System Behaviors and Measures: Logical Complexity and State Complexity in Naval Weapons Elevators

Roger J. Anderson
Robert H. Sturges
Department of Industrial and Systems Engineering
Virginia Tech

Behavior-based optimization is proposed to take advantage of relationships between complexity and optimality with respect to both performance and robustness. Two dynamic measures are presented: logical and state complexities in the case of naval weapons elevator design. Logical complexity measure is defined as the ratio of the length of the logical evolution to the temporal evolution length, while state complexity is the measure identifying the number of unique states in an evolution. A system with more complexity is found to result in an increase of throughput.

1. Introduction

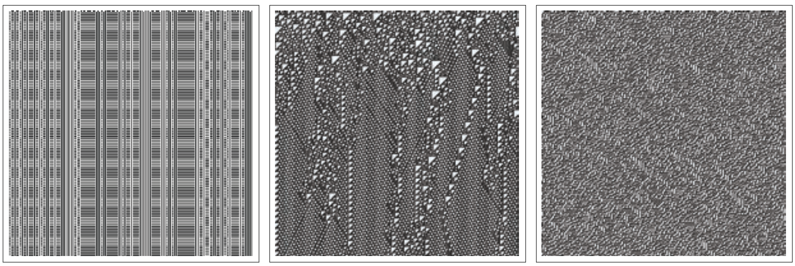
In complex systems having nonlinear interactions and system element independencies, the global system behavior may not be predictable from the set of rules dictating local interactions among elements. The characteristics of complex systems present difficulties with respect to optimization using traditional techniques. These techniques utilize some measure or measures of performance through the identification and manipulation of relevant system variables, largely ignoring the behavior of systems during operation.

Here we explain the essence of the system behaviors by using elementary cellular automata with a transition function. It might initially appear that elementary cellular automata are capable of producing only simple behaviors because of the apparent simplicity of the rule sets. Despite their simple construction, elementary cellular automata are capable of producing all types of behavior qualitatively possible in any system. Furthermore, the initial conditions required to produce this range of behavior need not be complicated. A certain form of behavior is often inherent to a given rule set, regardless of the initial conditions. It should be noted that there is a possibility for a given rule set to behave differently based on the initial conditions of the system.

https://doi.org/10.25088/ComplexSystems.22.3.247

On this basis, system behaviors can be classified into four groups by intuitive perception.

Generally, all behaviors have a qualitative nature associated with them. We have been able to classify the behavior mainly through our intuitive perceptive abilities [1, 2]. In example cellular automata evolutions, shown in Figure 1, it is apparent that (b) is the most complex of the three evolutions. Figure 1(a) is simple to represent by describing the fixed point attractor state, while Figure 1(c) is indistinguishable from a series of states generated from a simple pseudorandom state generator.



(a) (b) (c)

Figure 1. Three qualitative behaviors from elementary cellular automata rule sets. Evolution (a) is from rule 4 and represents simple class II behavior. Evolution (b) is class IV complex behavior from rule 110 (the description of rule numbers is given in Section 2). Class III random behavior appears in (c) from rule 45.

This perception of complexity, while powerful, is often subjective, and only an objective measure of complexity provides an absolute description of a system that can validate intuition. Although there is no exact quantitative definition of complexity, several parameters can together explain a characteristic of a system behavior. Since there are so many measures, both static and dynamic, only the particular measures applied in our research are presented in this paper.

Additionally, we discuss how the process of coevolution—the simultaneous evolution of interdependent species—evolves to a complex regime at the transition region between order and chaos. More importantly, we will see that the complex regime corresponds to optimal performance in terms of fitness in one of the models.

After the system behaviors have been presented, the layouts of elevator group control and naval weapons elevators, where our experiment was performed, are illustrated. It is beneficial to be familiar with the system. Understanding how it works can easily draw a brief picture of how the complexity of such a system arises.

https://doi.org/10.25088/ComplexSystems.22.3.247

With complexity measures presented, we are ready to set up the experiment, although there is the fact that the naval weapons elevator system is extremely large. The use of exhaustive simulations is not intended to suggest that this approach is required when using behavior toward optimization, but only as a means for full characterization of the possible relationships. Consequently, simulation has been proposed as the only approach providing the predictive shortcut to indicate the performance or behavior of that configuration. In Section 4, all measures of complexity for weapons elevator simulations, both static and dynamic, are briefly introduced. Then, we discuss two dynamic measures, logical complexity and state complexity in Sections 9 and 10, respectively.

Logical complexity is the ratio of the length of the logical evolution to the temporal evolution length. It indicates the amount of information required to express the logical sequence of states. As we examine the relationship between performance and behavior of the system, throughput will be measured with respect to complexity throughout this paper. The results of our experiment indicated that evolution acting relatively more logically yields greater throughput or performance. Furthermore, we present the logical complexity of the systems of different sizes. For each size of the system, we measured logical complexity of evolution subsets composed of nonhalting, complete, robust, unique and nonhalting, unique and complete, unique and robust, and mimics evolutions to illustrate the trends of the complexity with respect to system sizes and types of evolutions.

State complexity expresses the number of unique states used in an evolution with respect to the temporal evolution length. It is the ratio of the number of distinct states entered to the product of the number of items carried and the system cycle time. From our simulations, there existed evidence suggesting the relationship between state complexity and throughput. Like logical complexity, the plots of state complexity with respect to configurations and evolution subsets are presented.

Although a naval weapons elevator system is such a rich model for the study of complexity of material handling systems, not all possible configurations are measured here. Due to a large amount of data available for each configuration, only those configurations representing each type of distinct behavior are presented.

2. Cellular Automata

In order to address the possibility of artificial self-reproduction, or the ability of a machine to replicate itself in an electronic computer and avoid the costs associated with hardware, John von Neumann, often

https://doi.org/10.25088/ComplexSystems.22.3.247

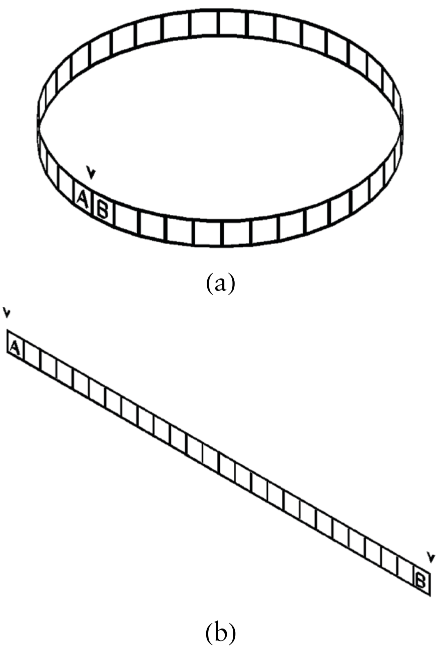
considered the father of the modern computer, created an abstract mathematical representation of a machine, known as *automata*. Automata are capable of self-replication if given the proper logic and initial conditions [3].

Cellular automata consist of a network of elements, or cells, that exist on some lattice structure. Cellular automata networks operate by the application of a set of transition functions Θ to each cell in the network. The network evolves as the transition functions are recursively applied.

To further explain cellular automata systems, we will use elementary cellular automata, the simplest configuration capable of all forms of possible behavior [1, 4, 5].

2.1 Elementary Cellular Automata

One-dimensional cellular automata networks consist of a line of cells, with each cell having a neighbor to the left and right. In a closed network with periodic boundary conditions, the cells therefore form a ring. When represented in two dimensions, the ring must be "broken" and unfolded to capture all information about the state of the network.



(a)

(b)

Figure 2. A one-dimensional closed network with periodic boundary conditions forms (a) a closed ring. Expressed in one dimension, the network is broken as in (b), but the ends A and B remain neighbors.

As a one-dimensional network is evolved, we can visualize the current state of the system in two dimensions as in Figure 2, which is updated with each successive application of the transition functions on all cells. To visualize a history of an evolution, we simply add on a new row for each complete mapping. The result is a two-dimensional grid n columns wide and $t+1$ rows long, where n is the number of cells involved in the network and t is the number of evolution steps. Evolution of the actual topography of the network shown in Figure 1 results in a hollow cylinder. Since the term evolution is used to describe the recursive mapping of the transition functions, we refer to each row in the evolution in integer values of time, so that the state of the system at time step t_1 evolves to the system state at time step t_2.

Cells in elementary cellular automata networks can exist in one of two states. In binary terms, these values are 0 and 1, equivalent to the Boolean values of false and true. In our visual representations, the 0 and 1 states are equivalent to white and black cells, respectively.

2.2 Transition Function

The transition functions associated with a cellular automaton represent the conditional logic that controls the operation of the system. Elementary cellular automata have a neighborhood with a radius of one cell, $r = 1$.

In general, for automata with k possible states and a neighborhood of r adjacent cells to the left and right, there are $k^{(2r+1)}$ combinations of cell states that form the input to the transition functions. For elementary cellular automata with two possible states and a neighborhood of three cells ($r = 1$), the number of combinations of cell states is $2^3 = 8$, which are presented in Figure 3.

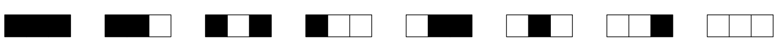


Figure 3. The eight combinations of three cells with two possible states that define elementary cellular automata transition function inputs.

Each of these sequences can evolve to one of two states. In general, there are $k^{k^{(2r+1)}}$ possible sets of transition functions. For the case of elementary cellular automata, there are $2^{2^3} = 256$ possible rule sets, or combinations of evolved states. This evolution structure leads to a convenient method for expressing a rule set without explicit description of cell digit sequences [5].

Each combination of cell sequences is assigned a decimal value equivalent to the binary representation of cells, letting black and white equal 1 and 0, respectively. Each of these decimal equivalents

https://doi.org/10.25088/ComplexSystems.22.3.247

represents a single bit in an eight-bit byte, with the bit evolving from the sequence of three black cells representing the most significant bit and the bit evolving from the sequence of three white cells representing the least significant bit. The eight-bit binary sequence of evolved cell states, converted to its decimal equivalent, yields a single coded "rule number" that completely describes the evolution logic. An example of the coding process is illustrated in Figure 4 for elementary cellular automata rule 22.

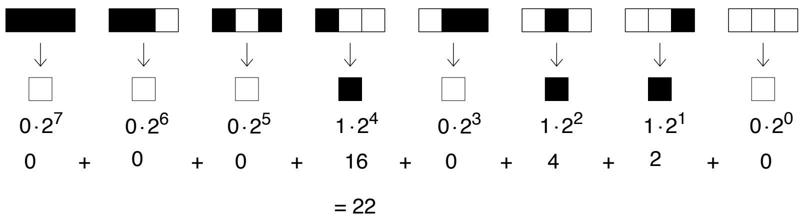


Figure 4. Rule numbers for cellular automata are based on the evolved binary value (white = 0 and black = 1) of a transition function and the corresponding bit value of the transition function, assuming transition functions are arranged in some standard order. For elementary rule 22, the first, second, and fourth transition functions' bits evolve to a black, or 1, state, meaning the evolved "byte" is (00010110). The decimal representation of this binary value is 22.

3. System Behavior

In general, a system behavior ranges from four to six classes depending on the source [1, 6–8]. In Wolfram's classification system, four classes of behavior are as follows.

Class I: Trivial Behavior. System states rapidly evolve to a uniform quiescent state. An example of trivial behavior is found in rule 0 as shown in Figure 5. This rule creates evolutions of all white cells in one step, regardless of initial conditions.

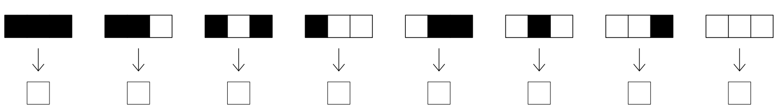


Figure 5. The rule set for elementary cellular automata rule 0. All possible combinations of three cells map to a white cell.

An example of a rule set that is normally regarded as producing simple behavior, but is capable of trivial behavior, is rule 90. Rule 90, shown in Figure 6, is an additive rule. An additive rule has the prop-

https://doi.org/10.25088/ComplexSystems.22.3.247

erty that the evolution is the spatial superposition of the independent evolutions of each initial cell. Because of this property, rule 90 is defined as simple despite the ability to produce apparently random evolutions. The additive property also leads to trivial behavior in rule 90. When the number of cells N in the network is equal to $2^n \mid (n = 1, 2, 3, \ldots)$, the evolution reaches a trivial state in exactly 2^{n-1} steps, regardless of the initial conditions. Example evolutions illustrating this behavior using various system sizes and initial conditions are shown in Figure 7.

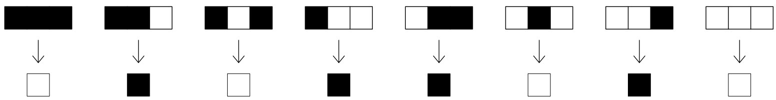
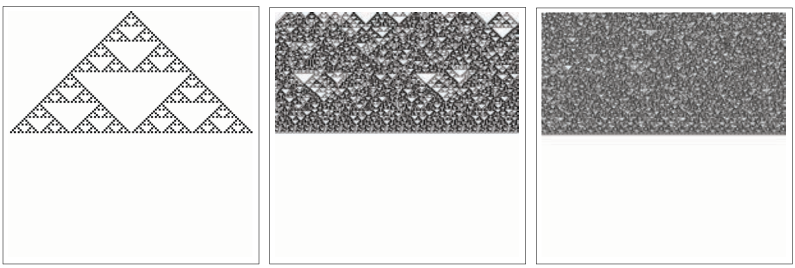


Figure 6. The rule set for elementary cellular automata rule 90, an additive rule.

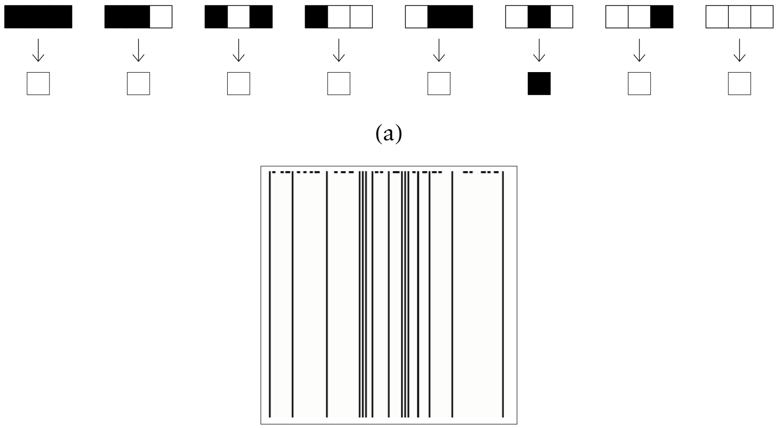


(a) (b) (c)

Figure 7. Trivial behavior in a noninherently trivial rule set. Rule 90, an additive rule, produces trivial behavior after 2^{n-1} evolution steps for a network with 2^n cells, regardless of the initial conditions. The evolution in (a) shows rule 90 starting from a single black cell and $N = 128$, (b) from random initial conditions with a 20% black cell density and $N = 256$, and (c) from random initial conditions with a 50% black cell density and $N = 512$.

Class II: Simple Behavior. The system evolves, yielding a repetitive pattern of finite limit cycles or streaks resulting from fixed-point attractors. Rule 4 provides an example of a simple behavior, known as a "filter" rule, which identifies the number of white-black-white cell sequences in the initial conditions. The result of rule 4 is shown in Figure 8. Since the number of streaks depends on the number of initial white-black-white cell sequences in the initial conditions, the evolution of the network is very predictable if the initial conditions are known.

https://doi.org/10.25088/ComplexSystems.22.3.247



(a)

(b)

Figure 8. (a) The rule set for elementary cellular automata rule 4. Rule 4 is a filtering rule, identifying the number of white-black-white sequences. (b) The first 150 evolution steps of rule 4 from random initial conditions. There are 150 cells in the network.

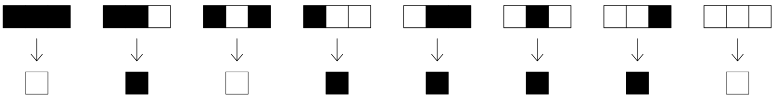
Figure 9(a) shows the transition functions for rule 94. Like rule 4, the evolution of rule 94 is dependent on the initial conditions. However, rule 94 typically has limit cycle attractors rather than fixed-point attractors. In the example evolution shown in Figure 9(b), rule 94 has a limit cycle with a period of six evolution steps, the first common multiple of the period 3 sequence in the leftmost band and the period 2 sequences on the right bands.

While the example evolution of rule 94 is more complicated than that for rule 4, the behavior in Figure 9(b) is still considered simple. Beyond the transient response, the network falls into a repetitive sequence of network states.

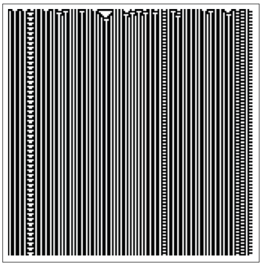
Class III: Chaotic Behavior. There are no attractors and the states are essentially random. Since cellular automata are deterministic systems, it is guaranteed that any cellular automata evolution is periodic. The behavior of a rule that creates chaotic behavior is therefore technically not random. However, the evolution may be required to evolve through $2\,N$ distinct states for a network consisting of N cells. For even modest size N, this means that a rule set exhibiting random behavior locally can essentially be treated as globally random.

The rule set for rule 30 is shown in Figure 10(a). Despite the simplicity of the eight rules defining the rule set, rule 30 inherently produces chaotic behavior for essentially all but a trivial set of initial conditions [9]. Figure 10(b) shows the evolution after 100 steps. Looking at this evolution, we see some periodic structure on the left of the tri-

https://doi.org/10.25088/ComplexSystems.22.3.247

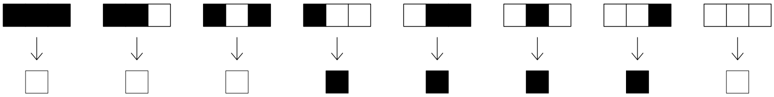


(a)

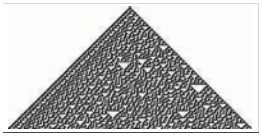


(b)

Figure 9. (a) The rule set for elementary cellular automata rule 94. Rule 94 often has limit cycle attractors rather than fixed-point attractors. (b) The first 150 evolution steps of rule 94 from random initial conditions. There are 150 cells in the network.



(a)



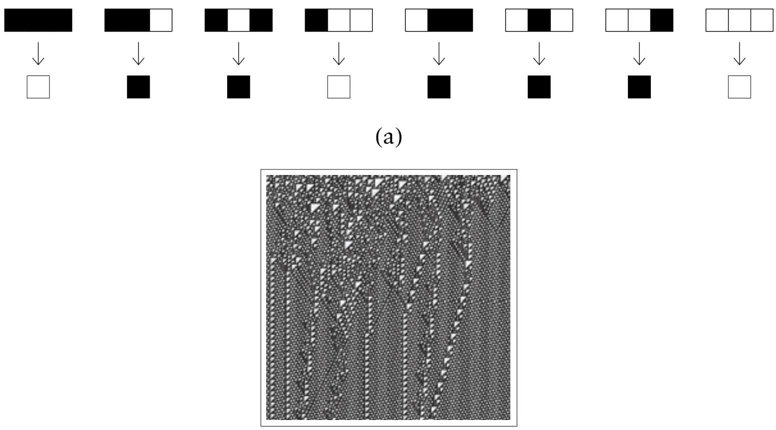
(b)

(c)

Figure 10. (a) The rule set for elementary cellular automata rule 30. Despite the simplicity of the rule set, rule 30 inherently produces chaotic behavior. Evolutions of rule 30 starting from a single centered black cell for (b) 100 steps and (c) 500 steps.

angle. However, on the right side, there is no discernible pattern, although there are similar structures, inverted triangles, throughout. These local structures are of different sizes and follow no global structure. We might at first assume that we have not evolved the network sufficiently to observe a pattern or attractor with a longer cycle than we have seen in simpler evolutions. But when the system is evolved further, as in Figure 10(c), the same behavior is apparent.

Class IV: Complex Behavior. Represents a mixture of simple and random behaviors, and has been described as a transition region between the two. Like a phase transition from solid to liquid, the complex regime is a transition regime from order to chaos. This form of behavior is illustrated in rule 110, described in Figure 11(a).



(a)

(b)

Figure 11. (a) The rule set for elementary cellular automata rule 110. Rule 110 results in complex behavior, a mixture of order and chaos. (b) The first 300 evolution steps of rule 110 from random initial conditions. There are 300 cells in the network.

Figure 11(b) illustrates an example evolution of rule 110 from random initial conditions for the first 300 evolution steps. After a transient period, the system settles down to a regular background, through which random structures meander and collide to form new structures with different trajectories. As with chaotic evolutions, the system is deterministic and will therefore have a repetition period. Like chaotic systems, this limit cycle is often very long, making complex evolutions unpredictable. To determine the state of a given cell at a specific evolution step, explicit evolution of the network from initial conditions is therefore required.

According to biological sciences, particularly evolutionary processes, there is evidence that the highest average fitness occurs when the coevolution process is in a complex regime. In other words, complex behavior allows optimality, but the other classes do not. For instance, genetic algorithms usually rely on the complex interactions of "schema" to search through large design spaces [10, 11].

Since complexity is usually only qualitatively defined, we must measure the level of complexity to quantitatively determine the complexity of the system.

4. Measures of Complexity

The behavior can be classified by intuitive perceptive abilities. Currently, there is no universal quantitative definition of complexity but several measures have been proposed, both static and dynamic. Static complexity measures, in practice, describe the potential types of behavior that a system can support while dynamic complexity measures describe the behavior that has emerged from a system. Since many measures have been proposed, only the most commonly known and applied measures are presented in this paper.

4.1 Static Measures

4.1.1 Shannon's Information

This measure is based on information theory for finding the information content in telecommunications. The information of a system of N parts is defined by I as shown in equation (1):

$$I = -K \sum_{i=0}^{N} p_i \log_2 p_i \tag{1}$$

where p_i is the probability of the i^{th} part/event/state occurring and K is a constant that accounts for units of measure. The information content of N, $I(N)$ is the sum of the information in each independent set of events as in equation (2):

$$I(N) = I(N_1) + I(N_2). \tag{2}$$

For example, the entropy of two outcomes with probability p and $q = 1 - p$ is given by $I = -\left(p \log_2 p + q \log_2 q\right)$. This example describes the entropy in the elementary cellular automata with $k = 2$ states. Entropy is maximal when $p_{\text{black}} = 0.5$ for a given cell. Entropy is 0 when either $p = 0$ ($q = 1$) or $p = 1$ ($q = 0$). In general, entropy is maximal for random systems and minimal for ordered ones.

4.1.2 λ Parameter

This measure, introduced by Langton, is the method used to describe the number of cell neighborhoods or transition functions of the 2^{2r+1}

https://doi.org/10.25088/ComplexSystems.22.3.247

possible neighborhoods evolving to a given state. It acts as an indicator of the potential activity of evolutions:

$$\lambda = \frac{m}{2^{2r+1}} \tag{3}$$

where m is the number of active cell neighborhoods and r is the radius of the neighborhood (e.g., in Figure 4, there are eight neighborhoods of three cells, each with $r = 1$). For $\lambda = 0$, there is no potential activity. As the value of λ is increased to 0.5, the corresponding behavior goes through fundamental changes, while for $\lambda = 0.5$, evolutions are chaotic without attractors. When λ is increased from low to high values, the behavior of evolutions passes through a transition region at some critical range corresponding to a complex regime.

4.1.3 Hierarchical Complexity

The complexity of hierarchical systems is based on the idea of clustering system components according to the strength of their mutual interactions. The most strongly interacting components are associated with upper parts of the hierarchy, followed by progressively weaker associated components. Complexity in the system is based on a measurement of the system described by the number of interactions between or within subtrees of a hierarchy. Complexity of a hierarchy $C(T)$ is defined as in equation (4):

$$C(T) = \log_2 D(T) = \log_2 \left\{ f(k_T) \prod_{j=1}^{k} D(T_j) \right\} \tag{4}$$

where $D(T)$ is the diversity of the tree found by counting all distinct interactions within clusters or subtrees of T. If the tree branches to a single leaf or set of single leaves, the diversity of each leaf is 1, as is the diversity of the root. The diversity becomes greater than 1 when at least two subtrees from one (sub)root have different structures. f_T is the form factor presenting the number of distinct subtrees stemming from a given (sub)root. It is calculated as the number of ways N_k that k subtrees can interact and is given as $2^{k_T} - 1$. A tree with a constant branching ratio has only one distinct subtree at every tier and therefore has a diversity of 1 and a complexity of 0. This follows intuition, as a constant branching ratio represents an ordered structure.

4.1.4 Simplicial Complexes

A simplicial complex is described by the association between the elementary networks combined to construct a system. The number of nodes in an n-simplex is $n + 1$.

https://doi.org/10.25088/ComplexSystems.22.3.247

In such a system, a single node or multiple nodes can be members of multiple distinct simplices. The relation between simplices is characterized by the simplices' q-connectivity, presenting the minimum number of shared edges between two particular simplices. The first structure vector $\vec{Q} = \{Q_D, Q_{D-1}, \ldots, Q_0\}$ describes the number of q-connected simplices of each value of q for $0 \leq q \leq D$, where D is the largest dimension simplex of the simplicial complex.

The complexity measure $K(C)$ of a simplicial complex based on the first structure vector $\vec{Q}$ is given by equation (5):

$$K(C) = \frac{2}{(D+1)(D+2)} \sum_{i=0}^{D} (i+1)\, Q_i. \tag{5}$$

4.1.5 Number of Components

A fundamental part of component counting, proposed by Bar-Yam, is that complexity is a function of the interactions between elements and with greater numbers of interacting elements comes greater potential for complex behavior. Key to using component counting as a complexity measure is the identification of relevant components and the scale of observation. Complexity, in general, is maximal on microscopic scales and decreases as the scales increase in size. The rate of decrease and the shape of the complexity profile depend on the behavior of the system.

The component counting method lacks ability to describe the behavior of the system but it is able to present the potential character of behavior supported by a system. In addition, component counting frequently ignores the connectivity between elements, an important consideration.

4.2 Dynamic Measures

4.2.1 Mutual Information

Mutual information describes how well information communicates through a system and presents the correlations between variables as well. The mutual information between two probabilities $\{p_i\}$ and $\{p_j\}$ using the joint probability $\{p_{ij}\}$ is given by equation (6):

$$M = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{p_i\, p_j}. \tag{6}$$

https://doi.org/10.25088/ComplexSystems.22.3.247

4.2.2 Algorithmic Complexity

Introduced by Solomonoff, Komogorov, and Chaitin [12–14], the algorithmic complexity $I_C(s)$ of a binary string s is defined to be the shortest program p that produces the output s on a universal computer U. Equation (7) shows the relation of the variables:

$$I_U(s) = \min_{U(p)=s} \log(p). \tag{7}$$

The value of $I_C(s)$ depends on the universal computer used. The difference between algorithmic complexities on universal computer U_1 and U_2 is limited by the additional information $\tau_{U_1\,U_2}$ required for each computer to emulate the other as presented in equation (8):

$$\left|I_{U_1}(s) - I_{U_2}(s)\right| \le \left|\tau_{U_1\,U_2}\right|. \tag{8}$$

It is practically difficult to set up an absolute measure of algorithmic complexity for a given system since no formal descriptive system can encompass all true theorems (e.g., Godel's incompleteness theorem). For example, consider two binary digit sequences, (a) 10101010101010101010 and (b) 10111001001010011010. Sequence (a) can be compressed to a short program `<Print '10' ten times>` and has low algorithmic complexity, while the sequence in (b) is incompressible and requires a program approximately as long as the digit sequence itself to represent it, resulting in high algorithmic complexity.

4.2.3 Computational Complexity

Computational complexity measures the minimal time or memory required by a universal computer to solve a particular problem. If $\sum_N^{(i)}$ is the initial state of a size-N problem with a final state solution using function f of $\sum_N^{(f)}$, then the computational complexity $H_C\,(\sum_N^{(f)}\,)$ is the time required τ for the program P running on the universal computer U to reach state $\sum_N^{(f)}$ from initial state $\sum_N^{(i)}$. The relation is in equation (9):

$$H_C(\sum_N^{(f)}) = \min_{U(P)=\sum_N^{(f)}} \tau_U(P). \tag{9}$$

Computational complexity treats all complexity measures in terms of functions, but a difficulty with this approach is that not all systems are describable in terms of functions.

4.2.4 Logical Depth

Logical depth D_U^L describes the time τ required for a universal computer U to execute a minimal program P^* to generate some output

https://doi.org/10.25088/ComplexSystems.22.3.247

representative of object O as in equation (10). Note that $S(O)$ is the binary string representation of the object O:

$$D_U^L(O) = \tau_U(P^*) \qquad \text{where } U(P^*) = S(O). \tag{10}$$

5. Complexity in Natural Systems

The real reason for looking at natural systems is that they show a strong correlation between complex behavior and adaptability. In natural systems, we will see that adaptability, not necessarily performance in a given environment, implies optimality. To illustrate the association between complex behavior, adaptability, and optimality in biological systems, we will investigate evolutionary theory and the models used to study evolution. Fundamental to the relationships of these attributes is the context of the network topology.

Coevolution is the simultaneous evolution of interdependent species that evolve to a complex regime at the transition region between order and chaos. Two models are presented to give an idea of the process of coevolution. The first model, described by Kauffman [15], shows that by tuning the parameters that determine how genes affect the fitness within an organism and among species, it is possible to create a continuum of evolutionary strategy behaviors, from stable to chaotic. In addition, the model shows that when the coevolution is in a complex regime, the aggregate fitness of all coevolving species is at a maximum. The second model, created by Holland [11, 16], uses a simulated ecosystem with no external measures of fitness, but still exhibits evolution toward complex regimes.

Kauffman describes two ultimate behaviors that can occur in a coevolving system of interdependent species. The first is known as the evolutionary stable strategy (ESS) regime of behavior. An ESS describes a Nash equilibrium [17] where each agent will always have at least one Nash strategy. The Nash strategy has the property that an agent will be better off following it as long as all other agents involved follow their own Nash strategy. In terms of fitness landscapes, all species have found a suitable local peak from which they have no incentive to move.

The second region of behavior is the Red Queen, or chaotic regime. In this regime, the coevolutionary process continually changes as a species deforms the fitness landscape of another. This second species in turns deforms the landscape of the first species, forming a feedback loop. This process becomes even more complicated as more species are added to the web of interactions.

Both the ESS and chaotic regimes result in low overall fitness for all species. In ESS behavior, species remain frozen at low local peaks on their fitness landscapes while in chaotic coevolution, the landscape of each species constantly changes, and a species that has found a peak is soon knocked off as the landscape is deformed by changes in other species.

Kauffman uses a four-parameter model to explore the relation between the relative rates of fitness landscape deformability and movement within a landscape and the resulting coevolutionary behavior with coupled fitness landscapes. The first parameter N describes the number of genes (or traits) that defines each organism. The values for each trait or gene are taken to be binary. The second parameter K describes the number of epistatic couplings between genes within an organism's genome. Epistatic coupling describes the fitness contribution of one allele (0 or 1) of one gene in relation to other specific genes in the genome. To define how genes interact between species and how fitness landscapes are coupled, the model uses a coevolution coupling parameter C, which indicates how each of the N genes in one species makes a fitness contribution that depends on C genes, again modeled as binary values, in each of the other connected species. The final parameter necessary in the model describes the number of species found in the ecosystem S.

Varying the values of the number of genes in each species, the number of intraspecies epistatic couplings, the number of interspecies gene couplings, and the species population in an ecosystem, it is possible to produce different evolutionary behaviors. When either the number of epistatic couplings K is high, or the number of interspecies gene couplings C is low, the ecosystem tends to settle to an evolutionary stable strategy where all species have found peaks in their fitness landscapes such that no mutation leads to increased fitness. When K is high, the landscape is rugged, resulting in many peaks for a species to get trapped on. When C is low, landscape deformation of species A that results from an adaptive walk of species B is minimal, so that the peak that A had found remains a peak. Behavior in the ESS regime also arises when the number of species in the ecosystem S is low. This result follows the logic that the fewer the species that can deform the landscape, the less the landscape has the potential to change. Kauffman identifies ESS behavior by plotting the fitness of species through successive generations [18]. When an ecosystem reaches a steady-state condition, no changes in fitness occur throughout the population, indicating no changes in genotypes. The ecosystem has found a Nash equilibrium, where all species have found their optimal configurations with respect to the configurations of the rest of the population.

Chaotic behavior results when K is low (when there are few peaks to get trapped on), when C is high (many interdependencies exist between species so that a change in one species can have a significant ef-

https://doi.org/10.25088/ComplexSystems.22.3.247

fect on another species' landscape), or when S is high (so that each fitness landscape is directly affected by many other species). In this regime of behavior, the peaks on each fitness landscape move away faster than the species can chase them and overall fitness is low.

The model indicates that the number of interspecies gene couplings and the number of species comprising an ecosystem have the greatest effect on the behavior of the coevolutionary process. When C is high, a single move by one species has significant impact on the fitness landscapes of other species because the chance that the genotype of a species is affected by a mutated gene in another species is greater than if C is low. Similarly, when the number of species in an ecosystem is high and connectivity is complete so that many species affect the fitness of a single species, there is a greater chance of a single landscape being deformed, and the landscape is also subject to the collective effects of multiple deformations.

These results indicate that both stable and chaotic regimes can exist in the same system, with the type of behavior dependent on the values of the input parameters. Tuning the parameters results in a transition between regimes, much like the change from simple to chaotic behavior observed when the value of the λ parameter, a characteristic of a rule set that indicates the level of activity, is increased in cellular automata systems. In this transition regime, as in cellular automata networks, coevolution is a mixture of ESS and chaotic behaviors.

To illustrate this mixture of behaviors, Kauffman uses a set of ecosystems coevolved with the same model parameter values. Multiple sets are used to demonstrate the effect of varying the number of epistatic couplings. The observed values are the number of ecosystems, evolved with the same number of epistatic couplings, that have found a Nash equilibrium within a specified number of generations. Each set of 50 ecosystems is assumed to contain 25 species, arranged in an arbitrary 5 by 5 lattice. Connectivity is not complete, but is restricted to the nearest neighbors to the north, east, south, and west. Boundary conditions are not periodic and species in the corners and on the edges of the lattice have two and three neighbors, respectively. The number of interspecies couplings is fixed to a single gene.

The results of varying the number of epistatic couplings indicate that, for low values of K, the coevolution process is always chaotic, with all 50 ecosystems in a state of perpetual variation at the end of 200 generations. Conversely, with K sufficiently high, all 50 ecosystems will find Nash equilibria within the 200 generations available. The rate at which the species approach their equilibria increases with increasing values of K. When K is at a value of 10 however, a percentage of the 50 trial ecosystems has not reached a stable strategy in the

https://doi.org/10.25088/ComplexSystems.22.3.247

200 generations available. This indicates a transition region from ordered to chaotic behavior for this set of parameters with respect to a time scale as the number of epistatic couplings increases.

As parameters are changed, the aggregate fitness also varies. As K is increased from low values, the average fitness of an ecosystem at first increases, and then decreases. Absolute values of fitness are also affected by the number of epistatic couplings. The most important result here, and key to the assumptions in this work, is that, unlike in cellular automata networks, where tuning the λ parameter corresponds to qualitative changes in behavior, a correlation exists between the behavior of the coevolution process and the performance of the species within the ecosystem, measured in terms of fitness. The highest average fitness occurs when the coevolution process is in a complex regime, in the transition region between ESS and chaotic behavior.

6. Elevator Group Control and Naval Weapons Elevators

A naval weapons elevator system is similar to any elevator system. For our purposes, the system consists of four types of elements that are entities (passengers or materiel), elevators, queues, and destinations. The main function of the system is the transportation of elements between the main deck and magazines sketched in Figure 12. Scheduled movements of the elevator can be divided into two types, that is, strike-down operations and strike-up operations. Like up-peak, a strike-down operation is a moving of an elevator from the main deck to magazines, while strike-up, like down-peak, is a moving of an elevator from magazines to the main deck. As up-peak empirically provides the greatest limitations on system performance, we will assume that strike-up and strike-down behave similarly and we will only consider strike-down operations for our study.

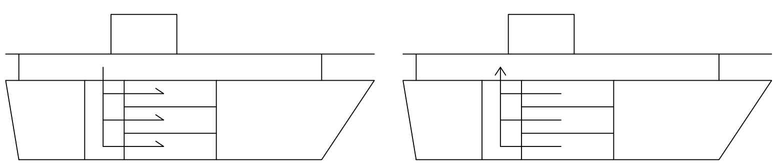


Figure 12. Strike-down and strike-up weapons transfer scenarios.

The most important measure of the performance of the weapons elevator system is the total cycle time. Minimizing the length of time ordnance remains on the main deck during strike-down and conveying ordnance to the main deck at a rate that matches the demand of flight operation during strike-up constitute the expected performance of the system.

For a better understanding of the system, we would like to review here the current configuration of a weapons elevator system. In an aircraft carrier, multiple elevators are employed to serve a number of magazines. Each elevator shaft operates with a single carriage always remaining within the confines of the shaft that has only a vertical degree of freedom. Elevator shafts are located along transverse watertight bulkheads. The shafts can usually access two compartments, one of which is through a transverse watertight bulkhead. Thus, a shaft can access multiple compartments at the same level as in Figure 13. To isolate the shaft from the compartments when the doors are closed, the doors between the shafts and magazines are also watertight. To isolate regions within the shaft, two hatches are provided within the elevator shaft shown in Figure 14.

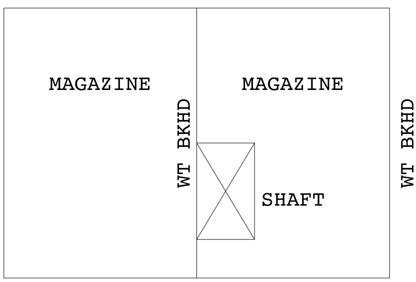


Figure 13. Plan view of a magazine layout. WT indicates a watertight partition.

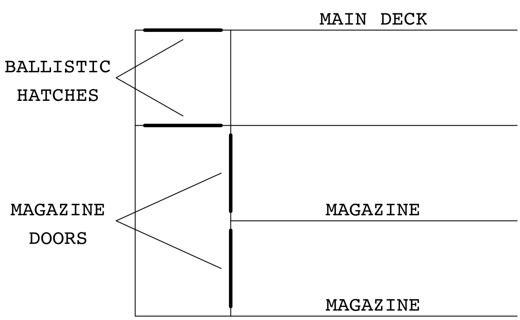


Figure 14. Elevation of magazine and shaft layout, showing ballistic hatches and doors.

The weapons elevator system operation commons involves multiple elevators, magazines, and queues that have significant interaction. The basic operational logic of the system can be described by the sequence of events in a strike-down operation in one elevator cycle in a system consisting of a single queue, carriage, and magazine as shown in Table 1. Note that the platform is initially located one level below the main deck and all doors and hatches are closed. Figure 15 illustrates the weapons elevator transfer operation events corresponding to Table 1.

Step	Event
1	Open main deck hatch
2	Move carriage to the main deck
3	Load carriage
4	Move carriage to the 02 level
5	Close main deck hatch
6	Open ballistic hatch
7	Move carriage to the magazine (below ballistic hatch)
8	Open magazine door
9	Unload carriage
10	Close magazine door
11	Move carriage to the 02 level
12	Close ballistic hatch

Table 1. Current weapons elevator transfer operation events.

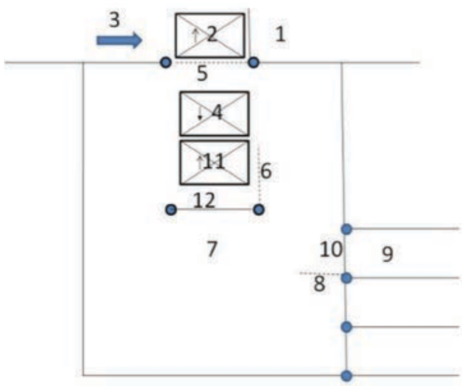


Figure 15. Theoretical operations of a current weapons elevator.

Although a naval weapons elevator system is similar to a commercial elevator system, there are some differences between them. The differences between both systems are with respect to the queues and the ballistic safety features (interlocks) of the weapons elevator system.

https://doi.org/10.25088/ComplexSystems.22.3.247

In a weapons elevator system, we have information about the queues, while the queues in the commercial elevator are considered as stochastic (equation (11)). The rate of passengers (items in queue) entering the system can be described by a Poisson distribution, where n is the number of calls being registered in the time interval T for an average rate of arrival λ:

$$p_r(n) = \frac{(\lambda T)^n}{n!} e^{-\lambda T}. \tag{11}$$

Due to several reasons, the queues in a naval weapons elevator system are not ideal and can encounter variability.

7. Naval Weapons Elevator Simulations

The physical layout and mode of operation employed in naval weapons elevators are two of an extremely large number of system configurations. In a large design space corresponding to systems with nonlinear dynamics, the process of reductionism, creating generalizations regarding the effects of a single variable, is not applicable. Simulation, therefore, is the only approach that provides the predictive shortcut to indicate the performance or behavior of that configuration.

Physical characteristics of a naval weapons elevator system describe three types of spaces—queues, shafts, and destinations—and also include the arrangement and interactions among the spaces. In this paper, which considers only the strike-down scenario, queues are the main deck areas and destinations are magazines. The relationships between spaces can be compactly described using three distinct incidence metrics: the matrix relating shafts and queues (SQ), the matrix relating shafts and magazines (SM), and the matrix relating queues and magazines (QM). Based on physical relationships, SQ and SM matrices are considered 0^{th}-order incidence matrices since they are adjacent spaces, and the QM matrix is considered a 1^{st}-order incidence matrix due to the indirect relationship between spaces. The relationship among the three matrices is shown in equation (12):

$$(QM) = (SQ)^T (SM). \tag{12}$$

Zeroth-order incidence matrices consist of only binary values indicating the presence or lack of interactions. However, elements in higher-order incidence matrices are not necessarily only binary values. The entries in the 1^{st}-order QM matrix indicate not only a relationship between queues and magazines, but also the number of shafts by which the spaces are related. For example, consider a conventional ele-

https://doi.org/10.25088/ComplexSystems.22.3.247

vator system with carriages fixed in shafts consisting of two shafts, two queues, and three magazines with the 0^{th}-order SQ and SM incidence matrices defined in Figure 16. The resulting 1^{st}-order QM matrix is therefore calculated in Figure 16. For the SQ matrix, the first shaft is connected only to the first queue while the second shaft is connected to both queues. For the SM matrix, the first shaft is connected to all three magazines but the second shaft is connected only to the third magazine. The QM matrix is determined from the SQ and SM matrices. The connectivity of the first shaft results in many of the connections between the first queue and all magazines. Since the first shaft is not connected to the second queue and the second shaft is only connected to the third magazine, there is no connection between the second queue and the first and second magazines.

$$(SQ) = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \qquad (SM) = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

$$(QM) = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix}$$

Figure 16. Incident matrices describing the relationship of shafts, queues, and magazines.

In descriptions of these systems, information regarding the directionality of shafts is included as a shaft direction vector (SDV) and the 0^{th}-order directed incidence matrices are determined by row-by-row scalar multiplication of each element in the shaft direction vector (or 1 minus the element) with the SQ and SM matrices, as shown in equations (13) through (16):

$$\overrightarrow{SQ}_{\text{up}_i} = SDV_i \, \overrightarrow{SQ}_i \qquad \{i \mid i = 1 \ldots s\} \tag{13}$$

$$\overrightarrow{SQ}_{\text{down}_i} = (1 - SDV_i) \, \overrightarrow{SQ}_i \qquad \{i \mid i = 1 \ldots s\} \tag{14}$$

$$\overrightarrow{SM}_{\text{up}_i} = SDV_i \, \overrightarrow{SM}_i \qquad \{i \mid i = 1 \ldots s\} \tag{15}$$

$$\overrightarrow{SM}_{\text{down}_i} = (1 - SDV_i) \, \overrightarrow{SM}_i \qquad \{i \mid i = 1 \ldots s\}. \tag{16}$$

For a system with q queues, s shafts, and m magazines (base q-s-m), there are 2^{sq} SQ incidence matrices and 2^{sm} SM incidence matrices, for a total of $2^{sq} .2^{sm} = 2^{s(q+m)}$ possible physical configurations. In these possible configurations, however, there are invalid configurations such as all 0 entries of both SQ and SM matrices, which

https://doi.org/10.25088/ComplexSystems.22.3.247

are a null system. Valid configurations are based on the following rules.

1. All shafts must be associated with at least one queue and magazine (no all-zero rows in the *SQ* or *SM* matrices).
2. All queues and magazines must communicate with at least one shaft (no all-zero columns in the *SQ* or *SM* matrices). For a unidirectional shaft, all queues and magazines must communicate with at least one up-shaft and one down-shaft.
3. *SM* matrices must be in the lowest energy state with respect to shafts and magazines.
4. *SQ* matrices must be in the lowest energy state with respect to queues.
5. Any rows in the *SQ* matrix corresponding to repeated rows in *SM* must be in the lowest energy state.

Operational logic of the elevator system is the set of rules that are applied to evolve a system from its initial state to a final state. In terms of discrete event simulation, the operational logic determines the evolution of states.

There are two categories of decision logic that are used in the elevator system simulations.

1. General operational logic is common to all elevator systems and represents the actual state changes in an evolution. For instance, a carriage goes up if its destination is above its current position.
2. Specific operational logic controls the timing of general logic. Then, it defines the operational behavior of the system. For instance, the two primary operational logic parameters that relate to the operation of carriages and hatches are the serial/parallel operation of carriages and doors/hatches, and interlock logic.

The input stream in a weapons elevator system is defined to be all of the elements to be transported by the system. These elements can be considered as individual items, or a load of items with a common destination. The four major characteristics defining an input stream are the number, type, and order of items in the queues and the arrival rate of items throughout an evolution. In this paper, queues are assumed to contain all items at the start of an evolution, so the interarrival rate of items is zero.

The number of items in a typical input stream is fairly large. The possible permutations of items within a single queue are also extremely large, following $n!$ for n distinct items. However, while there exists a large variety of types of ordnance stored on an aircraft carrier, each item is ordinarily not distinct, as multiple items of the same type are typically found. Since the order of the identical items is therefore not important, the total number of possible input streams is significantly reduced.

https://doi.org/10.25088/ComplexSystems.22.3.247

Compatibility between certain types of ordnance is another factor that results in grouping of identical items in queues. These constraints result in the ability to make generalized statements about the contents or arrangements of queues, reducing the total number of possible arrangements of items in queues.

The ordering of items assumes only one item is available at a time and items are serially treated. This assumption significantly increases the design space of the input streams and is associated with the factorial terms. On the other hand, the factorial terms disappear when the queues are assumed to be parallel, in which a carriage can access all items in queues. The parallel queues, however, are not unrealistic if a queue is composed of a limited number of item types. In a parallel queue, items/destinations are selected by the carriage based on some decision logic intended, but not guaranteed, to minimize total operation time. The order of items selected in a parallel queue defines a near upper bound on performance for the given configuration. How close the resulting performance is to the actual upper bound is dependent on the decision logic (heuristic) used. If a serial queue is ordered using a heuristic applied in a parallel queue instead of random selection of one of the $n!$ item arrangements, system performance is identical to the performance of a parallel queue system with the same set of items. This concept also yields an arguably better performance than random selection.

The common motives for the number of assumptions used in the simulation model are to simplify the search space while offering complete characterization of the ranges of behavior and performance. These assumptions occur with respect to physical attributes, operational logic, and input streams.

Our goal is not to evaluate the performance of the actual system toward its optimization but to identify the relationships between behavior, performance, and robustness. Therefore, the vertical distance between each queue and magazine is identical and actual distances between spaces are not specified. Only the connectivity between spaces is considered.

In another attempt to limit the design space, we assume a set of queues based on a predetermined ratio of item types. Additionally, all queues contain identical item distributions at the start of a given simulation. We are interested in complete characterization of how the elevator systems respond in variable environments. By varying item ratios systematically, however, each configuration is bound to experience queue distributions that both are and are not suited to its specific connectivity, which could be thought of as varying the level of perturbation in input streams from ideal sets. Identical queues of systematically varied item distributions are therefore intended to offer complete characterizations of the range of behavior and performance while limiting the set of required simulations to tractable levels.

An encoding process, analogous to the encoding of evolution rules in cellular automata, is used to systematically define and identify alternatives. Encoding expresses the configuration in the compact form of a decimal number rather than a cumbersome description. A configuration code is based on the system size, or base value, which is the number of queues, shafts, and magazines. The total number of bits b required to describe the physical configuration is equal to the sum of the product of the number of queues and shafts and the product of the number of magazines and shafts, equal to the total number of entries in the *SQ* and *SM* incidence matrices. This relationship is shown in equation (17). The maximum decimal value for b bits is $2^b - 1$, the maximum number of distinct configurations corresponding to a given base:

$$b = s.q + s.m = s(q + m). \tag{17}$$

Valid configurations are determined by creating and testing all configurations corresponding to codes 0 to $2^b - 1$. Configurations are constructed from a decimal value by first converting the decimal value to a binary digit sequence. To derive *SQ* and *SM* matrices, the digit sequence is partitioned into two bytes, with lengths equal to the number of entries in the *SQ* and *SM* matrices. The first byte is $s.q$ long and is comprised of the most significant bits. For binary code 01110111, the partitioning process is illustrated in Figure 17.

$$01\,110\,111 \to (011\,101)\,(11) \qquad 011\,101 \to (011)\,(101) \to \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

(a) (b)

$$11 \to (1)\,(1) \to \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

(c)

Figure 17. (a) Partitioning code 119 into *SQ* and *SM* bytes. (b) Partitioning *SQ* bytes to form rows in the *SQ* matrix. (c) Partitioning *SM* bytes to form rows in the *SM* matrix.

For all coded configuration descriptions, the code number and the base system are provided along with a description of the operational logic used in a binary format. We let serial shaft operations equal 0, parallel shaft operations equal 1, the lack of interlocks equal 0, and the use of interlocks equal 1. The distribution of item types may also be included if a description of the input stream is required. An example is shown in Figure 18.

$\boxed{119 \quad 3-2-1 \quad 0-1 \quad (100)}$

Figure 18. The code description of system 119, consisting of three queues, two shafts, and one magazine. The system has serial shaft operations and uses interlocks. All of the items in the queue are bound for the single magazine.

Visualization techniques are used to describe the performance, since a numerical representation does not necessarily fully characterize the behavior of the evolution and provide a means for involving subjective human perceptive abilities. For long evolutions, explicit state representations, especially in terms of verbal descriptions, are practically impossible.

There are several visualization techniques that can be used to present evolutions. We primarily will use evolution histories, with respect to both temporal and logical evolution lengths, of both system states and the states of individual carriages. However, for completeness, we also discuss cellular evolutions and state evolution trajectories. These techniques are not used extensively, although they have unique characteristics that may be advantageous for particular conditions.

Cellular representations are similar to the cellular automata evolutions but the elevator system is inhomogeneous, with cells representing distinct system attributes. A row in a cellular representation represents the system state at a given instant, and a column of cells represents the history of a single system attribute with time advancing down the page. The width of the evolution is proportional to the number of attributes required to completely describe the system and increases with larger system sizes. For consistency, all cellular representations are composed of only the attribute listed in Table 2 along with a description of their possible states. All of the system states are binary variables and are represented by a colored cell (black = true, white = false). For long evolutions, the cellular representation is physically cumbersome, while evolution trajectories address this space issue by illustrating behavior patterns.

Evolution trajectories are visualized by constructing a list of coordinates of the form (s_{t-1}, s_t) for $\left(t \,\middle|\, 1 \leq t \leq t_f\right)$ where s_t is the decimal representation of the state of the system at time step t and $s_{t=0}$ is the initial state of the system. Figure 19 presents an evolution corresponding to a 1-1-1 base conventional system for 100 time steps. Since there is only one item type, one source of items, and one carriage, the evolution is repetitive and will follow the same evolution trajectory regardless of the number of items or length of evolution.

https://doi.org/10.25088/ComplexSystems.22.3.247

Attribute	Possible States
Carriage location	integer value
Carriage movement	true / false
Destination type	queue / magazine
Direction	true / false
Carriage loading	true / false
Carriage unloading	true / false
Upper ballistic hatch opening	true / false
Upper ballistic hatch open	true / false
Upper ballistic hatch closing	true / false
Upper ballistic hatch closed	true / false
Lower ballistic hatch opening	true / false
Lower ballistic hatch open	true / false
Lower ballistic hatch closing	true / false
Lower ballistic hatch closed	true / false
Magazine door opening	true / false
Magazine door open	true / false
Magazine door closing	true / false
Magazine door closed	true / false

Table 2. System attributes included in cellular representation visualization techniques.

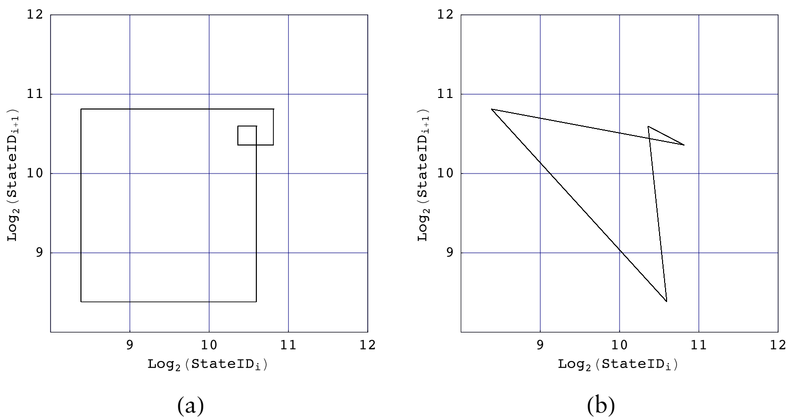


Figure 19. Evolution trajectory for the simple 1-1-1 size system. There are only four states entered by the single carriage. (a) In the evolution trajectory using temporal evolution, repeated states are encountered, so all edges are parallel. (b) Each step in the logical trajectory corresponds to a state change.

https://doi.org/10.25088/ComplexSystems.22.3.247

Evolution histories are used to retain the information associated with the sequencing of states missing in evolution trajectories. These histories are similar to cellular evolutions, except states are represented as encoded values. As with evolution trajectories, evolution histories can be presented with respect to both the temporal evolution length and the logical evolution length. Additionally, evolution histories can be presented in terms of both system states and individual carriage states. The number of possible states that define valid history or trajectory points is dependent on the system size and the arbitrary level of detail used to describe the system. The finest detail considers all system states, including timers and item locations, and no states are ever identical. In conventional systems, the carriage attributes used to define a carriage state are presented in Table 3, along with the number of bits required to describe each attribute.

Attribute	Number of Bits
Global destination type	1
Global destination number	Max(Ceiling($\log_2 q$), Ceiling($\log_2 m$))
Current location type	2 (0 = queue, 1 = magazine, 2 = shaft)
Current location number	Max(Ceiling($\log_2 q$), Ceiling($\log_2 m$))
Loading	1
Unloading	1
Up direction	1
Down direction	1
Magazine reservation number	Ceiling($\log_2 m$)
Queue reservation number	Ceiling($\log_2 q$)
Carriage unsure of queue	1

Table 3. The carriage attribute used to define evolution states in conventional elevator systems.

These attributes are used to describe a carriage in five possible carriage states: carriage loading, moving down to a destination magazine, carriage unloading, moving up with a known destination, and moving up without a known destination, respectively. Combinations of these state variations in turn define system states, the validity of which is based upon the physical limitations and constraints imposed on the system. The attribute combinations defining each of the five states are presented in Table 4.

State 1	1 xx00xx1010xx000
State 2	1 xx10000001xx000
State 3	1 xx01xx010100000
State 4	0 xx1000001000xx0
State 5	0 xx1000001000001

Table 4. The five possible state descriptors of the attribute combinations in Table 3 and bit locations for a conventional system. Variable bits are indicated by x and can be filled by binary values corresponding to possible attribute values. The number of x symbols is not fixed and depends on the system size. Possible state variations are found by finding valid combinations of possible bytes indicated by x.

In virtual conveyor systems (unidirectional shafts), a carriage must wait to select an available queue before moving up from a magazine, since the selection of a queue depends on the selection of an appropriate shaft. The "carriage unsure of queue" in conventional systems (bidirectional shafts) seen in Table 3 is therefore not required in virtual conveyor systems. The virtual conveyor systems assume that a carriage travels in a down-shaft while loaded and up in a shaft when unloaded. Virtual conveyors therefore use a bit to describe if the carriage is loaded, which eliminates the direction bits used in conventional system state descriptions. The attributes used to describe carriage states in virtual conveyor systems are listed in Table 5, along with the number of bits required to describe each attribute.

Attribute	Number of Bits
Global destination type	1
Global destination number	Max(Ceiling($\log_2 q$), Ceiling($\log_2 m$))
Current location type	2 (0 = queue, 1 = magazine, 2 = upper shaft, 3 = lower shaft)
Current location number	Max(Ceiling($\log_2 q$), Ceiling($\log_2 m$))
Loading	1
Unloading	1
Magazine reservation number	Ceiling($\log_2 m$)
Queue reservation number	Ceiling($\log_2 q$)
Carriage loaded	1

Table 5. The carriage attribute used to define evolution states in virtual conveyor elevator systems.

https://doi.org/10.25088/ComplexSystems.22.3.247

With mobile carriages, it is possible to have multiple carriages in a single shaft, magazine, or queue. As there are multiple carriages in the same location, virtual conveyors are divided into zones, each with a maximum occupancy limit. The zones in a virtual conveyor are listed, along with a description of their boundaries or the state of the carriages within them in Table 6. The use of zones results in an increase in the carriage states. There are 11 carriage states, as shown in Table 7.

Queue	Contain carriages loading or waiting to load.
Queue to Shaft	All carriages are loaded. Bounded by the upper ballistic hatch of the destination shaft.
Upper Shaft	From the upper to lower ballistic hatch.
Lower Shaft	From the lower ballistic hatch to all magazine doors in that lower shaft.
Shaft to Magazine	Loaded carriages moving from the magazine door connected to the down-shaft to the magazine unloading area.
Magazine	Carriages are unloading or waiting to unload.
Magazine to Shaft	From the magazine unloading area to the selected up-shaft door. All carriages are unloaded and have selected their queue and shaft.

Table 6. Zone definitions for virtual conveyors. Zones are defined by physical boundaries of the state of carriages within them.

The qualitative descriptions of evolution histories are loosely related to the concept of algorithmic complexity. In a simple evolution, it is possible to describe the patterns that occur in a compressed form, while for the most complex trajectories, a description of the pattern is incompressible and explicit representation is required to describe the pattern without losing information.

8. Measure of Complexity for Weapons Elevator Simulations

As described in Section 7, both static and dynamic measures are used for the elevator systems. Static measures describe the potential complexity of the system, such as the total number of possible valid states the system can enter, the physical connectivity, and the logical connectivity of the system. Dynamic measures are based on actual complexity. Dynamic measures are therefore more accurate, but the accuracy also comes with cost, in this case, time/computational cost. Since the

https://doi.org/10.25088/ComplexSystems.22.3.247

State Number	State
1	Carriage loading
2	Moving from queue to shaft
3	Moving down in upper shaft
4	Moving down in lower shaft
5	Moving to magazine (includes entering magazine from shaft)
6	Unloading
7	Finished unloading and unsure of destination
8	Moving from magazine to shaft with known destination queue
9	Moving up in lower shaft
10	Moving up in upper shaft
11	Moving to queue (includes entering queue from shaft)

Table 7. The 11 possible carriage states used in a virtual conveyor elevator system.

complete design space of elevator systems is quite large, the use of static versus dynamic measures and relative accuracy are therefore of great significance. The static and dynamic measures used in this paper are presented in Table 8.

Static	Dynamic
Total number of possible states	Number of unique states used
Average physical connectivity	Fraction of states used
Fraction of potential connectivity	Logical complexity
Average logical connectivity	State complexity
	Compressed state complexity

Table 8. Static and dynamic measures of complexity specific to elevator systems.

The total number of possible states is based on the counting of states as an estimate of complexity used by Bar-Yam [14]. When used as a static measure, all potential system states are counted, rather than the number of states a system exhibits during an evolution. The number of carriages in the system has a significant impact on the number of possible states, as shown in equation (18):

$$\text{total valid states} \leq I^n \tag{18}$$

https://doi.org/10.25088/ComplexSystems.22.3.247

where I is the number of possible states for an individual carriage and n is the number of carriages in the system.

Average physical connectivity is associated with the λ parameter, described in another format with connectivity by Kauffman [12]. Kauffman concludes that, for systems with low connectivity with one or two connections, behavior can only be trivial or simple. For a system with complete connectivity, behavior is almost always chaotic, except for the simplest of initial conditions. In between, complex behavior is found with the proper tuning of the λ parameter and the system connectivity. Average physical connectivity provides an indication of system architecture.

Physical connectivity is calculated from SQ and SM incidence matrices and measures the accessibility between spaces. The expression used to calculate average physical connectivity is presented in equation (19):

$$\text{average physical connectivity} = \frac{2\left(\sum_{i=1}^{s}\sum_{j=1}^{q} SQ_{ij} + \sum_{i=1}^{s}\sum_{k=1}^{m} SM_{ik}\right)}{s+q+m} = \frac{2\sum_{i=1}^{s}\left(\sum_{j=1}^{q} SQ_{ij} + \sum_{k=1}^{m} SM_{ik}\right)}{s+q+m}. \tag{19}$$

The fraction of potential connectivity represents the normalization of the average physical connectivity with respect to the maximum average physical connectivity for a system with the same number of physical spaces. Equations (20) and (21) present the maximum average physical connectivity and the fraction of physical connectivity, the ratio of the average connectivity to the maximum average physical connectivity, respectively:

$$\text{maximum average physical connectivity} = \frac{2\,(sq+sm)}{s+q+m} \tag{20}$$

$$\text{fraction of potential connectivity} = \frac{2\sum_{i=1}^{s}\left(\sum_{j=1}^{q} SQ_{ij} + \sum_{k=1}^{m} SM_{ik}\right)}{sq+sm}. \tag{21}$$

The average logical connectivity describes the amount of information comprising an evolution rule associated with a system attribute. In an inhomogeneous system like the elevator system, where the system attributes can be thought of as analogous to the cells of a cellular automata network, the amount of information required to evolve system attributes is variable and connections to a single attribute may be

https://doi.org/10.25088/ComplexSystems.22.3.247

distributed over a wide range of the attribute network, not restricted to a local neighborhood.

Logical connectivity is determined by searching through the simulation code and identifying the number of variables that are involved in the decision logic for each attribute. Since different evolution logic may be used depending on the control logic, the values of logical connectivity may vary.

The expressions for the logical connectivities for conventional and virtual conveyors for the various control logics are presented in Table 9, where l is the number of items carried per carriage load. The expressions in Table 9 indicate that the logical connectivity is constant for a fixed system size and control logic, regardless of any physical or structural differences among the systems of that size. In this respect, the average logical connectivity is analogous to the measure of the maximum average physical connectivity of a system, measuring the potential complexity of a system size.

In addition to the operational logic, the values of the logical connectivities are dependent on the system size, which illustrates that more entities affect the evolution of an attribute in larger systems, increasing the complexity of the system toward chaos. Furthermore, the expressions and values of average logical connectivity are also dependent on the evolution logic and number of attributes considered.

A dynamic measure of logical connectivity is costly in large simulations. It is therefore not considered in this paper.

Operational Logic	Average Logical Connectivity
Conventional, serial, interlocks	$\frac{335+59\,n+l+68\,m+10\,q}{42}$
Conventional, serial, no interlocks	$\frac{329+59\,n+l+38\,m+10\,q}{42}$
Conventional, parallel, interlocks	$\frac{296+59\,n+l+51\,m+10\,q}{42}$
Conventional, parallel, no interlocks	$\frac{290+59\,n+l+22\,m+10\,q}{42}$
Virtual conveyor, serial, interlocks	$\frac{507+346\,n+53\,m+6\,q+50\,s}{58}$
Virtual conveyor, serial, no interlocks	$\frac{491+331\,n+20\,m+6\,q+44\,s}{58}$
Virtual conveyor, parallel, interlocks	$\frac{476+211\,n+8\,m+6\,q+44\,s}{58}$
Virtual conveyor, parallel, no interlocks	$\frac{507+346\,n+53\,m+6\,q+50\,s}{58}$

Table 9. The average logical connectivity for tested operational logics. The logical connectivity is dependent on system size and the operational logic employed.

https://doi.org/10.25088/ComplexSystems.22.3.247

The number of unique states used is based on the complexity estimation technique of Bar-Yam [14], and the number of states visited in an evolution is the dynamic equivalent to the static measure of the total number of possible states comprising an evolution state space. The measure of the number of visited states is dependent on the scale of the definition of the attributes used to define system states.

For a given evolution, the finer the resolution, the greater the number of unique states found in an evolution. A greater number of visited states is an indication of greater complexity.

The fraction of states used is a mixture of dynamic and static complexity measures. However, since explicit simulation is required, it is defined as a dynamic measure. The expression for the fraction of states used is presented in equation (22):

$$\text{fraction of states used} = \frac{\text{number of unique states used}}{\text{total number of possible states}}. \tag{22}$$

A comparison of the actual space explored to the potential space indicates how the actual evolution “lives up” to its potential. The comparison also validates the use of the total number of possible states as a potential measure of complexity.

The logical complexity/logical compression is a normalized complexity measure enabling comparison of evolutions of systems of different sizes. The logical complexity C_{x_L} describes the amount of information required to express the logical sequence of states and is defined in equation (23):

$$C_{x_L} = \frac{\text{logical evolution length}}{\text{temporal evolution length}}. \tag{23}$$

Logical compression describes the amount of redundant information in an evolution with respect to the logical sequence of states. The logical compression C_{m_L} is defined in equation (24):

$$C_{m_L} = 1 - \frac{\text{logical evolution length}}{\text{temporal evolution length}} = 1 - C_{x_L}. \tag{24}$$

The logical evolution length is defined as the length of an evolution logically equivalent to an evolution history but with at least one system attribute change per evolution step. The logical evolution therefore does not change the sequencing of states, but removes the temporal dependence. A logical evolution by definition is always less than or equal to the equivalent temporal evolution length, and the logic complexity and compression therefore range from 0 to 1.

Logical complexity/compression is based fundamentally on the measure of algorithmic complexity. Algorithmic complexity is greatest when the algorithm is incompressible. An evolution with repeated states can be compressed by substituting the explicit state representations with a description of the number of repetitions. The information compression for a single repeated state follows the expression in equation (25) for a state composed of b bits with p repetitions:

$$1 - \frac{b+p}{bp}. \tag{25}$$

For states composed of a large number of attributes, the additional information required to describe the number of repetitions in the compressed form becomes negligible, and equation (25) reduces to equation (26), which essentially states that any information additional to the declaration of the first state is redundant:

$$1 - \frac{1}{b}. \tag{26}$$

State complexity/state compression is another application of algorithmic complexity that accounts for global patterns and local patterns. The state complexity identifies the number of unique states in an evolution, indicating the information necessary to represent both runs of identical states and possible global sequences. The number of unique states is normalized with respect to the temporal evolution length shown in equation (27), where the state complexity is denoted by C_{x_S}:

$$C_{x_S} = \frac{\text{number of unique states used}}{\text{temporal evolution length}}. \tag{27}$$

The state compression C_{m_S} describes the amount of redundant information in a temporal evolution when the local and global patterns are taken into account, and is related to the state complexity as in equation (28):

$$C_{m_S} = 1 - \frac{\text{number of unique states used}}{\text{temporal evolution length}} = 1 - C_{x_S}. \tag{28}$$

Following the definition of algorithmic complexity, maximum complexity occurs when there is no redundant information and each state in the temporal evolution is unique, that is, $\left(C_{x_S} = 1,\ C_{m_S} = 0\right)$.

The number of unique states visited is normalized by the length of the temporal evolution to obtain a complexity measure corresponding to the actual evolution.

https://doi.org/10.25088/ComplexSystems.22.3.247

The compressed state complexity/compressed state compression is identical to the state complexity, except that the number of unique states visited in an evolution is normalized by the length of the logical evolution, rather than the length of the temporal evolution. The expression for the compressed state complexity C_{x_C} is given in equation (29):

$$C_{x_C} = \frac{\text{number of unique states used}}{\text{logical evolution length}}. \tag{29}$$

The compressed state compression C_{m_C} describes the amount of redundant information in the logical evolution, related to compressed state complexity by the relationship in equation (30):

$$C_{m_C} = 1 - \frac{\text{number of unique states used}}{\text{logical evolution length}} = 1 - C_{x_C}. \tag{30}$$

The compressed state complexity is related to the logical complexity and the state complexity by the expression in equation (31). The relation in equation (31) indicates that the compressed state complexity is the ratio of the information required to account for local and global patterns to the information required to account for local patterns only. The amounts of compression are not related in this manner, and the compressed state compression does not represent the excess information describing repetitions of global patterns:

$$C_{x_C} = \frac{C_{x_S}}{C_{x_L}}. \tag{31}$$

In the case of longer evolutions resulting from a greater number of repetitions of a single sequence of states, the relative values of the fraction of unique visited states that a set of repeating states comprises, the number of states in the set of repeated states, the number of repetitions of the set, and the original logical evolution length collectively affect the complexity. The compressed state complexity is generalized in equation (32), where u is the number of unique states visited in the evolution, excluding the states comprising the set of repeating states x. The original number of logical evolution steps T_L increases with each additional repetition p of the x states:

$$C_{x_C} = \frac{u + x}{T_L + p x} \qquad \text{where } u + x \leq T_L. \tag{32}$$

https://doi.org/10.25088/ComplexSystems.22.3.247

9. Logical Complexity

9.1 Simple Conventional System

As described in Section 8, logical complexity is defined as the ratio of the length of the logical evolution to the temporal evolution length and is intended to identify the level of logical activity in evolution. We defined the baseline scenario as the 1-1-1 system used for defining target operation cycle times. Figure 20 presents the relationship between the logical complexity and throughput for all conventional SIL systems. Throughput is in units of items per minute, where one item per carriage is carried per trip. The correlation between complexity and throughput for all evolutions is 0.921, indicating a relatively strong relationship. This measure of throughput is used throughout this paper. But, as shown in Figure 20, the correlation between complexity and throughput is apparently meaningless. For a large range of logical complexities, there also exists a large range of throughputs. The general trend resulting from the common skew of the left and right boundaries, however, is that more logically active evolutions have increased performance.

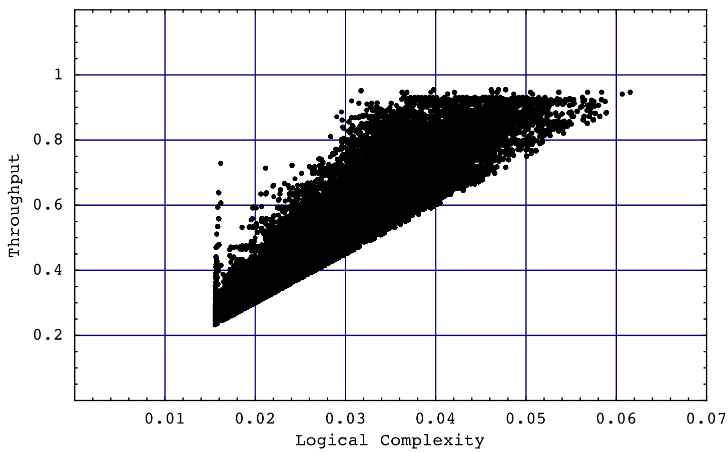


Figure 20. The distribution of all evolutions for conventional SIL systems with respect to logical complexity and throughput; 546 919 data points are represented. The variables have a correlation of 0.921.

Our analysis begins with the simplest of systems, 1-1-1. The throughput and the logical complexity are predictable and both the values are minimal when compared to Figure 1. Thus we should expect the complexity to be minimal. Figure 21 presents a data point of a 1-1-1 system.

The compressed state trajectory in Figure 22 shows the system cycling through four states, and the compressed and full state histories presented in Figure 23 identify the same pattern—a simple repetitive cycle through four states—loading, traveling, unloading, and traveling back.

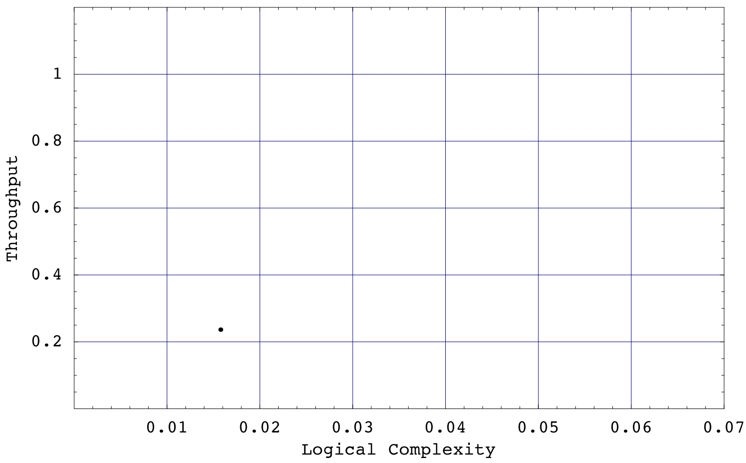


Figure 21. A single 1-1-1 configuration and evolution. The logical complexity and throughput are both minimal values for all evolutions.

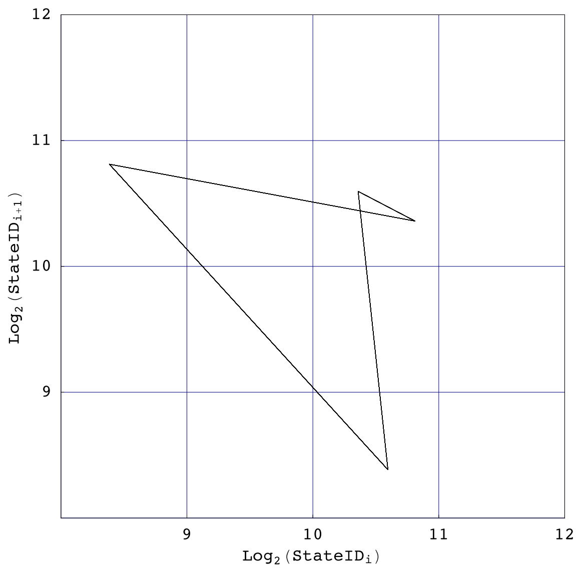


Figure 22. The compressed state trajectory for configuration 3 in 1-1-1 evolution shows a simple cycle through four states.

https://doi.org/10.25088/ComplexSystems.22.3.247

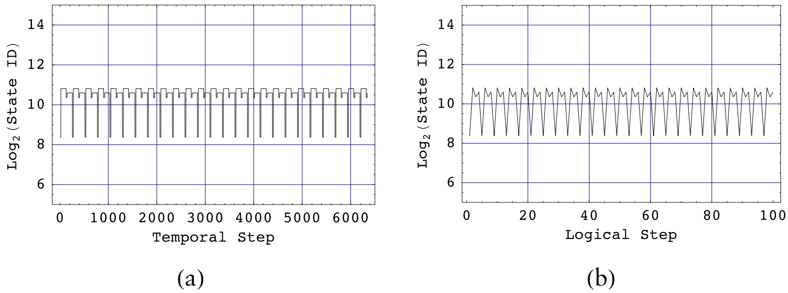


Figure 23. The full and compressed state histories for the configuration 3 in 1-1-1 evolution. These histories also identify a simple cycle through four unique states.

Larger systems with a single carriage all behave like a 1-1-1 system and theoretically have the same throughput and logical complexity. The state trajectories and state histories may appear slightly more complex in this case, however. There possibly exist different cycles due to the carriage's visiting different magazine numbers, but the evolution is characteristically simple and has the same throughput as the evolution with a single magazine. The logical complexity is also identical, as it does not identify unique patterns in an evolution, only the state change.

In systems with more than one carriage, variations in connectivity are possible and a relationship between logical complexity and throughput begins to develop. Figure 24 shows the correlation between logical complexity and throughput for all 2-2-2 systems, which has 66 valid evolutions. Since some evolutions have identical throughput and logical complexity, the three-dimensional frequency landscape is employed to present a truer picture of the performance/behavior relationship.

9.2 Evolution Sets

The sets of evolutions 2-2-2 that have been presented have consisted of all nonhalting evolutions. Halting evolutions are defined as those in which the system logic freezes a carriage in place, leaving deliverable items undelivered. This condition is distinct from an evolution with incomplete item transfer, where undelivered items are undeliverable because of absent connections between queues and item destinations.

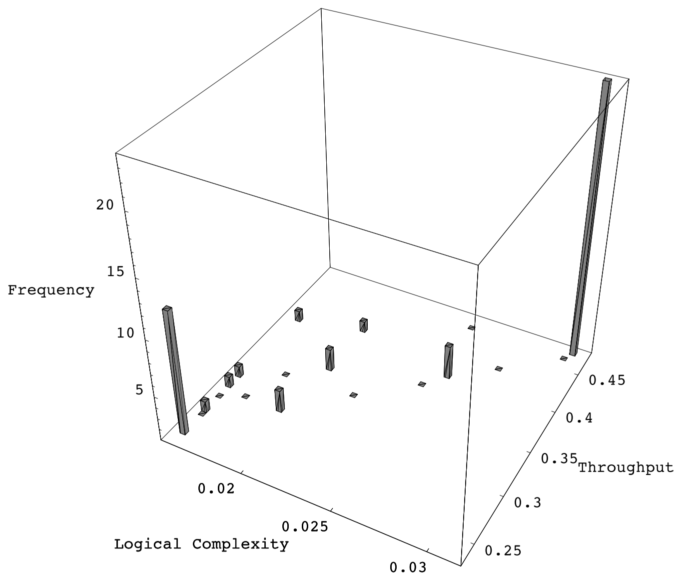


Figure 24. The frequency landscape for all 2-2-2 evolutions. The various frequencies indicate the presence of close-to-identical complexity and throughput values.

In an optimal search, the nonhalting and complete evolution sets are larger than they should be. If an evolution of a physical configuration with one queue distribution fails, then all evolutions of that configuration should be removed from the set. The remaining configurations, however, are not always necessarily robust. Our measure of robustness is based on the set of queue distributions considered in the evolutions. Not halting or complete delivery for these queue distributions is necessary, but not sufficient for inclusion of a configuration into the respective sets. For many configurations, it is undecidable whether any configuration will ever result in a halting or incomplete evolution for any set of queue distributions without explicit evolution of the configuration for all of those queue distributions. Since the complete set of queue distributions has been shown to be quite large, a complete demonstration of robustness is practically impossible. The set of queue distributions used is only intended to represent a significant variety and serve as a go/no go test for robustness.

The decision to include or exclude configurations for which at least one evolution fails with respect to halting or complete delivery is based primarily on what we are attempting to characterize. While we are generally interested in characterization of the full range of behaviors, characterization of a robust set is important, as are refining the set and indicating the requirements for consistent inclusion in the set. They also help establish a correlation between performance and complexity—generalizations are possible from the requirements for robustness and the aggregate performance and complexity.

A comparison of the values and distribution of throughput and complexity of sets of evolution of varying degrees of robustness with respect to completeness with all nonhalting and all complete evolutions in 2-2-2 configurations yields some initial correlations. As a result shown in Figure 25, the correlation of logical complexity and throughput for the most robust configurations of 0.997 is greater than the correlation for all nonhalting evolutions of 0.931. The mean for the most robust configurations is greater than the mean for all evolutions (0.0267 vs. 0.0235), as is the mean throughput (0.407 vs. 0.375). Since the most robust and adaptable configurations have a higher mean complexity, adaptability is then directly related to complexity. Similarly, the most adaptable configurations are also the best performers.

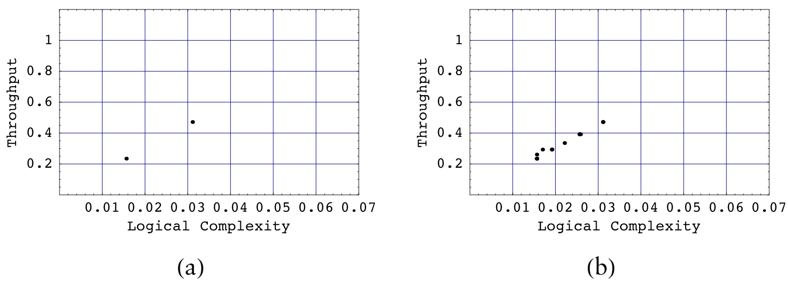


Figure 25. The distributions of 2-2-2 evolutions with (a) one complete evolution and (b) six complete evolutions (most robust configurations).

Any general statement as in Section 9.2 is tentative since these results are drawn only from a 2-2-2 configuration set, and the correlation and mean values depend on the distribution of evolution sets, which may vary relatively for evolution sets with different sizes.

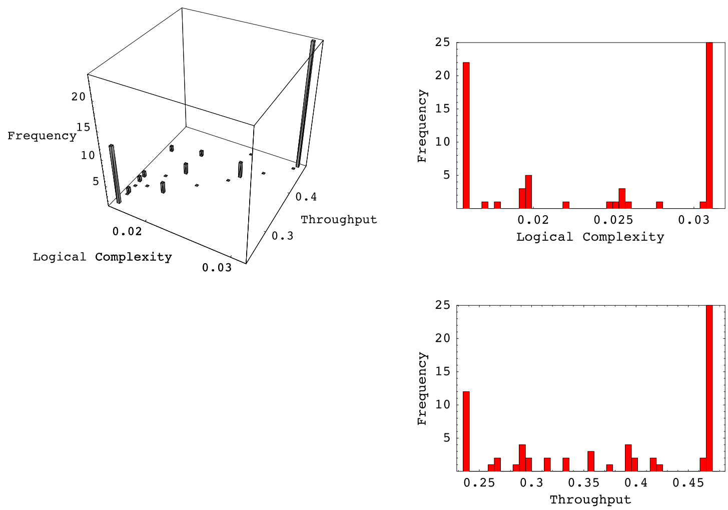
Since the differences between the distributions of different sets of evolutions is not completely apparent in a comparison of two dimensions, three-dimensional frequency landscapes and histograms of cross sections of complexity and throughput presented in Figure 26 are used for comparisons.

As in Figure 26, a comparison of the frequency landscapes and histograms of the complete evolutions and the evolutions of the most robust configurations with the set of nonhalting evolutions further illustrates the differences in the distributions that lead to higher aggregate throughput and complexity for the most robust configurations.

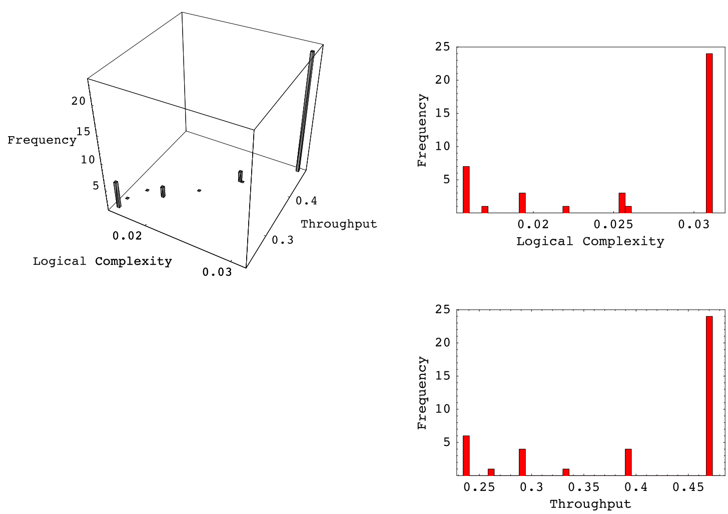
https://doi.org/10.25088/ComplexSystems.22.3.247

Although the absolute number of high-complexity evolutions is lower for most robust configurations, the ratio of the highest- to lowest-complexity evolutions increases substantially when the logically simpler evolutions are removed. The differences in the sets of evolutions have a similar, but lessened, effect on throughput.

(a)



(b)



https://doi.org/10.25088/ComplexSystems.22.3.247

(c)

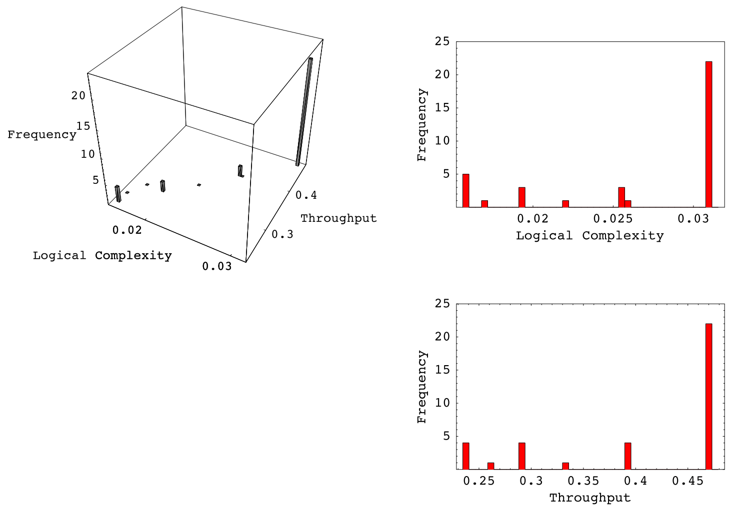


Figure 26. The three-dimensional distribution and the histograms of the evolutions for (a) nonhalting evolutions, (b) complete evolutions, and (c) the evolutions of the most robust 2-2-2 configurations.

One of the most apparent features of the histograms is that none of them are normal. However, the use of the mean implies that we are dealing with normally distributed data. Therefore, characterization of the distributions using their mean values has a lower significance. The mean does characterize the differences between various subsets of evolutions, like the complete and robust evolution sets. Based on the unique characteristics of sets, it is possible to make general statements regarding complexity and throughput with the mean.

9.3 Mimicry

Mimicry results from a combination of the physical connectivity and the queue distribution. We can think of these two forms as mimicry with respect to carriages and mimicry with respect to magazines. There is no mimicry with respect to queues because all queues must contain some set of items and be connected to at least one shaft in a valid configuration, so all queues specified in a configuration description are always used.

A requirement for mimicry is a queue distribution with at least one item type absent. If the physical connectivity of the configuration is such that all carriages are utilized to some extent and simply do not visit the magazines corresponding to the absent item types, mimicry is defined to be with respect to magazines. If there is at least one carriage connected only to the magazines for which there are no items in

https://doi.org/10.25088/ComplexSystems.22.3.247

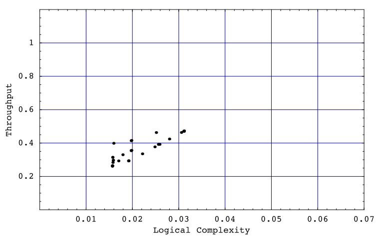
the queues, mimicry is defined to be with respect to carriages. In these configurations, the combination of the physical connectivity and queue distribution keeps the carriage(s) idle, and the configuration mimics a configuration with as many magazines as nonzero item types and as many shafts as nonzero shaft utilizations. Mimics with respect to carriages result in evolutions with both complete item transfer and incomplete item transfer. The set of all nonhalting evolutions and the set of complete evolutions can therefore both contain mimics.

From the unique sets and comparisons of subsets of the unique sets, correlations between complexity and performance are established that are truer than the correlations made from the sets that include mimics because of the elimination of redundancies. Therefore, the set of the evolutions of the most robust and unique configurations is created to establish correlations between complexity and throughput and to identify trends applicable in an optimal search.

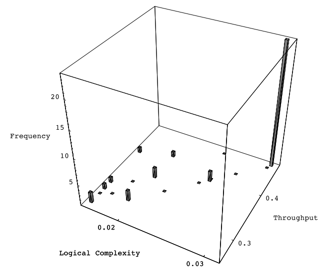
The two-dimensional distribution of nonhalting evolutions unique to 2-2-2 size systems, shown in Figure 27(a), does not look significantly different than the distribution of all nonhalting evolutions. Much of the similarity results from the limited number of mimics with respect to carriages—there are just 12 mimics out of a total of 66 evolutions. The frequency landscapes, however, are different, with many of the low complexity/low throughput evolutions presented in Figure 26(a) for the set of all nonhalting evolutions absent in the set of unique nonhalting evolutions in Figure 27(b).

There are 54 unique nonhalting 2-2-2 evolutions, meaning 12 of the 66 2-2-2 evolutions are equivalent to smaller systems. All 12 evolutions are of low logical complexity and low throughput, but represent a mixture of complete and incomplete evolutions. Since the logical complexity and throughput are minimal for all mimics, the mean logical complexity and throughput for the set of unique, nonhalting evolutions are both greater than for the set of all nonhalting evolutions. The mean logical complexity for the unique evolutions is 0.0252 and 0.0235 for all nonhalting evolutions, while the mean throughput for the unique evolutions is 0.406 and 0.375 for all nonhalting evolutions. A higher mean logical complexity and throughput for unique systems imply that increases in the number of carriages result in an ability to support additional complexity in behavior. Additionally, since mean throughput also increases, we can suggest a correlation between logical complexity and throughput—greater complexity supports greater performance. Once again, these conclusions are tentative, since they are based on small systems with few evolutions. However, they identify trends to examine with the results of larger systems.

The set of unique nonhalting evolutions containing incomplete evolutions may be considered as misrepresenting the true nature of the relationship between complexity and performance. Removal of these evolutions affects the mean logical complexity and throughput. The distribution appears to be identical to the distribution of complete evolutions shown in Figure 28, which has 34 unique, complete 2-2-2 evolutions. There are, however, six different evolutions between unique complete 2-2-2 evolutions and complete evolutions, appearing at the lowest logical complexity and throughput, which cause the slight difference of the mean logical complexity and throughput between both evolutions (0.0283 vs. 0.0264 for logical complexity and 0.431 vs. 0.402 for throughput).



(a)



(b)

Figure 27. (a) The two-dimensional distribution of nonhalting evolutions unique to 2-2-2 size systems. (b) The frequency landscape of unique 2-2-2 evolutions.

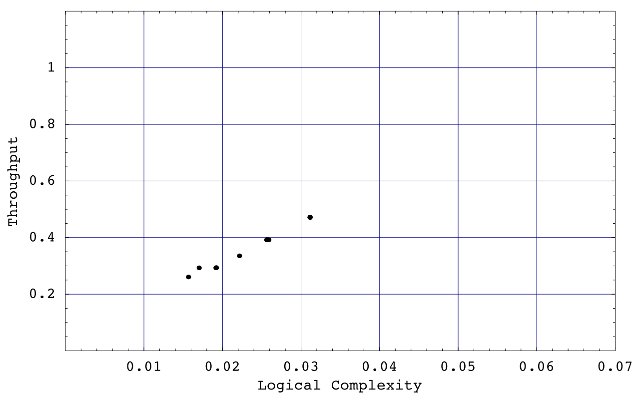


Figure 28. The two-dimensional distribution of unique, complete 2-2-2 evolutions is nearly identical to the distribution of all complete evolutions primarily because the sets differ by only six evolutions. The six evolutions have low logical complexity and throughput, resulting in a greater mean logical complexity and throughput for the unique evolutions.

https://doi.org/10.25088/ComplexSystems.22.3.247

The set of unique, robust evolutions is a subset of unique, complete evolutions and contains only the evolutions of configurations that completely deliver all queue distributions. The two-dimensional distribution of the evolutions of robust configurations is shown in Figure 29. The 18 evolutions from the three configurations that populate this set share the same logical complexity of 0.0311 and throughput of 0.471, which correspond to the maximum values for the set of complete evolutions. Since all logical complexities and throughputs are identical and maximal, the mean logical complexity and throughput are greater than the means for both unique, nonhalting evolutions and unique, complete evolutions. If we associate adaptability with robustness, then the most adaptable configurations are also the most logically complex and exhibit the greatest throughput.

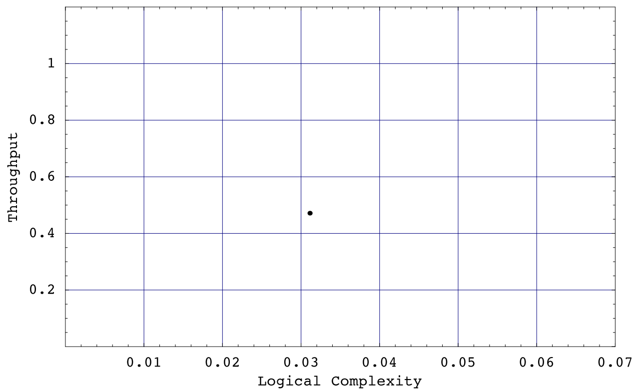


Figure 29. The two-dimensional distribution of the evolutions of the unique and robust 2-2-2 configurations.

9.4 Larger Systems

Subtle differences occur to the correlation between the logical complexity and throughput with increases in system size, which are evident in both the sets of all evolutions and unique evolutions. Tables 10 and 11 present the mean logical complexities and throughputs for the evolution subsets for all systems (size considered) and indicate the same trends.

https://doi.org/10.25088/ComplexSystems.22.3.247

System	N	C	R	UN	UC	UR	M
1-2-2	0.02574	0.02574	0.02574	0.02773	0.02773	0.03081	0.01577
1-2-3	0.02491	0.02491	0.02491	0.02695	0.02695	0.03081	0.01577
1-2-4	0.02475	0.02475	0.02475	0.02671	0.02671	0.03081	0.01577
2-2-2	0.02347	0.02639	0.02673	0.02519	0.02829	0.03112	0.0157
2-2-3	0.02332	0.02492	0.02531	0.02388	0.02701	0.03112	0.0157
2-3-2	0.02856	0.02917	0.02921	0.02981	0.03017	0.03231	0.02457
2-3-3	0.02742	0.02841	0.02884	0.02951	0.02994	0.03261	0.02285
2-3-4	0.02718	0.02820	0.02893	0.02943	0.02992	0.03276	0.02250
2-4-2	0.03357	0.03416	0.03446	0.03541	0.03614	0.04216	0.02874
2-4-3	0.03249	0.03315	0.03391	0.03540	0.03594	0.04291	0.02812
3-2-2	0.02170	0.02448	0.02492	0.02306	0.02581	0.02700	0.01565
3-3-2	0.02775	0.02968	0.02983	0.02927	0.03174	0.03657	0.02306
3-4-2	0.03194	0.03309	0.03293	0.03317	0.03427	0.04016	0.02878
4-2-2	0.02105	0.02388	0.02457	0.02229	0.02498	0.02602	0.01562
4-2-3	0.02041	0.02288	0.02366	0.02159	0.02431	0.02602	0.01562
4-2-4	0.02027	0.02234	0.02304	0.02136	0.02385	0.02602	0.01562
4-3-2	0.02691	0.02915	0.02939	0.02828	0.03070	0.03362	0.02274
4-3-3	0.02562	0.02799	0.02860	0.02764	0.02985	0.03363	0.02131
4-4-2	0.03179	0.03427	0.03419	0.03331	0.03644	0.04496	0.02781

Table 10. The mean logical complexity for evolution subsets of different system size (N = nonhalting, C = complete, R = robust, UN = unique and nonhalting, UC = unique and complete, UR = unique and robust, M = mimics).

System	N	C	R	UN	UC	UR	M
1-2-2	0.3860	0.3860	0.3860	0.4159	0.4159	0.4621	0.2365
1-2-3	0.3736	0.3736	0.3736	0.4042	0.4042	0.4621	0.2365
1-2-4	0.3713	0.3713	0.3713	0.4006	0.4006	0.4621	0.2365
2-2-2	0.3748	0.4015	0.4069	0.4058	0.4310	0.4715	0.2355
2-2-3	0.3616	0.3800	0.3859	0.3913	0.4126	0.4715	0.2355
2-3-2	0.4849	0.4884	0.4915	0.5208	0.5195	0.5549	0.3704
2-3-3	0.4670	0.4713	0.4832	0.5172	0.5117	0.5464	0.3574
2-3-4	0.4653	0.4673	0.4852	0.5174	0.5102	0.5411	0.3563
2-4-2	0.5970	0.6061	0.6174	0.6447	0.6561	0.7336	0.4722
2-4-3	0.5780	0.5873	0.6091	0.6509	0.6583	0.7518	0.4686
3-2-2	0.3492	0.3710	0.3780	0.3751	0.3916	0.4078	0.2348
3-3-2	0.4796	0.5036	0.5102	0.5224	0.5550	0.6361	0.3479
3-4-2	0.5875	0.5992	0.6028	0.6300	0.6381	0.7065	0.4777
4-2-2	0.3407	0.3618	0.3713	0.3651	0.3782	0.3923	0.2343
4-2-3	0.3351	0.3468	0.3582	0.3599	0.3691	0.3923	0.2343
4-2-4	0.3347	0.3394	0.3492	0.3584	0.3630	0.3923	0.2343
4-3-2	0.4653	0.4923	0.5016	0.5054	0.5344	0.5810	0.3426
4-3-3	0.4456	0.4672	0.4827	0.4956	0.5151	0.5812	0.3395
4-4-2	0.5884	0.6188	0.6250	0.6381	0.6767	0.7910	0.4587

Table 11. The mean throughput for evolution subsets of different system size.

10. State Complexity

10.1 Simple System

We would like to analyze the simplest conventional system to determine how closely state complexity approaches the definition of algorithmic complexity and the usefulness of state complexity in optimization and establishing correlations between complexity and throughput. The state complexity is the ratio of the number of distinct states entered to the product of the number of items carried and the system cycle time. Since there is only one queue, one magazine, and one path in the simplest system, the carriage enters only four distinct states throughout the course of the evolution. For the deterministic cycle times used, each system cycle corresponds to 258 time steps, and the number of items in a queue at the start of an evolution is always 25. The minimum theoretical state complexity is $4/(25 \cdot 258) = 0.000620$. The actual state complexity for the simple system is, however, 0.000784.

The discrepancy results from two factors related to the initial and terminal conditions. All carriages are assumed to begin unloaded at the top of their respective shafts. A system cycle therefore requires a round trip between a queue and a magazine and, to complete an integer number of cycles, a carriage must return to its starting position at the end of the evolution. However, since evolutions end when the last item is delivered, the carriage is located outside of a magazine and the last system cycle is incomplete. The actual temporal evolution length is therefore less than the theoretical value, which increases the state complexity. Because there are items left in the queue after the carriage unloads the last item, its destination queue is unsure. Although the carriage is only in this state for one time step, it still represents a distinct state and increases the total number of distinct states to five, further increasing the state complexity. Although a state is entered only once in the course of an evolution, the state complexity can be affected significantly. In the simple case considered, the system is in one of the four cycle states for 6341 of the 6342 time steps (99.9842%), which corresponds to a state complexity of 0.000631. The addition of the single state results in an increase of the state complexity of 24.3%. However, the disproportionate increase in the state complexity from the addition of this one state agrees with the definition of algorithmic complexity. Changes in the complexity from additional cycles involving the four most frequent states are negligible—only the bits describing the number of cycle changes, which are small relative to the bits required to describe the cycle. An additional state (or pattern), regardless of the frequency of occurrence, requires a complete description that involves significantly more information.

The limit of the number of items involved is inversely proportional to the square of the number of items. For sufficiently large quantities of items, the effect of the number of cycles is therefore negligible, bringing state complexity closer to agreement with the definition of algorithmic complexity.

The impact of the temporal evolution length on the state complexity suggests that only comparisons between evolutions involving the same number of items are valid. While the minimum and maximum theoretical throughputs are unaffected by the number of items delivered, a decrease in the number of items delivered tends to decrease to the temporal evolution length and bias the state complexity toward larger values. Since the number of items delivered is the only dynamic evolution measure common to all evolutions of a particular system size with respect to system complexity, halting evolutions cannot validly be compared to nonhalting evolutions.

To perform an analysis consistent with that for logical complexity, we next explore the set of 2-2-2 evolutions, a set small enough to explore exhaustively but large enough to exhibit a range of behaviors and performance. Although the possible minimum and maximum state complexity boundaries are less apparent, the distribution suggests a triangular boundary with one vertex at the minimum state complexity/minimum throughput combination corresponding to a mimic.

Larger system sizes always contain a mimic of a q-1-1 system, which always shares approximately the same logical complexity despite differences in the number of items transported because the number of items, the number of logical steps, and the number of temporal steps tend to increase proportionally. Doubling the number of items will not change the logical complexity because it also doubles the number of logical steps. For state complexity, however, the relative number of temporal steps and the actual states visited are important. To be a mimic of a q-1-1 system and be complete, an evolution must transport all items from all queues to the magazine corresponding to the lone item type.

Mimicry does not result when queue distributions are comprised of a single item type if the proper physical connectivity is present. Evolutions with the minimum state complexity at the maximum throughput reflect this condition and correspond to configurations with both carriages connected to the magazine corresponding to the lone item type. Each shaft is connected to a unique queue. Since carriages run between a single queue and magazine, each carriage has only one pattern throughout the evolution. However, state complexity is based on the number of system states and therefore a combination of individual carriage states. Since carriages share a common magazine, a phase lag is introduced in the evolution, resulting in eight combinations of carriage states per system cycle, twice the number of states per cycle for an individual carriage.

The maximum theoretical number of distinct system states possible from two carriages with equal numbers of distinct individual states is always twice the number of individual carriage states, regardless of the individual cycle times of each carriage state. The actual states for the deterministic, a two-carriage evolution with a phase lag, is presented in Table 12.

Since the carriages are phase lagged, one carriage ends before the other and adopts the "unsure of queue" state while the remaining operational carriage continues to transport its items. For the 2-2-2 systems considered, there are three additional system states resulting from the transient period at the end of the evolution corresponding to the combination of the unsure first carriage with the second carriage's traveling to a magazine, unloading, and becoming unsure of its own destination queue. Transients also occur at the start of these evolutions as a result of the interaction of the carriages. The delay of the second carriage when both carriages arrive at the same magazine in the first system cycle, meaning the system experiences the distinct state of both carriages' loading simultaneously in their respective queues, is a state not experienced at any other point in the evolution.

First Carriage	Second Carriage
Loading	Travel to a queue
Travel to a magazine	Travel to a queue
Travel to a magazine	Loading
Travel to a magazine	Travel to a magazine
Unloading	Travel to a magazine
Travel to a queue	Travel to a magazine
Travel to a queue	Unloading
Travel to a queue	Travel to a queue

Table 12. The dominant sequence of carriage states in a two-carriage system with a constant phase lag. The queue and magazine identification numbers are omitted because, with a constant phase lag, the sequence of the same types of states is identical.

The fourth state resulting from transients at the start and end of the two-carriage, single-pattern evolution, which represents 1/3 of the total number of distinct states, significantly increases the complexity. The single state resulting from the transient at the end of the evolution of the mimic also has a significant impact on the total number of

https://doi.org/10.25088/ComplexSystems.22.3.247

distinct states, but at 25%, the effect is less than that for a two-carriage system. Once again, we see that the transients have a significant impact on the state complexity and cannot be ignored even though they represent a small fraction of the entire temporal evolution length.

In addition to the additional states resulting from transients, the state complexity corresponding to the two-carriage, single-pattern evolution is also greater than the state complexity of the mimic because the temporal evolution length of the mimic is approximately twice that of the two-carriage system.

For configurations with complete shaft-magazine connectivity with respect to the magazine corresponding to the single present item type and complete shaft-queue connectivity (like configurations 247 and 255), both carriages visit both queues, so each carriage has two distinct patterns. With two distinct patterns for each carriage, each comprised of four states, four combinations of patterns are possible, each with a theoretical maximum of eight state combinations, for a total of 32 possible state combinations (ignoring transients). When states common to each pattern are considered, the possible number of distinct states is 18, assuming the phase lag is constant and the sequencing of carriage states is identical to the two-carriage, single-pattern evolution. This number can also be determined by calculating the number of unique combinations of states at each stage in the sequence defined in Table 12.

Assuming a constant number of states resulting from transients, it is therefore possible for the state complexity to be significantly greater than the measured value. However, because of logic dictating that a carriage selects the first available queue it finds, and because queues are evaluated for availability in order, lower-numbered queues have a de facto priority. Both carriages therefore empty the first queue before simultaneously switching to the second queue, and only two of the possible four combinations of carriage patterns are utilized, resulting in 14 distinct states when repeated states are removed. With the proper ratio of item types in each queue to the number of carriages, or in the presence of additional logic specifying how resources are shared, additional system patterns are possible that result in greater state complexity.

10.2 Theoretical Boundaries

A maximum theoretical state complexity can be calculated based on the number of queues, shafts, and magazines and the sequence of state changes for the corresponding number of shafts (carriages). It is possible to calculate the maximum theoretical state complexity corresponding to a specific queue distribution—queue distributions with a single item type effectively limit the number of carriage destinations

https://doi.org/10.25088/ComplexSystems.22.3.247

and decrease the possible state complexity. The maximum theoretical state complexity as described is only applicable at the maximum throughput, where all carriages are operational for the entire evolution and carriages only enter the patterns defined. The maximum theoretical state complexity at any given throughput is, however, calculated by assuming the number of items transported is constant. At the minimum throughput, the number of operational carriages is known, since the minimum throughput always corresponds to a mimic system with a single carriage.

Calculation of the minimum theoretical state complexity for evolutions with either one or all carriages operational is also different than for evolutions with carriages that halt partially through the evolution. The minimum number of states in a complete evolution will always correspond to the queue distribution with the lowest diversity—one containing a single system type—because it effectively eliminates the possible number of destinations. The minimum theoretical state complexity at the maximum possible throughput should therefore always correspond to an evolution in which all carriages are connected to a respective queue and the magazine corresponds to the lone item type. The minimum number of phase lags occurs, but there is at least one for the evolution to be complete, since all carriages must share a common magazine. At the minimum throughput, a single carriage delivers all items from all queues (to be complete) and the state complexity is described by equation (33) where τ is the number of temporal steps per additional item, c is the number of items in each additional queue, and q is the number of queues:

$$C_S = \frac{2\,(q+1)+2}{\tau c(q+1)}. \tag{33}$$

The theoretical boundaries for the state complexity for 2-2-2 size systems are presented in Figure 30. The boundary lines reflect the minimum and maximum state complexities for evolutions with various numbers of operational carriages through the range of possible throughputs. The points A and B respectively represent the minimum and maximum theoretical state complexities at the maximum theoretical throughput, when the number of operational carriages and therefore the number of possible states are known. Similarly, point C corresponds to the minimum/maximum state complexity at the minimum theoretical throughput, when only a single carriage is in operation. The evolutions are not necessarily within the boundaries, because of states resulting from transient periods and because not all possible states are visited. The differences between the minimum/maximum state complexity points at the minimum and maximum throughputs and the minimum/maximum state complexity boundary lines are a re-

https://doi.org/10.25088/ComplexSystems.22.3.247

sult of the lack of knowledge of the actual number of states used in evolutions with carriages that halt.

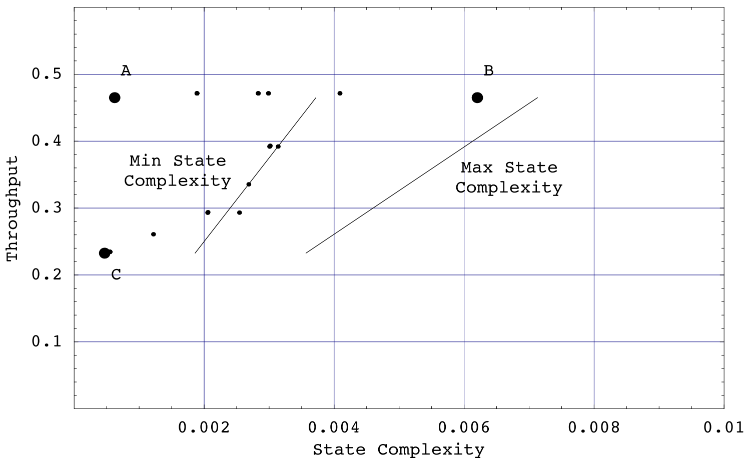


Figure 30. The relationship between the minimum/maximum theoretical state complexities and the actual distribution of evolutions.

The differences between the actual and theoretical minimum and maximum state complexities occur for two reasons. First, the theoretical state complexities consider all states that are possible, given the variety of queue distribution to account for the lack of information regarding the actual numbers of operation carriages. Second, the minimum and maximum state complexities consider only the possible steady state cycles created when a phase lag is introduced and therefore ignore the effect of transients on the state complexity, which can significantly increase the state complexity.

Since the theoretical state complexity boundaries account for all possible carriage state combinations that are feasible within the constraints of model definitions, the differences between the actual and theoretical state complexities may be thought of as being equivalent to the result of the introduction of randomness or variability into the model. This variability could be a function of nondeterministic cycle times, the inclusion of actual physical dimensions, or the elimination of the assumption of identical distances between each queue and magazine.

Variability tends to both shift and stretch the complexity boundaries toward greater complexity values. The increase in complexity is simply a reflection of the effects of variations, and is not a function of the underlying relationships between complexity and performance that may be applicable toward optimization. Since our task is to

https://doi.org/10.25088/ComplexSystems.22.3.247

identify the presence, strength, and causes of relationships, we ignore nondeterminism in our analyses in order to capture the fundamental reasons why relationships between behavior, performance, and robustness may exist, while recognizing that the usefulness of behavior as part of an optimization tool may be limited, depending on the amount of system variability.

The relative effects of the inclusion of all potential states and the omission of transient states are apparent at the minimum theoretical state complexities corresponding to the minimum and maximum throughputs, because of the additional information known regarding the evolutions. At throughputs between the theoretical minimum and maximum that correspond to evolutions with halting carriages, less information is known about the actual states visited because the halting sequence is unknown.

Figure 30 suggests that the state complexity can be maximal at less-than-maximal throughputs because the potential number of distinct states, created from the combinations of carriage states for various numbers of operational carriages, is greater. However, whether or not an evolution with suboptimal throughput corresponds to the maximum state complexity is dependent on the relative effects of halting creating additional evolution states and the increase in the temporal evolution length resulting from the loss of operational carriages.

Normalization of the number of distinct states with respect to the temporal evolution length has a correlating effect between state complexity and throughput and implies that, to achieve the maximum state complexity at a less-than-optimal throughput, halting must occur late in the evolution. As carriages halt earlier in the evolution, the number of states may be greater than the number of states for an evolution involving all carriages throughout the entire evolution, but the increase in the temporal evolution length caused by disproportionate carriage utilization will result in lower state complexity.

To achieve the maximum number of states per pattern, an evolution must also have the maximum number of phase lags possible because the number of combinations of individual carriage states is greater with phase lags. The maximum number of phase lags, however, does not necessarily result in the maximum number of system states. The effects of the number of phase lags on state complexity through the number of combinations of individual carriage states must therefore be determined in the context of the number of system patterns.

10.3 Qualitative Characterizations

The boundaries in Figure 30 are created assuming a constant, maximum number of phase lags. If the states created from various phase lags are included, the maximum state complexity boundary shifts to the right, approaching the maximum state complexities possible for nondeterministic systems, where all combinations of individual carriage states are possible. Phase lags are therefore related to the state complexity, but do not characterize the state complexity in the same way as with logical complexity. Figure 31 presents a mapping of logical complexity values into state complexity space, where the darkness of a point indicates the value of the evolution's logical complexity (darker values indicate greater logical complexity). Since the average number of phase lags is directly related to the logical complexity, the mapping therefore indicates the effects of the number of phase lags on the state complexity. If the number of phase lags were a direct indicator of state complexity, then the darkness should increase as the state complexity increases across any line of constant throughput, and points of constant darkness should be in line. Figure 31 shows that the state complexity does not necessarily increase with the average number of phase lags. Across any line of constant throughput, there may be various numbers of phase lags with no general trend, or similar numbers of phase lags distributed across a range of state complexity values. This mapping indicates that the number of phase lags does not necessarily result in more system patterns and, despite the additional combinations of states associated with larger numbers of phase lags, state complexity does not necessarily increase. Figure 31 also illustrates the differences between the definitions of logical complexity and state complexity.

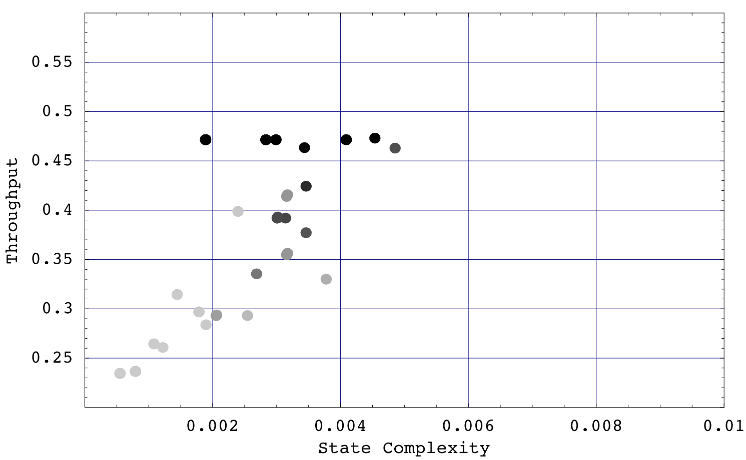


Figure 31. The mapping of logical complexity values in state complexity space.

With only a single item type, all carriages are forced to the same magazine, which results in time delays and phase lags in complete evo-

https://doi.org/10.25088/ComplexSystems.22.3.247

lutions. When more carriages are involved, the state complexity increases. However, increasing the variety in the queue distribution will tend to increase the state complexity for the same throughput because variety in the item types increases the variety of locations visited and therefore the number of states entered.

The resultant qualitative characterization of state complexity with respect to variety in the queue distribution and the onset of carriage halting is presented in Figure 32. For systems with more than two magazines, there are varying degrees of queue distribution variety beyond variety and no variety, and these generally correspond to the relative state complexity. Ultimately, the state complexity is a function of the queue distribution taken in the context of the physical connectivity. There is therefore no distinct line corresponding to queue distribution variety, limiting the characterization of the state complexity/throughput space to a qualitative one.

10.4 State Complexity and Throughput

For state complexity, the number of distinct states and the number of patterns are not scalable with temporal evolution length, and comparisons of incomplete and complete evolutions can be quite misleading. As a result, we only consider complete evolutions and their subsets in establishing correlations between state complexity and throughput, although comparisons with nonhalting evolution sets are used to illustrate the differences between these evolutions and complete evolutions.

The summary of mean values for the state complexity for nonunique and unique evolution subsets for 2-2-2 size systems is presented in Table 13. For 2-2-2 size systems, the mean state complexity of incomplete evolutions is less than the mean for the entire set of nonhalting evolutions, resulting in a greater mean state complexity for complete evolutions. To properly evaluate the effects of incomplete evolutions on the mean state complexity, an analysis of additional system sizes is required. Comparisons of complete evolution sets for different system sizes remain valid and show that the mean state complexity for the evolutions corresponding to robust configurations is greater by approximately six percent. This difference implies that the most adaptable configurations correspond to the most complex evolutions. Since the throughputs remain unchanged regardless of the complexity measure used, the greater mean throughput associated with the evolutions of robust configurations also implies a direct relationship between complexity and performance and therefore between adaptability and performance.

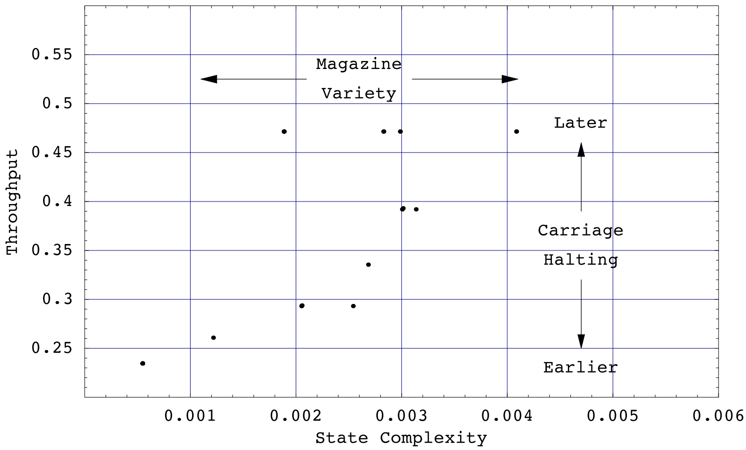


Figure 32. The qualitative characterization of the state complexity space with respect to halting carriages and variety in the queue distribution. Variety in the queue distribution permits greater exploration of possible states, increasing the state complexity. Carriages that halt later in an evolution result in lower temporal evolution lengths and therefore greater throughput and greater state complexity.

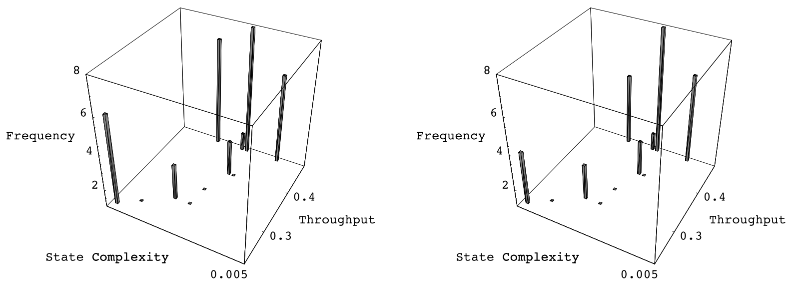
	Mean C_S	Mean R
Nonhalting	0.002386	0.3748
Complete	0.002453	0.4015
Robust	0.002591	0.4069
Unique nonhalting	0.002767	0.4058
Unique complete	0.002790	0.4310
Unique robust	0.002969	0.4715
Mimics	0.000668	0.2355

Table 13. Summary of the mean state complexity and throughput for 2-2-2 evolution subsets.

Although the set of evolutions corresponding to the most robust configurations have the greatest mean state complexity and throughput of any set achieved, the mean values are not equal to the maximum values achieved, although some evolutions in the set may correspond to maximum values. Figures 33 and 34 illustrate the distribution of values for complete and robust evolution sets with three-dimensional frequency landscapes and cross-sectional histograms. Because the same scale is used, it is evident that the maximum frequen-

https://doi.org/10.25088/ComplexSystems.22.3.247

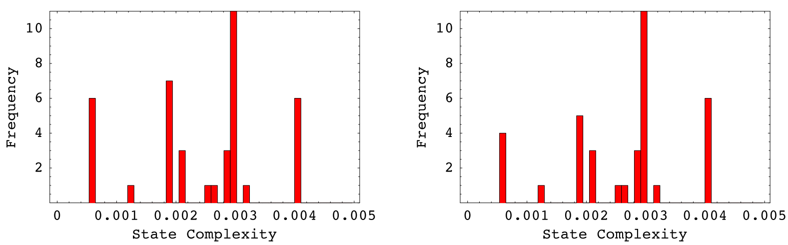
cies remain relatively unchanged for all evolution sets. The similarities exist because the evolutions located at the maximum throughput for a range of state complexities are common to all sets and help explain the trend toward increased throughput and state complexity with set refinement.



(a) (b)

Figure 33. The three-dimensional state complexity/throughput frequency landscapes for (a) complete evolutions and (b) robust evolutions for 2-2-2 size systems.

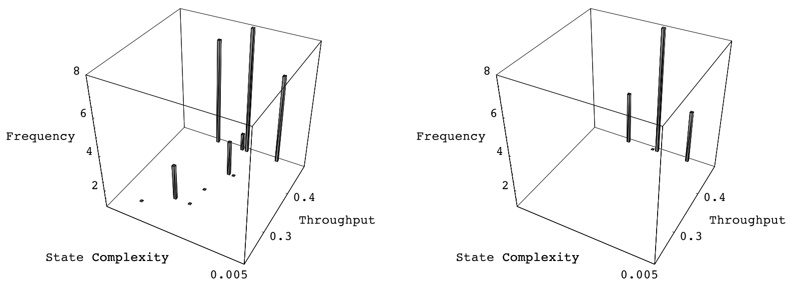
These clusters of evolutions are also evident in Figures 35 and 36, which show the three-dimensional frequency landscapes and histograms with respect to state complexity for unique sets of complete and robust evolutions. The greater mean values of these unique sets relative to their corresponding nonunique equivalents again indicate a relationship between adaptability, complexity, and throughput. They also suggest that system size is related to state complexity and throughput. Unique evolutions represent the largest size configurations possible in the set and are able to support greater state complexity and throughput.



(a) (b)

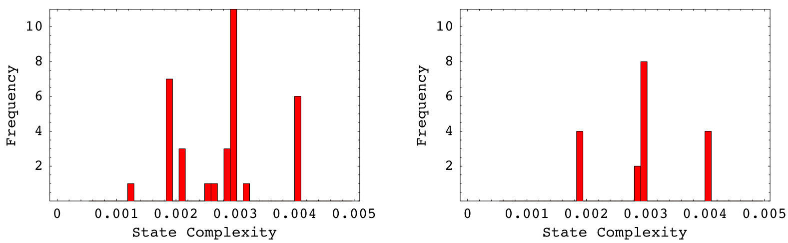
Figure 34. The cross-sectional histograms with respect to state complexity for (a) complete evolutions and (b) robust evolutions for 2-2-2 size systems.

https://doi.org/10.25088/ComplexSystems.22.3.247



(a) (b)

Figure 35. The three-dimensional state complexity/throughput frequency landscapes for (a) unique complete evolutions and (b) unique robust evolutions for 2-2-2 size systems.



(a) (b)

Figure 36. The cross-sectional histograms with respect to state complexity for (a) unique complete evolutions and (b) unique robust evolutions for 2-2-2 size systems.

10.5 More Complex Systems

The analysis of systems of other sizes, particularly larger systems containing significantly more evolutions, provides supporting evidence for the preliminary conclusions regarding state complexity drawn from observations of 2-2-2 evolution sets. Analysis of different system sizes illustrates the effects of changing the absolute and relative number of queues, shafts, and magazines in the context of the logic used. The distributions of other system sizes further demonstrate the practical application of the definition of algorithmic complexity and also illustrate the effects of determinism on state complexity. The mean state complexities for the various evolution subsets of nontrivial system sizes are listed in Table 7. The mean values in Table 7 illustrate why comparisons between incomplete and complete evolutions are in-

https://doi.org/10.25088/ComplexSystems.22.3.247

valid as well. For 2-2-2 size systems, the mean state complexity of all nonhalting evolutions is less than the mean state complexities of the complete and robust evolutions sets. The same is true with respect to the mean state complexities of unique evolution sets. This same relationship for logical complexity led to the conclusion that a relationship exists between robustness with respect to uncontrollable queues, logical complexity, and throughput. The mean state complexities of nonhalting evolutions in Table 7 are greater than the mean state complexities of respective complete evolutions. Even if the number of distinct states entered in an incomplete evolution is less than the number of distinct states entered in a complete evolution, the state complexity may still be greater if halting occurs early enough in the evolution. The differences in the mean values therefore reflect the number of incomplete evolutions, the numbers of distinct states entered by each, and the point in each evolution at which halting occurs. For all system sizes in Table 14, the mean state complexity corresponding to the most robust configurations resulting in complete evolutions for all queue distributions considered is greater than the mean state complexity of all complete evolutions of the same system size.

11. Conclusion

The models presented serve as evidence that some relationship is present between behavior and optimality. Given the freedom to evolve their characteristics, systems with complex interactions have been shown in many cases to evolve to a complex regime, which offers a combination of stability to take advantage of niche environments and adaptability to respond to environmental changes, which can be indirectly self-induced. The same characteristics of complex behavior that result in adaptability also result in good performance. The stability allows a system to remain at optimal or near-optimal fitness peaks, while sufficient flexibility permits state-space exploration in search of greater performance.

https://doi.org/10.25088/ComplexSystems.22.3.247

System	N	C	R	UN	UC	UR	M
1-2-2	0.003649	0.003649	0.003649	0.004221	0.004221	0.004724	0.000788
1-2-3	0.004212	0.004212	0.004212	0.004921	0.004921	0.005985	0.001035
1-2-4	0.004651	0.004651	0.004651	0.005396	0.005396	0.006767	0.001223
1-3-2	0.008441	0.008441	0.008441	0.010109	0.010109	0.012195	0.002606
1-3-3	0.008612	0.008612	0.008612	0.010858	0.010858	0.014131	0.003313
1-3-4	0.009440	0.009440	0.009440	0.011946	0.011946	0.016131	0.00394
1-4-2	0.014686	0.014686	0.014686	0.017853	0.017853	0.021516	0.005184
1-4-3	0.015008	0.015008	0.015008	0.019555	0.019555	0.024709	0.007309
1-4-4	0.015842	0.015842	0.015842	0.021012	0.021012	0.024709	0.008367
2-2-2	0.002386	0.002453	0.002591	0.002767	0.002790	0.002969	0.000668
2-2-3	0.002874	0.002843	0.003059	0.003344	0.003322	0.003712	0.000873
2-3-2	0.004914	0.004891	0.005171	0.005921	0.005995	0.007109	0.001703
2-3-3	0.005507	0.005431	0.006017	0.006955	0.006877	0.008962	0.002345
2-3-4	0.006245	0.006071	0.006914	0.007870	0.007660	0.010203	0.002845
2-4-2	0.007732	0.007269	0.007609	0.009391	0.008957	0.010914	0.003382
2-4-3	0.008471	0.008203	0.008924	0.010956	0.010553	0.012724	0.004743
3-2-2	0.001876	0.001935	0.002076	0.002170	0.002155	0.002271	0.000578
3-2-3	0.002318	0.002289	0.002527	0.002694	0.002635	0.002893	0.000751
3-2-4	0.002691	0.002574	0.002867	0.003104	0.002991	0.003316	0.000884
3-3-2	0.004076	0.004156	0.004585	0.004969	0.005191	0.006635	0.001331
3-3-3	0.004664	0.004723	0.005422	0.005947	0.006104	0.008803	0.001928
3-3-4	0.005347	0.005340	0.006200	0.006791	0.006862	0.010266	0.010413
3-4-2	0.006358	0.006192	0.006689	0.007827	0.007908	0.010413	0.002566
4-2-2	0.001545	0.001537	0.001668	0.001779	0.001684	0.001772	0.000523
4-2-3	0.001948	0.001798	0.001991	0.002260	0.002043	0.002206	0.000678
4-2-4	0.002287	0.002014	0.002211	0.002638	0.002320	0.002517	0.000796
4-3-2	0.003508	0.003628	0.004124	0.004285	0.004502	0.005629	0.001122
4-3-3	0.004072	0.004138	0.004964	0.005205	0.005346	0.007361	0.001667
4-4-2	0.00564	0.005779	0.006420	0.006976	0.007502	0.010867	0.002158

Table 14. The mean state complexity for evolution subsets of different system sizes (N = nonhalting, C = complete, R = robust, UN = unique and nonhalting, UC = unique and complete, UR = unique and robust, M = mimics).

The naval weapons elevator simulations that have been presented are able to efficiently predict behavior. Taken together with the complexity measures, the simulations are extremely helpful to state the behavior of the system. From our simulations, we find several results of potential interest. Not surprisingly, the larger the system size, the more logical and state complexities exist. Likewise, both logical and state complexities increase in robustness with more evolutions. It is also clear that higher throughput occurs in more complex systems. With this fact in mind, the relationships between logical complexity and throughput as well as state complexity and throughput follow the same trend, indicating that a system with more complexity results in

https://doi.org/10.25088/ComplexSystems.22.3.247

an increase in throughput. Despite higher performance with larger systems, there are some evolutions of configurations that act like smaller systems. Their lower performance is reflected in the values of logical and state complexities. These kinds of systems are referred to as mimics. However, although the relationships between complexities and throughput are presented in both logical and state terms, a naval weapons elevator system remains to be investigated by additional measures also introduced in this paper to clearly describe the complexity of the system. Those remaining measures will be presented in our future work.

Acknowledgments

Our thanks to Mr. Gary Good of NGNN for vital configuration alternatives, and Mr. Ranon Jientrakul for manuscript editing.

References

[1] S. Wolfram, *A New Kind of Science*, Champaign, IL: Wolfram Media, Inc., 2002.

[2] A. Ilachinski, *Cellular Automata: A Discrete Universe*, Singapore: World Scientific, 2001.

[3] J. von Neumann, *Theory of Self-Reproducing Automata* (A. W. Burks, ed.), Urbana, IL: University of Illinois Press, 1966.

[4] S. Wolfram, "Universality and Complexity in Cellular Automata," *Physica D*, **10**(1–2), 1984 pp. 1–35. doi:10.1016/0167-2789(84)90245-8.

[5] S. Wolfram, "Statistical Mechanics of Cellular Automata," *Review of Modern Physics*, **55**(8), 1983 pp. 601–644. doi:10.1103/RevModPhys.55.601.

[6] H. V. McIntosh, "Wolfram's Class IV Automata and a Good Life," *Physica D: Nonlinear Phenomena*, **45**(1–3), 1990 pp. 105–121. doi:10.1016/0167-2789(90)90177-Q.

[7] W. Li and N. Packard, "The Structure of the Elementary Cellular Automata Rule Space," *Complex Systems*, **4**(3), 1990 pp. 281–297. http://www.complex-systems.com/pdf/04-3-3.pdf.

[8] W. Li, N. Packard, and C. Langton, "Transition Phenomena in Cellular Automata Rule Space," *Physica D: Nonlinear Phenomena*, **45**(1–3), 1990 pp. 77–94. doi:10.1016/0167-2789(90)90175-O.

[9] R. Anderson and B. Sturges, "Inherent Behavior in Complex Systems," *Proceedings of the 2004 Japan–USA Symposium on Flexible Automation*, Denver, CO, American Society of Mechanical Engineers, 2004.

[10] J. Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor, MI: University of Michigan Press, 1975.

[11] M. Mitchell, *An Introduction to Genetic Algorithms*, Cambridge, MA: The MIT Press, 1998.

[12] G. Chaitin, *Information, Randomness, and Incompleteness: Papers on Algorithmic Information Theory*, Singapore: World Scientific Publishing, 1987.

[13] G. Chaitin, *Exploring Randomness*, New York: Springer, 2001.

[14] G. Chaitin, *Algorithmic Information Theory,* Cambridge, UK: Cambridge University Press, 1987.

[15] S. Kauffman, *The Origins of Order, Self-Organization and Selection in Evolution*, New York: Oxford University Press, 1993.

[16] S. Forrest and T. Jones, "Modeling Complex Adaptive Systems with Echo," in *Complex Systems: Mechanism of Adaptation* (R. J. Stonier and X. H. Yu, eds.), Washington, DC: IOS Press, 1994 pp. 3–20.

[17] J. F. Nash, Jr., "Equilibrium Points in *n*-Person Games," *Proceedings of the National Academy of Sciences of the United States of America*, **36**(1), 1950 pp. 48–49. doi:10.1073/pnas.36.1.48.

[18] S. A. Kauffman, *At Home in the Universe: The Search for the Laws of Self-Organization and Complexity*, New York: Oxford University Press, 1995.

https://doi.org/10.25088/ComplexSystems.22.3.247