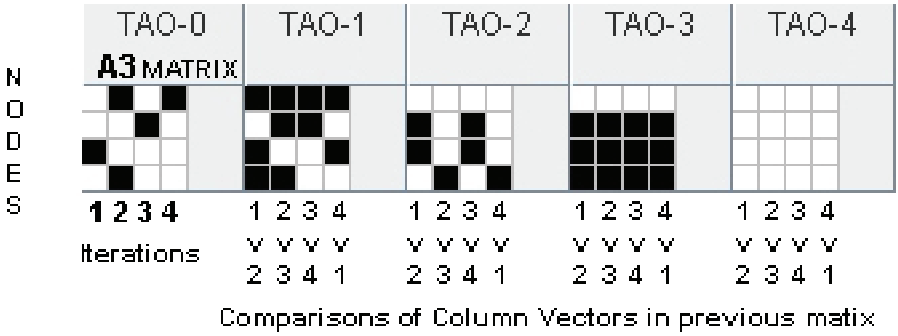


Figure 3 *Visualization of Attractor A3 and its first four derivatives. The attractor matrix (TAO-0) and first four derivatives (TAO-1 through TAO=4) of A3. A white cell = 0 and a black cell =1 in Boolean values. In TAO-0 the vertical matrix dimension indexes the four nodes (A at the top, D at the bottom) and the horizontal dimension indexes time (iterations). In the TAO-1 through 4 matrices, the horizontal dimension indicates the results of an XOR comparison between two columns (e.g., 1 v 2 or 2 v 3) in the previous (to the left) matrix. A3 has been normalized but the TAO's have not been rotated so that the calculation of each TAO is easier to follow.*

Now examine Node *B***,** which changes twice during the attractor cycle (from the second iteration to the third and from the third iteration to the fourth). Thus the Node *B* row in the TAO-1 matrix is white, black, black, white or {0110}. Using this logic, the reader can also examine and confirm the TAO-1 values for Nodes *C* and *D*. Recall that the attractor cycle is a flow of changes in Boolean values. Therefore the full TAO-1 matrix codes all the moment-to-moment changes in the flow of changes in the attractor cycle. Like velocity codes the momentary changes in position, TAO-1 codes the momentary changes in Boolean values in an attractor's cyclic dynamics.

Recursive TAO

The TAO function can be applied (like the derivative function) to its own output. TAO-2 simply takes the first column vector of TAO-1 and compares it, node by node, to the second column vector of TAO-1 yielding 1 (black) for a change and a 0 (white) for no change. TAO-3 applies TAO to TAO-2 and TAO-4 applies TAO to TAO-3. This analysis should be easy to follow in Figure 3.

In Figure 3 the derivative diminishes to the **0** matrix (all cells white). As reported in the Malloy & Jensen (2008) this always happens when the attractor cycle length, *l*, is a power of 2. In the case of A3 (*l* = 4), since 4 is a power of 2 the derivative function diminishes to no change in change by the fourth derivative. As a side note, when *l* is not equal to a power of 2, rather than the higher order derivative matrices resolving to **0**, the derivatives begin repeating themselves in long, complex patterns.

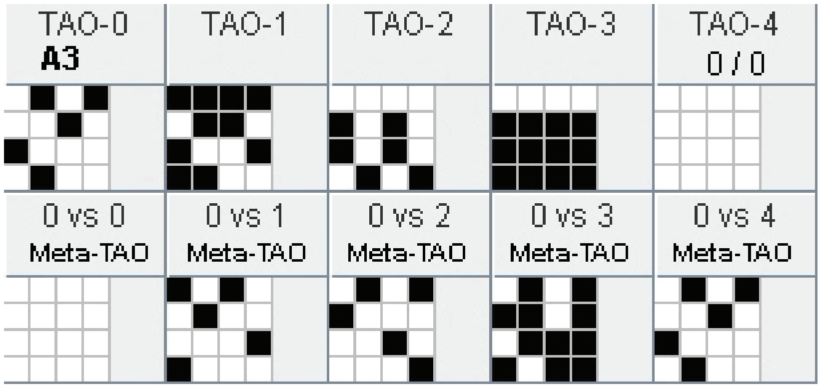
Meta-TAO: The Boolean Phase Portrait

Systems theory has long been deeply entwined with the idea of a velocity flow-field in the form of a phase portrait, phase space, or state space. Going all the way back to Poincaré's solution to the three body problem, these topographical representations of change give insight into the simplicity of some patterns and the complexity of others. Specifically, they generate visually identifiable and measurable properties that define many of the emergent processes that might otherwise remain implicit in the behavior of systems. Attraction, repulsion, attractor cycles, separatrices, basins, and even Lyapunov exponents all have visual definitions in the flow-field. Despite this strong, interdisciplinary interest in landscape dynamics ever since Wright (1931) first proposed the idea and Kauffman (1969) formalized Boolean networks, Boolean landscapes have lacked, to our knowledge, this fundamental analysis tool of dynamical systems.

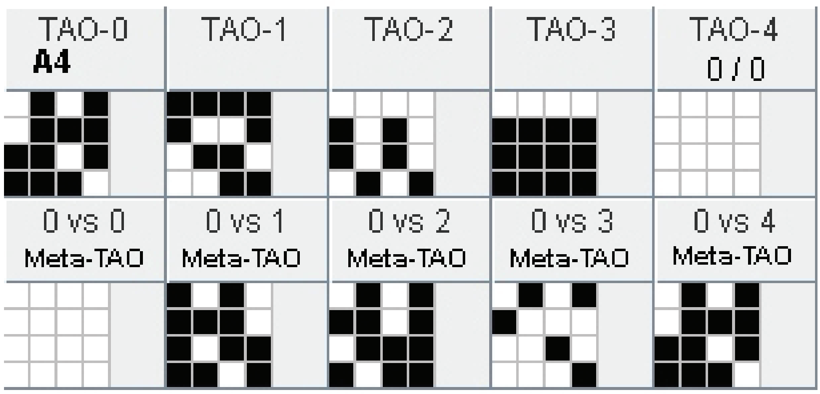
Through the TAO matrices, in historical trace form, we capture one half of the velocity flow field—it represents velocity (and higher order derivatives). By parsing the TAO matrices we approach a direct analog to the phase portrait. To begin our construction of an operation that describes the relationship among Boolean derivatives, we define a matrix operation whereby every corresponding cell in two conformable matrices (containing binary values) is compared (by the XOR operator); if the two cells have the same value then XOR returns a 0 (or, visually, a white cell) and if the two corresponding cells are different XOR returns a 1 (black cell). To construct a Boolean phase portrait the two matrices so compared are always two TAO matrices (usually including TAO-0 as one of the pair); thus we call the operation Meta-TAO. In terms of the Boolean logic of the systems we are analyzing, Meta-TAO can also be thought of as a way of assessing the Boolean relationship between two TAO matrices akin to a phase space.

Figure 4 has three panels. The upper row of Panel (a) shows the A3 matrix along with its four derivatives, TAO-1 through TAO-4. Note that the A3 matrix is also labeled TAO-0. The lower row of Panel (a) shows Meta-TAO matrices visualizing the Boolean relationship between the attractor matrix (TAO-0) for A3 and each of its derivatives. Panels (b) and (c) show the same information for attractors A4 and A6. The 0 v 0 Meta-TAO comparison in all three panels is an all white matrix because it is comparing TAO-0 with itself and so there are no differences.

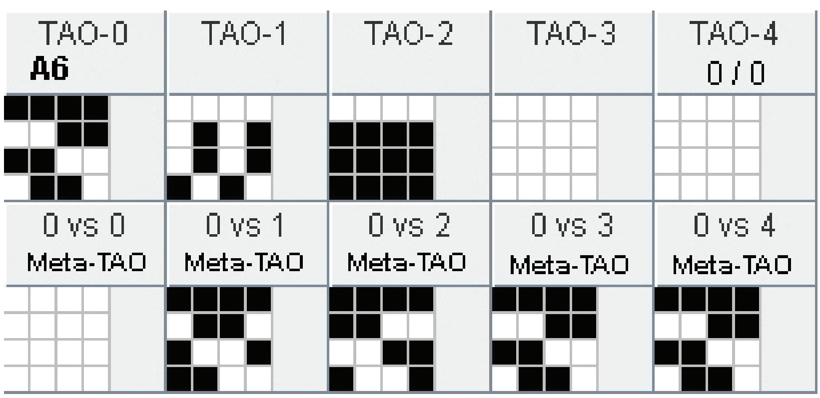
A system's phase portrait is "the key to the geometric theory of dynamical systems," (Abraham & Shaw, 1992: 11) and is typically plotted in classical oscillator theory (Abraham & Shaw, 1992: 64) as a first derivative (velocity) against zero-order derivative (position) and portrays the trajectories of a system. It then maps the relationships amongst basins: where they are in relation to one an-



(a) TAO and Meta-TAO (lower row) matrices for A3



(b) TAO and Meta-TAO (lower row) matrices of A4



(c) TAO and Meta-TAO (lower row) matrices for A6

Figure 4 *TAO and Meta-TAO analyses for three attractors. Complete TAO and Meta-TAO analyses for attractor cycles A(3), A(4) and A(6) in panels (a), (b), and (c) respectively. In each panel the upper row of patterns are the TAO derivatives from TAO-0 to TAO-4. The lower row of pattern areMeta-TAOs comparing TAO-0 to itself (0 v 0), to TAO-1 (0 v 1), to TAO-2 (0 v 2), and so on from left to right.*

other. As mentioned already, the basins themselves are easily identifiable as the stable attractor cycles and their constituent tributaries. However, Meta-TAO provides the relations amongst the basins, just like the phase portrait.

To illustrate this point we wish to draw your attention to the non-identity Meta-TAO (0 v 3) matrix for A3 in Figure 4a; to review, this matrix codes the Boolean XOR relation between A3s TAO-0 matrix and its TAO-3 matrix, that is, between the zero-order derivative and the third-order derivative. An important aside is that theMeta-TAOs in Figure 4 have not been rotated to start with the lowest Boolean value. What is important to notice is that the Meta-TAO (0v3) matrix for A3 (if you rotate it in your mind to begin with the lowest binary column vector value) is exactly the A4 attractor matrix (see Figure 2 or Figure 4b). In other words, the Meta-TAO-3 of A3 transforms A3 into A4. This establishes a deep relation between the attractors of two different basins (A3, A4) from the same landscape.

Since it is an analogue to a phase portrait, the ability of Meta-TAO to visually map relations among attractors is not surprising since that is one of the most valued characteristics of phase portraits (e.g., Abraham & Shaw, 1992: 64). What is surprising is that a phase portrait analysis, in this case Meta-TAO-3, transforms one attractor into another; at least we were surprised when we first came across this finding. Note that in contrast, Figure 4c shows that the Meta-TAO (0 v 3) matrix for A6 yields A6 itself; that is, Meta-TAO (0v3) for A6 is an identity Meta-TAO. Thus A3 and A4 are related to each other by a Meta-TAO analysis while A6 is related to itself. Taken together, these relations form a map of how each basin is related to one another, spatially depicting the stable patterns of change. As we will now argue, this relationship between the Meta-TAOs exists because the Meta-TAO calculation can be thought of as a transformation of the matrices—the very transformation that defines symmetry and group theory. These observations foreshadow our proposed new evolutionary strategy which now will describe in detail.

Symmetry As A Basis For Systemic Evolution

Evoking a famous poem by Blake, modern theorist such as Stewart and Golubitsky (1992) and Zee,(1986) have titled their books *Fearful Symmetry*. We argue that fearful symmetry is critically involved in the emergence of a landscape from the network of processes that generate the landscape (although we will not develop our full argument in this paper). Symmetries in the structure of the landscape will be expressed in symmetries in the dynamics of the emergent landscape. And, for the essential point of this paper, the fearful symmetry of a landscape can be a basis for fearless movement from one attractor on the landscape to another.

Symmetry Groups

Any transformation that leaves an object apparently unchanged is a symmetric transformation (Stewart & Golubitsky, 1992: 28). For the transformation to be nontrivial something must have changed but to speak of symmetry the object in some sense must appear the same before and after the transformation. An infinitely long horizontal line can be translated to the right and it will appear identical; it is said to have translational symmetry. An equilateral triangle whose upper vertex points at 12 o'clock can be rotated 120 degrees about its center and appear identical. It has rotational symmetry through 120 degrees. But the triangle does not have rotational symmetry through, for example, twenty-seven degrees. Assuming we have a frame of reference, an equilateral triangle rotated twenty-seven degrees will appear different before than after a twenty-seven degree rotation.

Guggenhemier (1977) summarizes Felix Klein's Erlanger program characterizing various geometries within group theory. More recently Stewart and Golubitsky (1992) describe the relation between symmetry and groups. A group is a closed set of transforms such that when any two transforms from this set are combined they produce the results of another transform in the set. Suppose with our equilateral triangle we define the closed set of six transforms as clockwise or counterclockwise rotations through 0, 120, and 240 degrees. Combing any two of these six transforms produces one of three triangles that are identical in appearance; no matter how many of the transforms are applied, in whatever order, the application of the transforms will result in one of these three triangles. The three triangles, identical in appearance, comprise a symmetry group defined by the six transformations. (The vertices of the triangles in this symmetry group point at 12, 4, and 8 o'clock).

The Curie Principle

Stewart and Golubitsky (1992) also describe the Curie Principle which proposes that there is a meta-symmetry across symmetries found in physical causes and effects. We will not discuss this principle in full here but a key aspect of the Curie Principle is that the symmetries of causes reappear in their effects. In terms of emergence, symmetries found in the lower order generating processes will reappear in some form in the higher order emergent processes.

Emergent Landscapes and the Curie Principle

Our conjecture, based on the Curie principle, is simple: Symmetries in an *nk* Boolean network generate symmetries in the emergent landscape such that basins of attraction occur in symmetry groups. We propose that attractors are related to each other by the transforms (in this case TAOs and Meta-TAOs) that define symmetry groups and that the Boolean basin landscape is not a messy, random sort of thing; rather, it can be elegantly described in terms of symmetry groups and their relations.

At any given moment the non-chaotic Boolean system will and must deterministically fall into a single basin of attraction and cycle within that basin's attractor. The specific attractor the system ends up in depends on initial conditions and subsequent perturbations of the system. If these attractors come in symmetry groups then taking the current attractor and applying appropriate transformations can precisely describe other attractors. Thus local information has global implications in terms of movement across a landscape and a search for other attractors. As we noted above, Mulvey, Amazeen, and Riley (2005, p. 308) argue that symmetry relations among symmetry groups code information "that otherwise would need to be specified individually." Quite simply, we propose that an attractor cycle in a Boolean landscape, when symmetry group relations exist, codes information about other possible attractor cycles.

Start with Symmetry

Our initial strategy is to create networks with sufficient symmetry to allow an examination of the Curie Principle; we start with Boolean XOR rings whose infrastructure (nodes, connections, logical operators) have a high degree of symmetry and whose effect (a Boolean landscape) should therefore have a high degree of symmetry. In this highly symmetrical context we can examine how the symmetries of the landscape are exhibited, how they are broken, and how they form symmetry groups. It is worth noting that the ring structure is not without wide application. Winter (2001) argues that rings are important and common in biology, particularly where biological rhythms and clocks are involved. He also points out that some bacteria and viruses have circular genomes.

The Symmetry of Boolean XOR Rings

XOR Rings

An *nk* Boolean system is defined in terms of three critical variables:

1. The number, *n*, of its nodes;
2. The connections by which a node receives input from other nodes (and sometimes from itself), *k*, and;
3. The logical operators which determine if a node is ON or OFF iteration on $t+1$ (based on the states of its inputs at *t*).

Self-Referencing Nodes

In an *nk* Boolean system the connections among nodes refer to how the nodes take and give input. Obviously, *k*, the number of inputs per node is important but there are other aspects of the connections that will influence the symmetry of the attractor cycles that emerge when the system runs. Kauffman (1993) explored connecting nodes pseudo-randomly as did Malloy *et al*. (2005) and Malloy *et al*. (2005). But nodes can be connected any way. One potent variable is whether a node is connected to itself; that is, whether it examines its own state at time t (in relation to the states of other inputs at time t) to determine its state at time $t + 1$. We call a node that examines its own state at time t a "self-referencing" node.

Rings

Figure 5 shows another aspect of the connections among nodes—the structure of the net within which the nodes reside. If the nodes are not connected pseudo-randomly then they must be connected in some systematic way. One way to connect nodes is to make a ring. In the case shown in Figure 5 the seven nodes take $k = 2$ inputs each. They are all self-referencing nodes so one of the $k = 2$ inputs is each node from itself; that leaves only one external connection by which a node receives input. In Figure 5, this single external input is a node's nearest clockwise neighbor. Thus the structure of the net is a ring.

XOR Operator

Throughout the paper, we have utilized the XOR operator to detect difference. It thus is also ideal for providing symmetry in the network structure via the truth tables. By this we mean that for the ring structure in Figure 5 each self-referencing node will compare (at time t) its own state with its neighbors state; if its own state is the same as its neighbor's state (both ON or both OFF) then the node will be OFF at time $t = 1$. Conversely if a node's own state at t is different than its neighbor's, it will be ON at $t + 1$. Wolfram (2002: 25, rule 90) has established that the XOR operator produces an affine Sierpinski gasket in cellular automata, a result we have generalized to Boolean networks (Cooper *et al*., 2007, July). The important point is that the Sierpinski gasket, like all fractals, is imbued with symmetry.

For XOR Rings we will examine the XOR operator slightly differently than we have in the past when we used it as a basis for Boolean derivatives. Here our interest is in the process of emergence: Given that XOR is an operator that generates a known kind of symmetry (Sierpinski gasket) what happens to that symmetry in the Boolean landscape that emerges from a small system whose nodes all use XOR as their relational operator? We will look for symmetry in the landscape of basins that emerges from such a system. Based on the work of Wolfram it is not hard to anticipate that the attractors will be related to affine Sierpinski fractals. Quite simply, the XOR operator makes a good starting point to explore the implications of the Curie Principle as a principle of emergence.

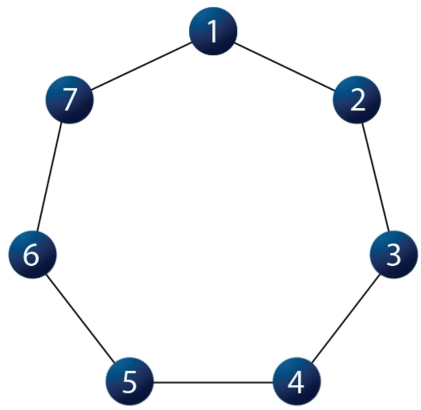


Figure 5 *An n=7, k =2 Boolean XOR ring. Each node is self-referencing so one of its k=2 inputs is from itself; its other input is from its clockwise neighbor. The operator for each node is the XOR operator. The XOR operator determines that a node will on ON at time t+1 only if a node's state differs from its neighbor's state at time t.*

Network Structure Generates Landscape

Malloy *et al*. (2005) argued that, computationally, the basin structure of Boolean system emerges from and is determined by the structural characteristics of the system. That is, the nodes, the connections among the nodes and the logical operators that nodes use constitute one level of analysis. The basins of attraction and the attractors in each basin are the behavior of the system and, as such, constitute another level of analysis. In terms of the Curie principle, the structure of the system (nodes, connections, and operators) causes the landscape of attractor basins. Thus we would expect to find symmetries in the effects (the basin landscape) when there is symmetry in the cause (XOR relations among nodes in a ring). Specifically, the highly symmetrical XOR Ring is expected to produce symmetry in the landscape of attractors. Most important, in terms of the Curie Principle we expect the attractors to come in groups related by symmetry.

Symmetry and Symmetry Breaking in the Basin Structure of an XOR Ring

Figure 6 shows the nine length-7 attractor cycles that emerge from the N7 XOR ring. The grid labeled A2 simply means Attractor 2, A3 is Attractor 3, and so on. In this way the attractor patterns, A2 to A10, shown in Figure 6 are visual representations of the Boolean dynamics of nine attractors in the landscape that emerges from the XOR Ring shown in Figure 5. This landscape has one other attractor, A1, that is not shown in Figure 6; A1 has a cycle length, *l*, equal to 1 and all seven nodes are OFF, that is they have a Boolean value of 0. If we were to visually represent A1 it would be a single column of WHITE cells (remember an attractor is characterized by an *n* x *l* matrix, thus for A1 where *l* = 1 the matrix would be 7 x 1).

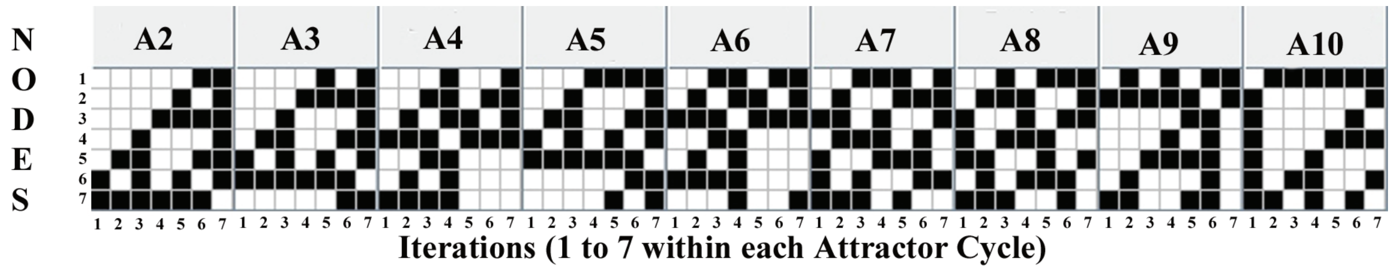


Figure 6 *The landscape of the n7 Self-referencing XOR Ring has 10 basins each with an attractor. The attractor cycles of nine of the ten basins have length l = 7 and are shown above, numbered A2 through A10. Iterations (1 to 7 fore each attractor) are on the horizontal axis; nodes are on the vertical axis. Basin 1 (not shown here) has an attractor of cycle length l = 1 with all nodes OFF (WHITE).*

In Figure 6 some parts of an affine Sierpinski pattern may be apparent to the reader in some of the attractor patterns. But neither the degree of symmetry nor how symmetry groups are defined is immediately obvious in Figure 6; the deeper forms of symmetry and of symmetry breaking and sharing require further analysis in the form of TAOs and Meta-TAOs.

To begin, we will look at the logic of the TAO and Meta-TAO operation using a single node as an example. Figure 7a, top row, shows A2 and its first eight derivatives (TAO-1 through TAO-8). Figure 7b, second row, shows eightMeta-TAOs (M1 through M8) of Attractor 2, each of which compares A2 (TAO-0) with one of its derivatives. As noted above, Meta-TAO does a Boolean comparison, cell by cell, of two conformable matrices; in Figure 7b allMeta-TAOs compare the A2 matrix with one of its derivatives. The first Meta-TAO (M0) in Figure 7b is all white because it is comparing the attractor (TAO-0) with itself; there are no differences, of course, between itself and itself.

The M1 matrix compares the TAO-1 matrix in the top row with the A2 matrix in the top row. Notice that this comparison, which we call Meta-TAO-1, or more compactly M1, is exactly the same as the original basin. Note that in Figure 7 we have normalized the Meta-TAO matrices (rotated them to begin with the lowest valued column vector) so that they can be perceptually compared with normalized attractor matrices. Thus if you calculate M1 by eye or by hand you will find that we have rotated M1 either1 iteration to the left or, equivalently, 6 iterations to the right to obtain this identity of pattern between M1 and A2. The rationale for normalizing matrices was discussed above. Continuing this logic, the third matrix (M2) in the second row of Figure 7 is Meta-TAO-2; it compares (in the top row) the second derivative with the original attractor. M2, when it is rotated either 2 iterations to the left or five iterations to the right, is identical to the A2 matrix. For the moment we skip M3 and go on to M4, which, when rotated, is also identical to the original attractor.

Identity and Non-Identity Meta-TAOs

Meta-TAOs 1, 2, 4 and 8 are identical, when rotated to normalized form, to the original attractor. Thus we can call them *identity* Meta-TAOs. In contrast, in Figure 7 we can see Meta-TAOs 3, 5, and 6 are different than the original attractor; a simple descriptive term for these is "non-identity" Meta-TAOs because whatever transformation, ***T***, of TAO-0 these Meta-TAOs represent they certainly do not generate the identical TAO-0 matrix no matter how they are rotated. The conceptual lynch pin for the rest of this paper is exploring ***T*** and determining how this transform relates to symmetry theory.

Meta-TAO-3

In Figure 7, notice particularly Meta-TAO-3 (M3); it is not identical to A2 as were Meta-TAO-1 and Meta-TAO-2 and Meta-TAO-4. Indeed, if you follow the arrow

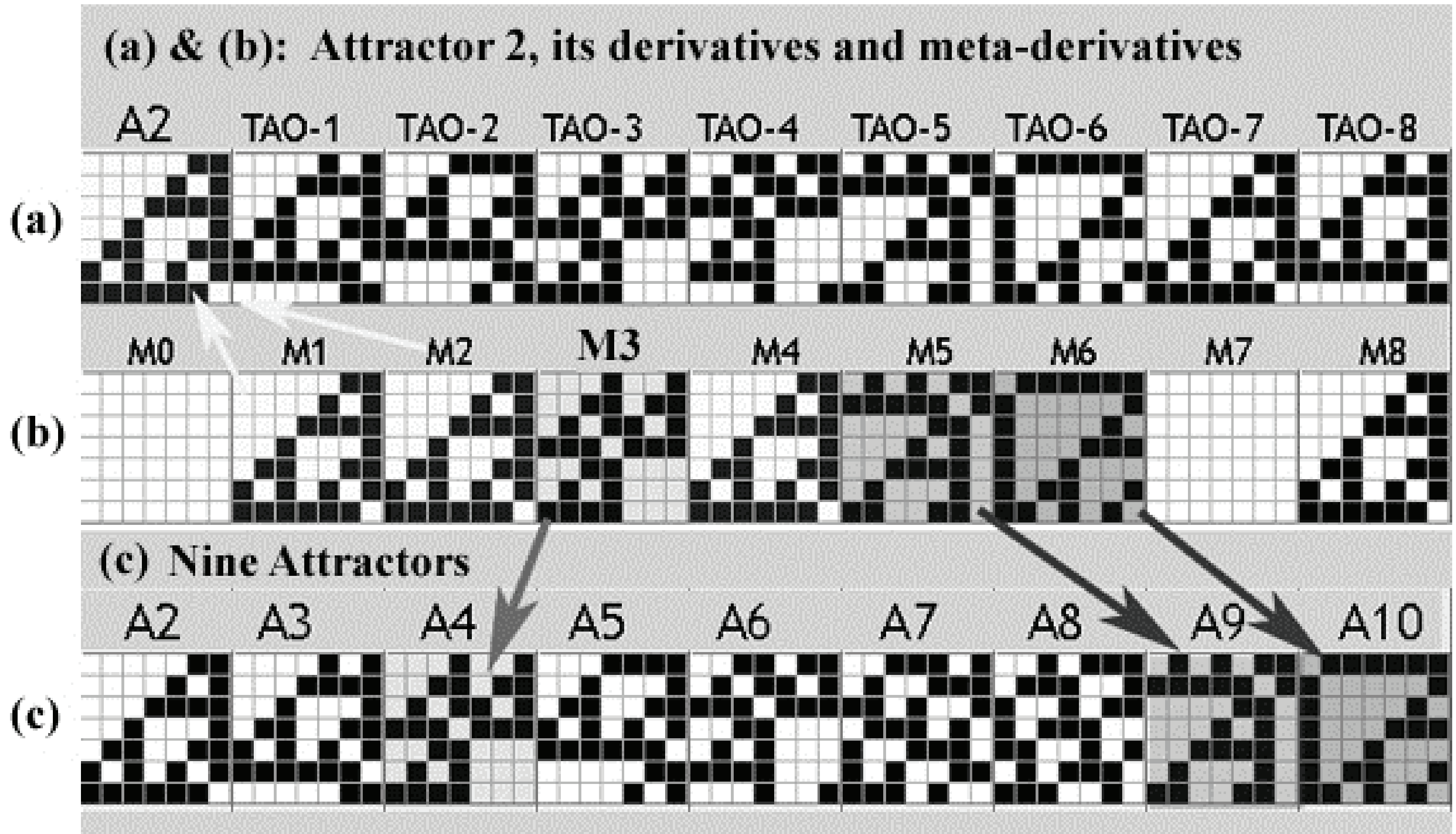


Figure 7 *Visualization of Symmetry Transforms. (a) Top Row: The derivatives (TAO's) from zero order to eighth order for Attractor 2. (b) Middle Row: Meta-TAOs comparing A2 to each of its derivatives. (c) Bottom Row: The nine length-7 attractors found in the XOR Ring landscape. In this figure all TAO's andMeta-TAOs have been rotated to begin with the lowest Boolean value where the lowest Boolean value is defined reading down the first column of each matrix.*

from M3 in Figure 7b to A4 in Figure 7c, you find that Meta-TAO-3 is identical to the attractor cycle for Basin 4 (A4) when both are rotated to normalized form. Even in the highly constructed and symmetrical case of an XOR Ring, this is provocative. We want to emphasize two things that happen when we use the Meta-TAO tool. First we have noted that certainMeta-TAOs (1, 2 and 4 in this case) are identity transforms: they produce the attractor cycle which is being analyzed—just like a transform that produces the identical pattern. Second, certainMeta-TAOs (3, 6, and 6 in this case) transform the current attractor cycle into a different attractor in the same landscape—just like a transform that defines the symmetry group. This second statement is important: Local analysis of local attractor dynamics yields the dynamic pattern of other attractors on the same landscape. Certain Meta-TAO matrices constitute a matrix transform, ***T***, by which one attractor becomes another. More succinctly, the Boolean Phase Portrait of one attractor's dynamics can, in some cases at least, yield another attractor on a Boolean landscape by capitalizing on the Boolean derivatives and phase portraits functioning as symmetry transformations.

We will discuss below more fully the limitations inherent in this highly symmetrical XOR Ring case. For the moment one detail requires comment. The reader may have noticed in Figure 7 that TAO-3 and Meta-TAO-3 are identical. Thus the Meta-TAO-3 analysis was not required to define the transform, ***T***, which changes the A2 matrix into the A4 matrix. Since TAO-3 and Meta-TAO-3 in this particular example are identical, ***T*** could have been defined by TAO-3, which is a prior and therefore simpler analysis.

In our ongoing work (Butner, Malloy, Cooper, & Smith, 2007) with less symmetrical Boolean node architecture, including, even, randomly generated Boolean systems, the key conceptual points in this paper hold up. Moreover, when we use less symmetrical Boolean systems to generate landscapes than our current XOR Ring (Butner *et al*., 2007) the Meta-TAO matrices will not be redundant with the TAO (derivative) matrices; rather it will be Meta-TAO matrices that act as the transform, ***T***, to find other attractors. Thus, since we are using this simple case to build a methodology, we focus on the non-identityMeta-TAOs (such as Meta-TAO-3) here as the critical analyses for defining symmetry groups (below).

Emergence of Symmetry Groups on a Boolean Landscape

We now have the requisite analyses for defining symmetry groups. Figure 8 shows crucial aspects of the symmetrical structure of the basin landscape of the $n = 7$ XOR Ring; this system produces ten basins, each with an attractor cycle and some number of tributaries. Only the attractors (numbered 1 through 10) are shown in Figure 8. Tributaries are not shown. A complete representation of the full basin landscape would also show the tributaries but for our current conceptual point the structure of the attractor cycle symmetry groups is sufficient.

Each attractor cycle is shown for seven iterations (horizontal axis). As usual, the nodes are shown on the vertical axis. Note also that M5 and M6 both also lead to an attractor; theseMeta-TAOs have the same properties as M3 and we could make the following arguments using either of them in place of M3.

Meta-TAO-3 as Symmetry Transform

Figure 8 summarizes symmetry relations among attractor cycles in the *n*7 XOR Ring based on the definition of the matrix operation:

$$\boldsymbol{T} = [\text{Meta-TAO-3}]$$

Recall that Meta-TAO-3 compares TAO-0 (the original attractor cycle) with its third derivative (TAO-3). We review the methodology: If we take the TAO-0 matrix of 0s and 1s (shown as black and white cells in our figures) and lay the third derivative matrix over it, then Meta-TAO-3 will yield us a matrix of 0s and 1s where the 1s indicate that for a particular cell TAO-0 was different than TAO-3.

The circle of attractor cycles at the bottom of Figure 8 represents a symmetry group that results from transforming each successive attractor matrix with its own Meta-TAO-3. Start with Attractor Cycle 2 at the top of the circle and operate on its NxL matrix with the Meta-TAO-3 matrix operation (**T**). *A*2 ***T* =** *A*4, where ***T*** is understood as operating on A2. Figure 7 already has shown the details of how operating on A2 with Meta-TAO-3 produces A4. Continuing this analysis, the Meta-TAO-3 transform applied to Attractor 4 is precisely Attractor 10, and so on, clockwise around the circle. Thus the Meta-TAO-3 transform defines an ordered set of attractors {2 $\Longrightarrow$ 4 $\Longrightarrow$ 10 $\Longrightarrow$ 5 $\Longrightarrow$ 9 $\Longrightarrow$ 3$\Longrightarrow$ 6 $\Longrightarrow$ 2 $\Longrightarrow$ etc.}. Defining ***T*** as M3 generates a symmetry group because repeated applications of the transform remain in the group.

Now notice in Figure 8 that attractors 7 and 8 each have a circle leading back to themselves; this indicates that, for both of them, Meta-TAO-3 generates the original attractor. This is also true for A1. Note that A1 is a 7x1 all-WHITE column vector; that is, it is a fixed point attractor. A1 is shown at the top of Figure 7 as if repeating across 7 iterations (a 7 x 7 all-WHITE matrix) to make its representation parallel to the other attractors. These three attractors, A1, A7, A8, each comprise a symmetry group in which they are the only element—for each, the Meta-TAO-3 transform stays in a one-element group.

In the end we have four symmetry groups defined by the Meta-TAO-3 transform. A1, A7, and A8 each define a group with a single attractor. Attractors 2, 4, 10, 5, 9, 3, 6, in that order, define a fourth symmetry group. Mulvey, Amazeen, and Riley (2005) have noted that transforms that define symmetry groups are highly compressed codes for the large amounts of information required to specify the members of the groups individually. That is, if you have purely local information from Attractor 2 and you can perform the Meta-TAO-3

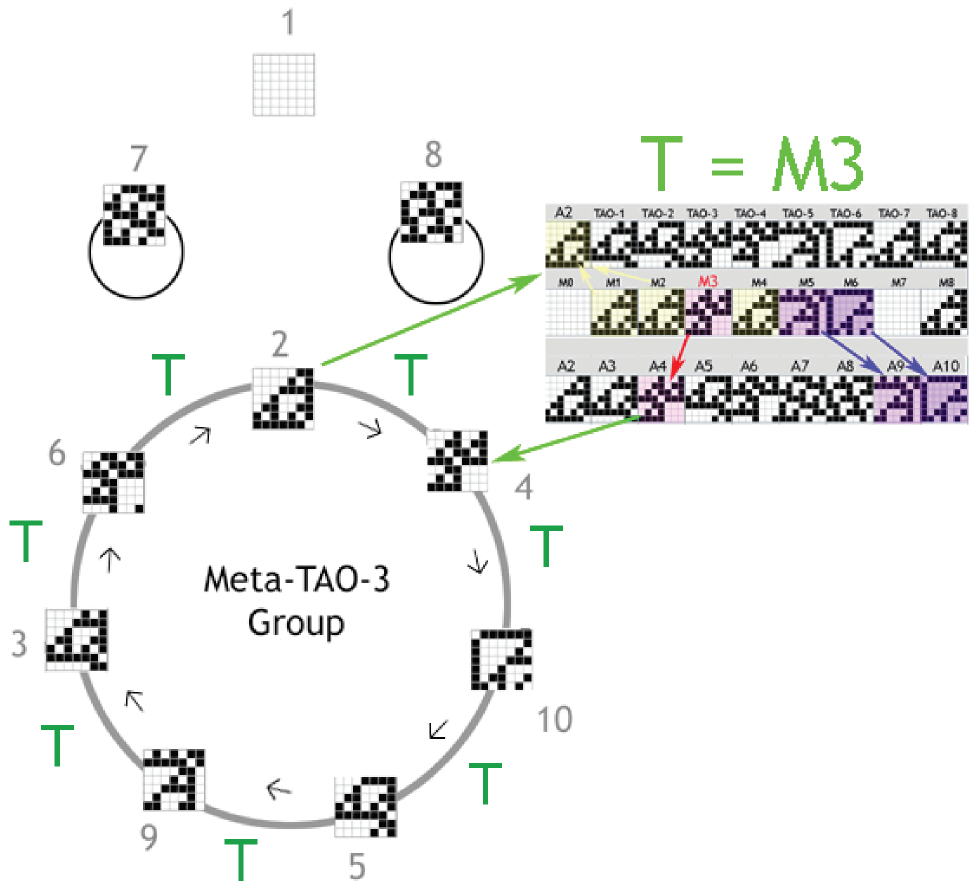


Figure 8 *The basin landscape of the n = 7 XOR Ring showing four symmetry groups. Only the attractors (which are numbered 1 through 10) are shown; tributaries are not shown. Iterations are on the abscissa and nodes on the ordinate. Basin 1, though of l = 1, is represented for 7 iterations to keep its image size comparable. All symmetry transforms, T, are Meta-TAO-3's of one Attractor leading into another attractor. Attractors A1, A7, and A8 each comprise a symmetry group in which they are the only element. Attractors A2, A4, A10, A5, A9, A3, and A6 comprise the fourth symmetry group.*

transform, you can derive any of the other attractors that are in the group with Attractor 2 without coding all the information required to specify the other attractor matrices individually.

Besides this informational efficiency, these transforms—by their definition—are a way for a system, based purely on information that is *local* to a given attractor, to move among its attractor cycles, at least within symmetry groups. In terms of an adaptive walk on a landscape, transforms, such as Meta-TAO-3, which produce symmetry groups are alternative and efficient procedures for coding large parts of a landscape and, most importantly, they provide a formal and therefore well-defined basis for how to navigate from one attractor to other attractors based on information contained in a single attractor. There is one obvious boundary condition on this principle. Notice that such transforms will not move a system between symmetry groups. In short, Meta-TAO-3 alone will never take a system from Basin 2 to Basin 7.

Extending the Results to Other Rings

We have so far given one example, an *n*7 XOR ring. We have produced enough other examples using more complex Boolean operators and different numbers of nodes to conclude that symmetry groups like the one we presented here for the N7 XOR Ring are frequent but not universal in the ring network structure.

Take, for example, a Ring with $n = 9$ and $k = 3$ where all nodes are self-referencing. Since all nodes take input from themselves, this leaves $k - 1 = 3 - 1 = 2$ inputs from other nodes. Since this is a ring, every node takes input from its two nearest clockwise neighbors. Thus the network appears as woven threads passing through and skipping nodes alternately. When $k = 3$, the Boolean truth table will have eight rows required by the combination of three binary inputs. The truth vector we used in this example was {10001110} which is NOR followed by NAND.

The landscape generated by this particular woven ring has seventy-five $l = 1$ (fixed point) attractors, one $l = 2$ attractor, and four $l = 18$ attractors. The results below show search results for the four $l = 18$ attractors (A2, A3, A4, and A5) seeking symmetry groups though META-TAO transformations.

- Basin 2 ⟹ [Basin 2] vs TAO-1 ⟹ META-1 vs TAO-3 ⟹ Basin 3;
- Basin 2 ⟹ [Basin 2] vs TAO-1 ⟹ META-1 vs TAO-7 ⟹ Basin 4;
- Basin 2 ⟹ [Basin 2] vs TAO-1 ⟹ META-7 vs TAO-8 ⟹ Basin 5;
- Basin 4 ⟹ [Basin 4] vs TAO-1 ⟹ META-1 vs TAO-6 ⟹ Basin 3;
- Basin 4 ⟹ [Basin 4] vs TAO-1 ⟹ META-1 vs TAO-7 ⟹ Basin 5;
- Basin 4 ⟹ META-1 vs TAO-5 ⟹ META-6 vs TAO-3 ⟹ Basin 2;
- Basin 5 ⟹ [Basin 5] vs TAO-1 ⟹ META-5 vs TAO-7 ⟹ Basin 2;
- Basin 5 ⟹ [Basin 5] vs TAO-1 ⟹ META-7 vs TAO-2 ⟹ Basin 3;
- Basin 5 ⟹ META-1 vs TAO-5 ⟹ META-5 vs TAO-1 ⟹ Basin 4.

In the first line of these search results the first matrix comparison [Basin 2- vs TAO-1] is itself compared to the results of the second matrix comparison [META-1 vs TAO-1]. The comparison of the two comparisons produces the attractor matrix (A3) for Basin 3.

If the reader sketches the connections (e.g., the attractor in Basin 2 goes to the attractor in Basins 4 and 5), it is easy to see that A2, A4, and A5 form a symmetry group in which all attractors can find each other. But in this more complex woven ring we were required to use the double transforms shown in the search results to move from one attractor to another. It is also evident that A3 does not have a transform that takes it to any of the other three attractors that have $l = 18$.

We also examined a hierarchical structure (see for example, Watts, 1999: 61). Obviously, hierarchical structures are at least nominally the structure of many organizations. This hierarchy had fifteen nodes branching in twos from the top node (the top row had one node, the second row had two nodes, the third row four nodes and the bottom row had eight nodes). All nodes gave input upwards to the level above. The top node had a NAND truth table and all others had an XOR truth table. There is no deep theory in our choices for designing this network; we simply wanted to explore at least one case of a hierarchical network. For this network we found no symmetry groups despite the fact that its landscape had around two thousand attractors. Thus this hierarchy's behavior had a great deal of variability in its behavior but it could not move directly from one attractor to another using symmetry transforms.

In our simulation results we have shown in detail how our phase portrait methodology can be used to find symmetry groups on a landscape generated by an *n*7 *n*2 XOR ring. The XOR operator (between two inputs) is {0110}. Then we hint at the generalizability of these results to at least some networks that have a ring structure by presenting briefly the results of searching for symmetry groups in an *n*9 *k*3 NOR-NAND Ring. The latter case is interesting for two reasons. First the ring structure is more complex (a weave). Second the relational operator used by each node is no longer the XOR operator (which could be argued constitutes a very special case because it is the basis of TAO, the Boolean derivative) but rather a complex truth-vector {10001110}. Both of these reasons are interesting ways to break out of the limits of the highly detailed example we have developed here. Finally, we have shown that preliminary explorations of a single case of a hierarchical network (imbued with symmetry) which did not yield symmetry groups on its emergent landscapes.

Discussion

Summary

In this paper we have presented two important developments for the understanding of *nk* Boolean landscapes. The first is a foundational methodology—the formalization of a discrete phase portrait for analyzing the dynamics of Boolean landscapes. The second, using the first, is the discovery that, in some cases at least, the attractors on a Boolean landscape can fall into symmetry groups. Symmetry groups are defined by transforms which, by their nature, transform one object into another. Consequently an attractor on a landscape, if it is in a symmetry group, can be transformed directly into another attractor (fitness peak). This has strong implications for theories of adaptation whether those theories address biological evolution, organizational behavior, or any of the other content areas that have used Boolean landscapes as a theoretical basis.

A Symmetry-Based Method for "Fearless" Leaps in Adaptive Walks

If we assume that model-based landscapes are useful models of open systems, our results provide a foundation for thinking of symmetry as a frame of reference for how adaptation might work. Adaptation is frequently modeled through recombination and mutation. Crossover recombination proceeds through small, local changes that are unlikely to move a system out of its current basin of attraction and that leave a system near its current attractor. Staying close to a fitness peak is highly functional for sexual reproduction, for example, where offspring need to be the same as parents in most important ways. In contrast, large perturbations through mutation can propel a system into another basin but, in our metaphor, large random perturbations are "fearful" leaps to unknown territory, which, based on chance alone, is likely to be a valley of low adaptation.

As fearful, in Blake's metaphor, as the pervasive symmetry underlying natural process may be, here we have shown that symmetry can provide a new method for getting unstuck from a fitness peak, for making, as it were, "fearless" leaps not back to the same location nor to a location of unspecified fitness but directly to other attractors via the transforms that define symmetry groups. If there are underlying symmetries in the structure of a system (gene regulation networks, communication networks, social networks, business organizations) then this paper suggests that the adaptive landscapes that emerge from the symmetries in the network can, by the Curie Principle, be imbued with some degree to symmetry too. As a consequence a system can capitalize on this symmetry to detect other stable basins on the same adaptive landscape. For functional adaptation an attractor in one of these other basins may be more adaptive for the current environment than is the current attractor. A context that is a crisis on one peak may well be an opportunity on another peak.

There is another critical point. The fearless leaps we propose require no meta-knowledge of the landscape. A symmetry-group leap directly to another attractor requires only information available in the current attractor. A system can move from its current location to another peak without prior knowledge of the new peak. And it can make such moves multiple times using the same symmetry transform.

But we do make another assumption. We assume that intrinsic to systems is some functionality formally equivalent to TAO's (derivatives) and Meta-TAOs (phase portraits). Of course natural systems don't have the exact functions found in mathematical models. We don't assume a dog who catches a ball bounced off a barn is calculating the equations of Newtonian mathematics as it watches the ball's complex trajectory and then runs and leaps; we only assume that some functionality in the dog's biology is equivalent to the Newtonian description. Here, we don't assume that genetic regulatory networks, social networks, power grids, business organizations, and so on calculate Boolean TAO's. Rather, we propose the Boolean model presented here is a proof of concept that if systems had some internal functionality that maps to our TAO's and Meta-TAOs they could evolve fearlessly.

Our final point is a reconsideration of the idea of fitness. In essence, symmetry transforms offer an alternative to the (often arbitrary) 0 to 1 metric that defines fitness on adaptive landscape models. Instead symmetry transforms redefine the fitness of systems as the ability to move directly to another fitness peak. In essence, adaptivity becomes defined as (1) a rich variety of stable patterns available to an evolving system (this is an issue we have not directly addressed here) and (2) the ability of a system to move nimbly between stable patterns as new environmental considerations arise. The second part of this definition of fitness is enhanced by the degree to which a system's stable patterns fall into symmetry groups, which itself is influenced by the symmetry of the network that generates those stable patterns.

The relentless adaptive pressures on species, on businesses, and other kinds of organizations and systems means they must change. In a dynamical systems perspective these change are not always, or even usually, steady incremental changes across time but rather sudden phase shifts. In terms of adaptive landscapes, incremental movement across fitness valleys are problematic at best. Symmetry transforms allow necessary shifts based on information that can be identified internally through an examination of the current stable pattern. Once a proper transformation has been identified and applied there is still the risk that a system will end up in a different pattern that is still fairly low in fitness for the current context or crisis. But symmetry transforms allow multiple moves directly to other peaks in a symmetry group increasing the chances of finding a peak that turns the current crisis into an opportunity.

References

Abraham, R. H., & Shaw, C. D. (1992). *Dynamics: The geometry of behavior part one: Periodic behavior*. ISBN 0-201-56716-4.

Barabási, A. L. (2005). *Linked*. ISBN 0-452-28439-2.

Brown, J. S., Cohen, Y. & Vincent, T. L. (2007). Adaptive dynamics with vector-valued strategies. *Evolutionary Ecological Research, ISSN* 1522–0613, *9*, 719-756.

Butner, J., Cooper, J., Malloy, T. E., & Smith, T. (2007, July). The knowable unknowns: Winding and unwinding Meta-TAO under broken symmetries. The

Society for Chaos Theory in Psychology and the Life Sciences Annual International Conference, Orange, CA.

Cooper, J., Butner, J., Malloy, T. E., & Smith, T. (2007, July). Folding Sierpinski Gaskets: The structure of Boolean Derivatives and Meta-Derivatives. The Society for Chaos Theory in Psychology and the Life Sciences Annual International Conference, Orange, CA.

Farnell, C. C. & Williams, J. D. (2010). Ion thruster grid design using an evolutionary algorithm. *Journal of Propulsion and Power*, ISSN 0748-4658, *26*, 125.

Gravner, J., Pitman, D., & Gavrilets, S. (2007). Percolation on fitness landscapes: Effects of correlation, phenotype, and incompatibilities. *Journal of Theoretical Biology*, *ISSN* 1522–0613, *248*, 627-645.

Guggenheimer, H. (1977). *Differential Geometry*, New ISBN 0-486-63433-7.

Holland, J. H. (1998). *Emergence*. ISBN 0-7382-0142-1.

Haslett, T. (2005). Local rules and fitness landscapes: A catastrophe model. *Nonlinear Dynamics, Psychology, and Life Sciences, ISSN* 1090-0578, *4*, 67-86.

Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, ISSN 0022-5193, *22*, 437-467.

Kauffman, S. A. (1993). *The origins of order: Self-organization and selection in evolution*. ISBN 0-19-507951-5.

Kauffman, S. A. (1995). *At home in the universe: The search for the laws of self-organization and complexity*. ISBN 0-19-509559-5.

Levinthal, D. (1997). Adaptation on Rugged Landscapes. *Management Science*, *ISSN: 0025-1909, 43*: 934–950.

Malloy, T. E., Bostic St Clair, C. & Grinder, J. (2005). Steps to an ecology of emergence. *Cybernetics & Human Knowing*, *ISSN*: 0907-0877, *12*, 102-119.

Malloy, T. E., Jensen, G. C., & Song, T. (2005). Mapping knowledge to Boolean dynamic systems in Bateson's epistemology. *Nonlinear Dynamics, Psychology, and Life Sciences*, *ISSN* 1090-0578, *9* , 37-60.

Malloy, T. E. & Jensen, G. C. (2008). Dynamic constancy as a basis for perceptual hierarchies. *Nonlinear Dynamics, Psychology, and Life Sciences, ISSN* 1090-0578, *12*, 191-203.

Mitchell, M. (2009). *Complexity*. *ISBN* 0195124413.

Mulvey, G.M., Amazeen, P. G., & Riley, M. A. (2005). The use of (symmetry) group theory as a predictive tool for studying bimanual coordination. *Journal of Motor Behavior*, *ISSN: 0022-2895, 37*, 295-309.

Rhodes, M. L, & Donnelly-Cox, G. (2008). Social entrepreneurship as a performance landscape: The case of 'Front Line.' *E:CO Emergence: Complexity and Organization*, ISSN: 1521-3250, *10*, 35-50.

Rivkin, J. W. & Siggelkow, N. (2002). Organization sticking points on NK Boolean landscapes. *Complexity*, ISSN *1076-2787*, *7*, 31-43.

Skellet, B., Cairnes, B., Geard, N., Tonkes, B., & Wiles, J. (2005). Maximally rugged NK landscapes contain the highest peaks. *GEOCC, 05*, 579-584.

Stewart, I. & Golubitsky, M. (1992). *Fearful symmetry*. ISBN 0-631-18251-9.

Watts, D. J. (1999). *Small worlds: The dynamics of networks between order and randomness*. ISBN 10: 0691117047.

Winfree, A. T. (2001). *The geometry of biological time*. (2nd Ed.) ISBN 0-387-98992-7.

Winter, S. G., Cattani, G., & Dorsch, A. (2007). The value of moderate obsession: Insights from a new mode of organizational search. *Organization Science*, ISSN: 1047-7039, *18*, 403-419.

Wolfram, S. (2002). *A new kind of science*. ISBN 1-57955-008-8.

Wright, S. (1931). Evolution in Mendelian populations. *Genetics*, *ISSN*: *0016-6731*, *16*, 97-159.

Yossi, B. & Poli, R. (2005). Information landscapes and the analysis of search algorithms. . *GEOCC*, *05*, 1287-1294.

Zee, A. (1986). Fearful Symmetry: The Search for Beauty in Modern Physics. *ISBN*-10: 0691134820.

Thomas E. Malloy, Ph.D., Associate Professor of Psychology, University of Utah. Tom's research interests include dynamical systems, particularly as they can be applied to mental processes in general and visual perception in particular. He has published articles in simulating Gregory Bateson's ecological epistemology using Stuart Kauffman's Boolean systems. He also publishes articles about his development of open-code educational software including OPEN Learning Management System, the online Utah Virtual Lab, GroupWare (software for student-graded online collaborative writing) and other innovative educational software. His work can be accessed directly at www.psych.utah.edu/dysys.

Jonathan Eric Butner, Ph, D., Associate Professor of Psychology, University of Utah. Jonathan's research interests include dynamical systems and quantitative psychology, particularly as they can be applied to groups and dyadic processes. He has published articles on modeling nonlinear oscillatory phenomena, affect coordination in couples, and the fractal pacing of conversations. His work involves the integration of quantitative and theoretical approaches and tends to focus on measurable phenomena in psychology, though he has some experience with simulations.

Chase Dickerson, Undergraduate of English and Computer Science, University of Utah. Chase's interests include dynamical systems theory and natural language processing More information about Chase can be found at http://www.eng.utah.edu/~dickerso/.

Joel M. Cooper, Ph. D. did the work for this paper as a graduate student in Cognitive Psychology, University of Utah. Joel's research interests span a variety of topics related to dynamical systems, traffic congestion, and driver distraction. He has published articles and conference proceedings that integrate these various disciplines in novel ways. He is currently at the Texas Transportation Institute at Texas A&M University.