Chapter 12

Alternaltive Logic

The model elevator system has been defined as the product of three primary variables: the physical configuration, the input stream, and the operational logic. Thus far, we have seen how the physical configuration and input streams affect system performance and adaptability, but in the context of a single set of system rules. Characterization of complexity measures in systems with variations on the operational logic offers a view of a larger state space and indicates the extent to which operational logic affects complexity and its relationship to throughput and robustness.

The systems presented in the discussion of complexity measures used the set of operational logic most representative of a conventional naval weapons elevator system. This system is characterized by serial carriage and ballistic hatch/magazine door operations and the application of interlocks for compartmentalization and survivability. By combining these attributes, we can obtain four distinct sets of system rules that dictate how a carriage interacts with hatches and doors and how it travels within its shaft.

The combination of the serial operation of carriages and hatches/doors and the application of interlocks is arguably the safest operational mode, but also the most prohibitive with respect to performance. Elimination of these restrictions increases the effectiveness of the weapons elevator system and, in general, the carrier, by reducing the time spent off-station during a reloading and strike-down operation, during which normal carrier flight operations are restricted and the carrier exhibits vulnerability. The elimination of one or both of these restrictions tends to increase throughput by effectively reducing or eliminating the waiting time associated with hatch/door operations from the perspective of the carriage. Theoretically, the changes in the waiting times can alter the sequencing of states and therefore the complexity. However, differences in complexity measures resulting from changes in shaft logic are predominantly a function of temporal, not state changes.

Figure 12.1 presents the distribution of evolutions with respect to logical complexity and throughput for all 2-3-4 configurations with (a) serial operation of hatches/doors and carriages and with interlocks (SIL), (b) serial operation of carriages and hatches/doors without interlocks (SNIL), (c) parallel carriage and hatch/door operations with interlocks (PIL), and (d) parallel carriage and hatch/door operations without interlocks (PNIL), with all distributions using the same scales for axes. A comparison of the ranges of the throughput values for each figure indicates that interlocks have a greater effect on performance than the serial/parallel nature of carriage/door operations. More importantly, a comparison of the distributions indicates a nearly identical distribution for all sets of operational logic, although the range of logical complexity and throughput values are proportionally scaled.

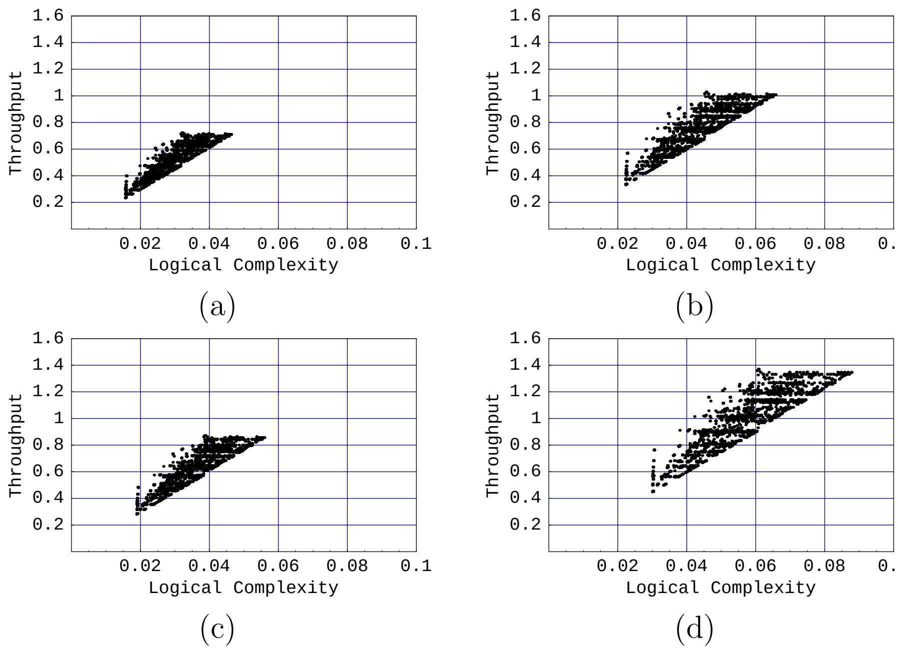


Figure 12.1: The distribution of 2-3-4 evolutions with respect to logical complexity and throughput with (a) SIL, (b) SNIL, (c) PIL, and (d) PNIL shaft logic. Changing the shaft logic does not affect the relative distribution of evolutions, but proportionally scale the logical complexity and throughput values. For the cycle times assumed, interlocks have a more significant effect on performance and logical complexity than the serial/parallel nature of carriages and hatches/doors.

Scaling with changes in operational logic is also apparent in the comparison of distributions of 2-3-4 configurations with respect to state complexity and throughput presented in Figure 12.2. However, scaling of both the complexity and throughput is not apparent in Figures 12.3 and 12.4, for the compressed state complexity and the number of states used. For these measures, only throughput, not complexity, varies with changes in the operational logic.

The reason for changes in both the complexity and throughput for the logical complexity and state complexity is that these measures are based on the temporal evolution length, which is dependent on the evolution logic. The number of unique states and the compressed state

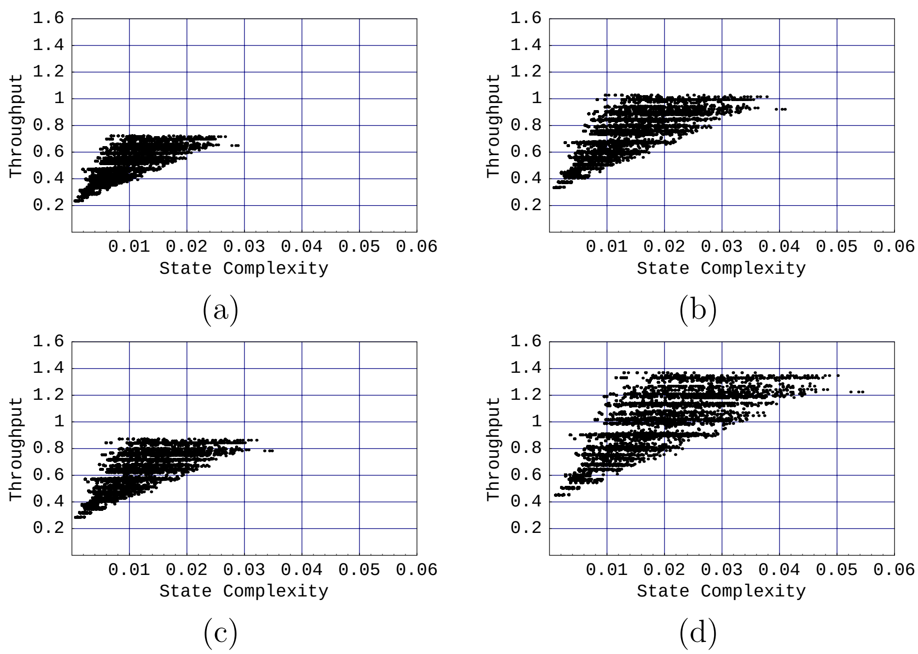


Figure 12.2: The distribution of 2-3-4 evolutions with respect to state complexity and throughput with (a) SIL, (b) SNIL, (c) PIL, and (d) PNIL shaft logic. Like logical complexity, changes in the shaft logic result in scaling of the state complexity and throughput in proportion.

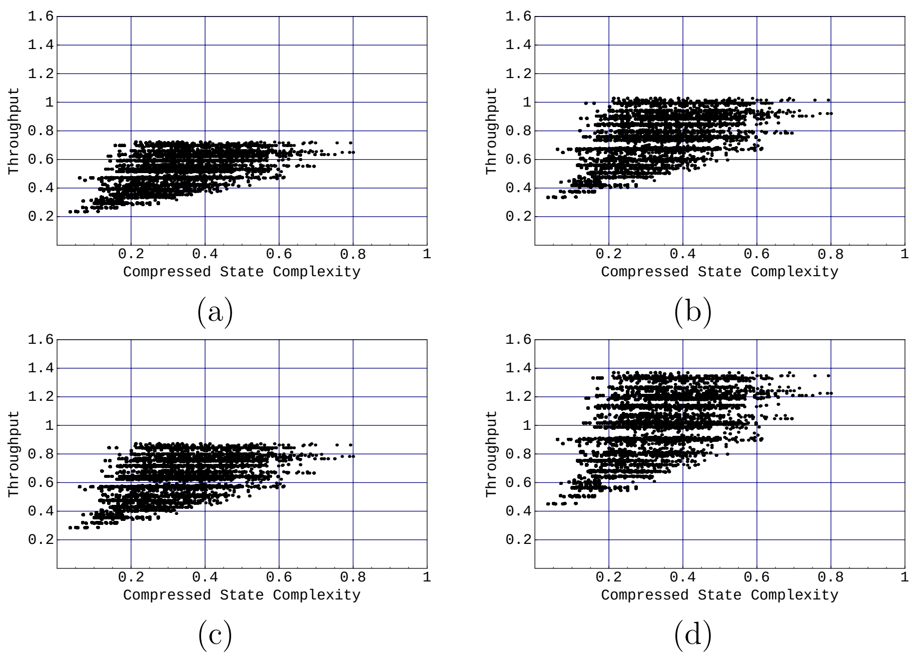


Figure 12.3: The distribution of 2-3-4 evolutions with respect to compressed state complexity and throughput with (a) SIL, (b) SNIL, (c) PIL, and (d) PNIL shaft logic. Unlike the logical and state complexities, the compressed state complexity does not scale with changes in shaft logic.

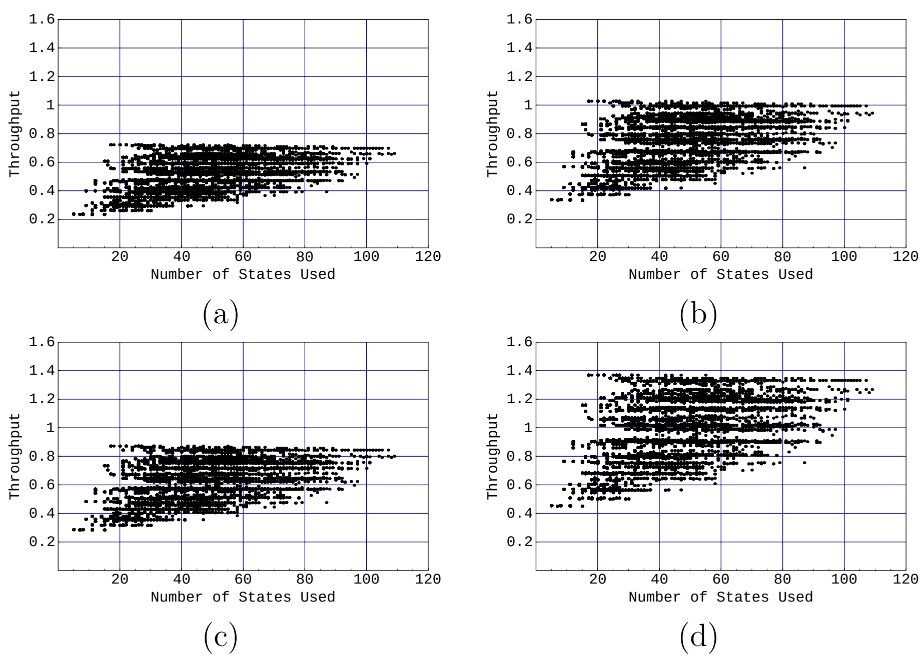


Figure 12.4: The distribution of 2-3-4 evolutions with respect to the number of states used and throughput with (a) SIL, (b) SNIL, (c) PIL, and (d) PNIL shaft logic. The number of unique states also does not scale with shaft logic, implying that shaft logic affects temporal evolutions, but not logical evolutions.

complexity are exclusively state-based measures - that is, these measures can be derived exclusively from the logical evolution - and the number of states and their sequences remain largely unchanged for a configuration and input stream.

An illustrative example of the effects of operational logic on complexity measures, we consider the evolution of the completely connected 2-3-4 configuration 262143 with a (60-40-0-0) queue distribution. For simplicity, we will only consider the two evolutions corresponding to SIL and PNIL operational logics, which have the greatest difference between them. Figure 12.5 presents the *temporal* evolutions under the two logic sets. The evolutions share similar patterns, yet the evolution corresponding to the PNIL system is significantly shorter, at 2542 steps versus 4836 ($\approx 52\%$). The shorter temporal evolution length is strictly a result of the shorter transit time from queue to magazine and from magazine to queue. The theoretical minimum transit time for a carriage through all possible levels with the assumed carriage speed of 7 steps per level is 42 steps. With serial carriage/hatch operations and the use of interlocks, the typical transit time is 108 steps while a PIL system requires 46 steps. The use of alternative operational logic can therefore be thought of as effectively reducing the waiting time associated with hatch/door cycle times and interlocking, or equivalently, as increasing the cycle/transit times of carriages, hatches, and doors. If the hatch/door cycle times are sufficiently low, the hatches/doors would appear non-existent with respect to the carriage, leaving transit time strictly a function of carriage speed.

Depending on the relative times for transit and hatch/door operations, sufficiently low hatch/door cycle times would result in a completely interlocked system without sacrificing non-interference of carriages and hatches/doors in a system with parallel operations. The decrease in the cycle times is apparent in a comparison of the evolution histories in Figure 12.5, where the relative widths of the cycles corresponding to loading and unloading are narrower for SIL logic.

While the temporal evolution length changes as a result of operational logic, the *logical* evolution does not. Figure 12.6 presents the logical evolution histories for the 262143 configuration with a (60-40-0-0) queue distribution and SIL and PNIL shaft logic. The evolutions are *identical*, in both the states and their sequence. For this, and most conventional evolutions, changing the operational logic with respect to carriage/hatch operations and with respect to the use of interlocks does not change the sequence of states and only consistently alters the transit time of carriages. One of the primary reasons for a lack of state changes with changes in operational logic is the simplifying assumption that all magazines exist at a common level. When magazines are located at different levels, there exists the possibility for changes in state sequences in a deterministic evolution depending on the relative transit times of carriages and cycle times of hatches/doors.

For the assumed systems, identical logical evolutions but varying temporal evolutions mean that any complexity measure associated with temporal effects are affected by changes in operational logic, but those defined exclusively by states are not. Thus, decreasing the temporal evolution length by decreasing the effective carriage transit times results in higher

(a)

(b)

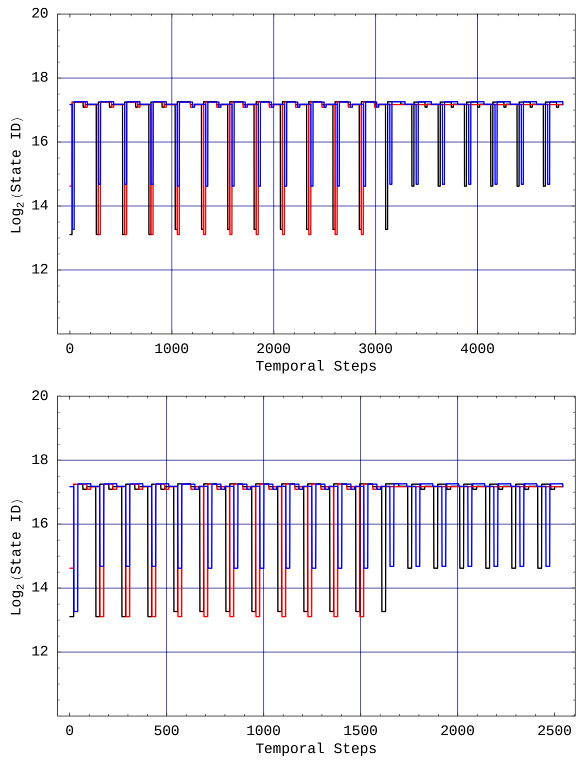


Figure 12.5: The temporal carriage histories for the completely connected 2-3-4 configuration 262143 with a (60-40-0-0) queue distribution with (a) SIL and (b) PNIL shaft logic. The patterns are identical, but the cycles for the SIL evolution are longer because of longer carriage transit times.

(a)

(b)

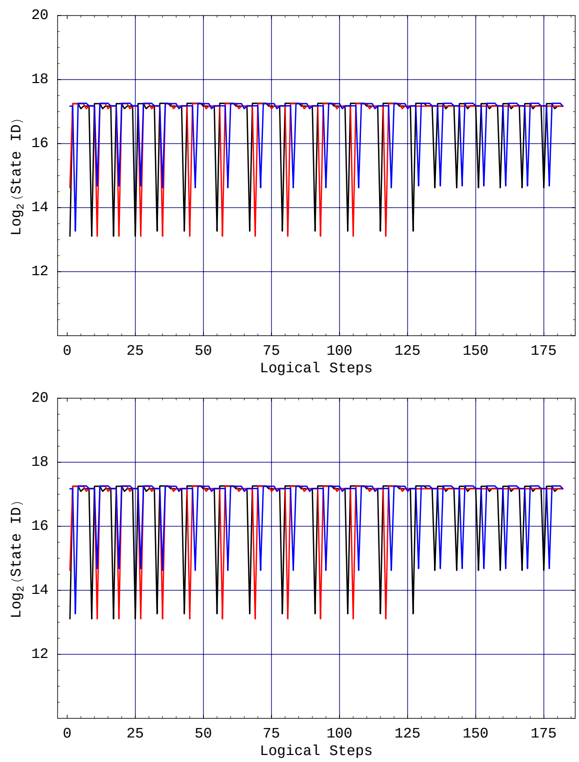


Figure 12.6: The logical carriage histories for the completely connected 2-3-4 configuration 262143 with a (60-40-0-0) queue distribution with (a) SIL and (b) PNIL shaft logic. Since the logical evolutions are identical but the temporal evolutions differ, complexity measures based on temporal evolution lengths are affected by changes in operational logic but strictly state-based complexity measures are not.

logical and state complexities, which are both normalized by the temporal evolution length. The number of unique states and compressed state complexity both remain constant because they are state-based measures.

The increase in complexity for some measures and the lack of change for others presents an apparent paradox. How can a change in an evolution result in both a change and no change in complexity simultaneously? This contradiction runs to the heart of the fundamental definition of algorithmic complexity, compression and randomness.

If the length of a binary sequence increases, but only involves first order repetitions (repetitions of a single bit, not bit sequences), then the information required to completely express the sequence is relatively unchanged as the information required to define the number of repetitions of a bit is largely unchanged by the number of repetitions, especially for longer repetition lengths. For example, in the binary sequences in Figure 12.7, the information required for a complete description is identical. For the sequence in (a), a first order description is <Print '10', Print 8 1's in a row, Print '010'> while a description of sequence (b) is <Print '10', Print 15 1's in a row, Print '010'>. The number of binary bits required to state the number of repetitions in both sequences is 4 (1000=8 and 1111-15), implying that the sequences have identical algorithmic complexity, despite the difference in sequence lengths. This interpretation suggests that the logical evolution and the state sequence are more relevant to the amount of information in a sequence and, for our measures, strictly state-based complexity measures are a better interpretation of algorithmic complexity.

(a) 1011111111010
(b) 10111111111111111010

Figure 12.7: In one interpretation of algorithmic complexity, the sequences in (a) and (b) are algorithmically equivalent because the bits required to describe each sequence in their shortest forms are identical.

However, the intent of algorithmic complexity is to measure the amount of randomness (or compression) in an information stream. The above algorithmic descriptions of sequences (a) and (b) in Figure 12.7 seem to suggest an equivalent compressed form of both sequences, yet the longer sequence in (b) is more compressible relative to its length, meaning the sequence in (a) is more random and therefore has greater algorithmic complexity. This interpretation suggests that temporal, or actual sequence lengths are more representative of algorithmic complexity and complexity measures must account for explicit evolutions.

The subtleties in the interpretation of algorithmic complexity suggest that different interpretations and therefore different complexity measures are applicable when making comparisons of systems based on various changes. Internal comparisons within a set of configurations of a given size with common cycle times and operational logic are always valid. When changes in the operational logic or cycle times are made, both logical and temporal effects are possible. In order to identify the effects on the logical evolution alone, ignoring the information present in first-order repetitions, state-based complexity measures such as compressed state complexity and the number of states should be used. To account for both logical *and* temporal effects, measures like the state and logical complexities that consider the complete sequence of states, including first order patterns, are applicable.

While changing the operational logic related to the serial/parallel nature of carriage/hatch operations and the use of interlocks can theoretically affect the states entered and their sequence in an evolution, thereby changing all complexity measures and their relationships to throughput and robustness, this logic only practically affects the effective transit time of carriages. However, we can envision other, more fundamental changes to the evolution logic which dictate changes in physical construction that significantly alter system dynamics, the potential state space, and the complexity-performance and complexity-adaptability relationships.

Chapter 13

Mobile Carriage Systems

In a conventional elevator system, carriages are assumed to be fixed within a shaft, which is necessarily bi-directional. The terms carriage and shaft are therefore interchangeable in a conventional system. If carriages have the ability to enter and exit the physical confines of shafts, then the terms are no longer synonymous and the state space of operational logics increases significantly. With additional degrees of freedom for carriage mobility, it is possible to create circuits, or networks of system elements that can fundamentally change the decision logic of carriages.

Since the mobility of carriages implies a circuitous operation, carriages are assumed to be uni-directional. Without some set of logic preventing the simultaneous upwards and downwards movement of multiple carriages in a single shaft, uni-directionality is a requirement for avoiding interference. The concept of uni-directionality alters the definitions of validity for connectivity. In order to be part of a circuit, each queue and magazine must be connected to at least one up-shaft and one down-shaft, so that the simplest system or subsystem contains one queue, two shafts, and one magazine. Because the directionality of shafts is important to the definition and operation of configurations, the shaft directionality is included in the configuration codes. In the shaft direction vector, or SDV, a 0 indicates a shaft is down only and a 1 indicates an up shaft.

In conventional elevator systems, the number of carriages is, by definition, equal to the number of shafts. However, in a system with mobile carriages, the number of carriages must be defined explicitly, independently of the number of shafts. For simplicity, the number of carriages is assumed to be equal to the number of queues such that carriages start an evolution in a distinct queue. However, these initial conditions do not imply that a carriage has any long term association with the initial queue, nor does it imply that a system with a specified number of queues is limited to an equal number of carriages. If the number of carriages exceeds the number of queues, the number of carriages in excess of the number of queues are all initially located in a single queue.

In addition to system definition and operation, uni-directionality has an impact on the system state space. With carriages fixed in shafts, the location of a carriage is implicitly defined while the carriage is moving. However, mobile carriages are able to travel in multiple shafts, requiring explicit definition of the current shaft and increasing the potential state space. Additionally, since carriages may have options in the destination queue and the path to the queue based on queue/path availability, a carriage may wait in a magazine while determining its destination after unloading items, eliminating the concept of an unsure carriage and its state definition. At the same time, explicit definition of a carriage's direction is not required, because shaft direction is specified in the configuration definition.

The operational logic and physical representation of mobile carriage systems therefore may have a distinctly different state space and potential for differences in state-based complexity measures (and logical evolutions) as compared with conventional systems. Comparisons of mobile carriage and conventional systems with respect to complexity measures are therefore invalid, although internal comparisons of systems with common logic and sets of attributes used to define states remain valid.

Because of the construction and operation that follow from the operational logic of mobile carriage systems, complete evolutions tend to have very similar performance, regardless of the queue distribution. In conventional systems, throughput is a function of when carriages halt in the course of an evolution, which is dependent on the physical connectivity and queue distribution. For complete evolutions, where present connectivity exists between all queues and magazines corresponding to item types present in the queue distribution, halting of the carriage is dependent on the specific connectivity of the corresponding shaft. Like conventional systems, systems with mobile carriages require present connectivity between queues and magazines with corresponding item types (present connectivity in both the QM_{UP} and QM_{DOWN} matrices) for a complete evolution, with some exceptions. This connectivity implies that each queue has some down path to each magazine and each magazine has some up path to each queue. In a complete evolution, a carriage therefore always has a path available to a queue containing items that are deliverable after unloading. Since carriages are not associated with a single shaft and its limited connectivity, each carriage is essentially completely connected and will never halt, therefore resulting in near maximum throughput for any complete evolution.

The sole exception to the effect of disassociation of carriages and shafts on carriage halting occurs under similar circumstances we identified previously for a conventional system with higher amounts of connectivity. A carriage bases its selection of a destination queue on the state of the queues at the time it finishes an unloading operation. A carriage may therefore base its destination on the presence of a single item that has been loaded by another carriage in the time required to travel from magazine to queue. Upon arrival to an empty queue, the carriage waits indefinitely and is no longer utilized, thereby decreasing throughput. As with the conventional system, without some reservation or item buffer logic, carriages will have the potential of getting "tricked" into travel to an empty queue.

Since carriages typically do not halt in complete evolutions, resulting in similar throughput values for any configuration with the same number of carriages, there should be little correlation between complexity measures and performance. The similarity in performance values does not suggest however, that ranges of complexity values are necessarily limited.

For any configuration with two shafts and a fixed number of carriages, there exists only one performance/complexity value, barring evolutions in which carriages are tricked into entering empty queues, because one shaft must be up and the other must be down, meaning every queue and magazine must be connected to both shafts in a valid configuration. However, if the number of shafts increases, thereby increasing the configuration design space, a range of patterns is possible. Figure 13.1 presents the distribution of complete 2-3-4 evolutions with respect to the number of distinct states and throughput. Although we used 2-3-4 sized systems for analyses of conventional systems, the 2-3-4 sized mobile carriage systems are not necessarily used as an example system for the same reasons and their use is not intended to imply that comparisons should be made between the behaviors of conventional and mobile carriage systems. Mobile carriage systems with three shafts represent the simplest systems that result in non-trivial behavior. As with conventional systems, four magazines provide a large cross section of queue distributions and item variety, which is useful in identifying trends with respect to robustness.

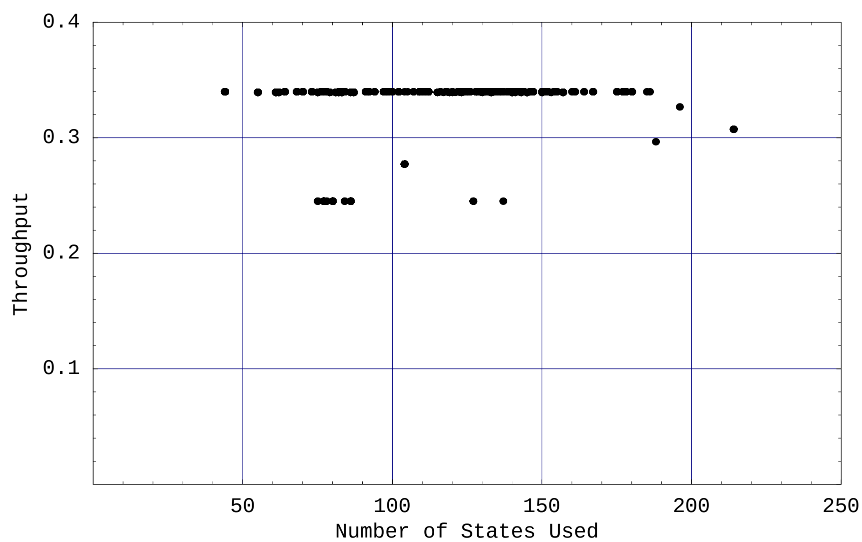


Figure 13.1: The distribution of complete 2-3-4 evolutions with respect to the number of states used and throughput using mobile carriages. Despite the appearance of some relationship, most evolutions have maximum throughput, but a range of complexity values.

Although there exist some evolutions in Figure 13.1 at lower throughput values that create a distribution that is vaguely reminiscent of those seen for conventional systems, these evolutions are misleading since they represent a small fraction of the total number of evolutions. Out of 2889 complete 2-3-4 evolutions, 2789, or 97%, have a throughput value greater than

0.33. The evolutions with lower throughput are the result of carriages tricked into halting.

Ignoring the effects of halting, the range of states entered is partly a function of item variety in the queue distributions. Every evolution with the minimum of 46 states entered corresponds to a queue distribution with a single item type. Carriages operate with a constant phase lag introduced by the common destination (and shaft) and operate in simple patterns, the number of which is dependent on the number of queues. For evolutions with the greatest number of unique states, the greatest item variety is a prerequisite. Item variety not only implies that more locations and therefore more states are entered, but also that logic specifying that carriages select the most abundant items in a queue creates more unique system patterns.

While item variety certainly has an effect on the number of states, evolutions with a common amount of item variety still have a range of values of the number of states. The range is largely a function of the actual connectivities of configurations and their effect on specific relative ratios of item types.

Since the number of states is variable with respect to a set of evolutions, but throughput remains fairly constant, the state complexity also exhibits no correlation with performance. Shown in Figure 13.2 is the distribution of 2-3-4 evolutions with respect to state complexity and throughput. A comparison with Figure 13.1 for the number of states reveals a nearly identical distribution, resulting from a similar temporal evolution length for all evolutions.

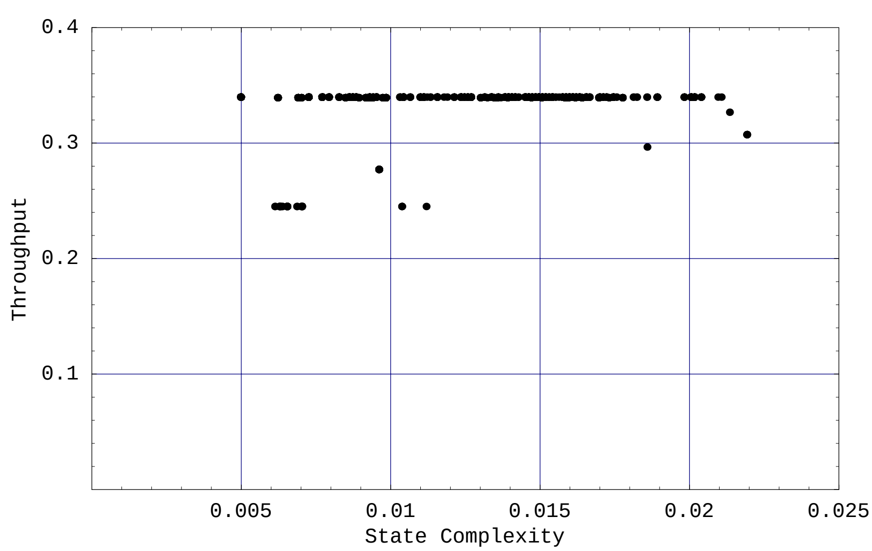


Figure 13.2: The distribution of complete 2-3-4 evolutions with respect to state complexity and throughput using mobile carriages. The shape of the distributions for the state complexity and number of states are nearly identical because the common throughput values of evolutions mean common temporal evolution lengths.

The distribution of 2-3-4 evolutions with respect to compressed state complexity and throughput, presented in Figure 13.3 also reveals a similar relationship between complexity and performance as in Figures 13.1 and 13.2. This distribution implies that both the logical evolution length and the temporal evolution length change proportionally and, since the temporal evolution length is approximately constant, the logical evolution length should be as well. The proportional change means that the logical complexity, the ratio of the logical and temporal evolution lengths, should be approximately constant. Figure 13.4, the three-dimensional distribution of evolutions with respect to logical complexity and throughput reveals that logical complexity is very nearly a constant. For the evolutions in which no halting of carriages occurs (2788 out of 2888 or 97% of evolutions), the logical complexity ranges from 0.0526 to 0.0532, a relative difference with respect to the maximum of 1.1%, compared with relative differences between the minimum and maximum with respect to the maximum of 76% for both the state and compressed state complexities.

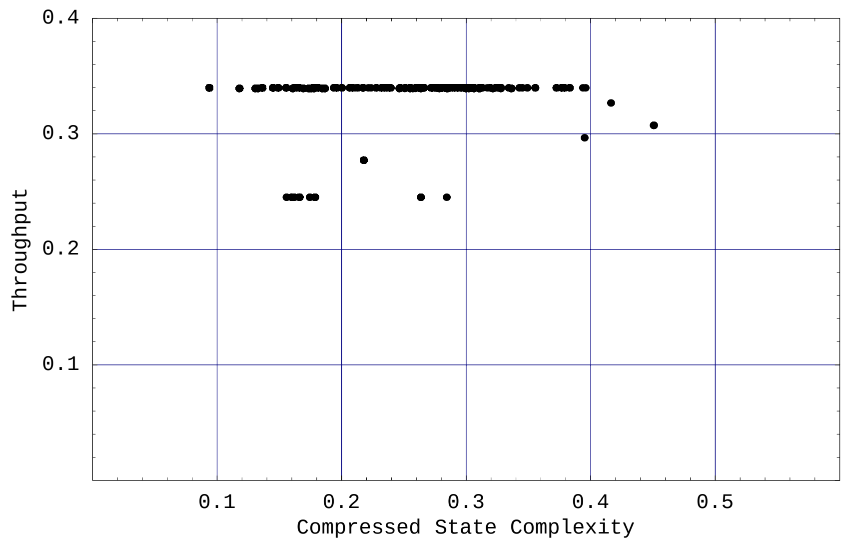


Figure 13.3: The distribution of complete 2-3-4 evolutions with respect to compressed state complexity and throughput using mobile carriages. Since the shapes of the distributions for state and compressed state complexity are similar to the shape of the distribution for the number of states, the logical evolution length must scale with the temporal evolutions length.

Of interest in Figure 13.4 is the distribution of evolutions with lower throughput values along a linear boundary, which is reminiscent of the boundary found in distributions for conventional systems. And, although there are relatively few evolutions defining this boundary, the reason for the relationship remains identical. When mobile carriages halt, correlations between logical complexity and throughput result as dissimilar carriage utilization results in longer evolution lengths, and therefore lower throughput, as well as fewer state transitions per system cycle and therefore lower logical complexity. The logical complexity distribution shares additional characteristics with distributions of conventional evolutions, with certain relative numbers of system elements.

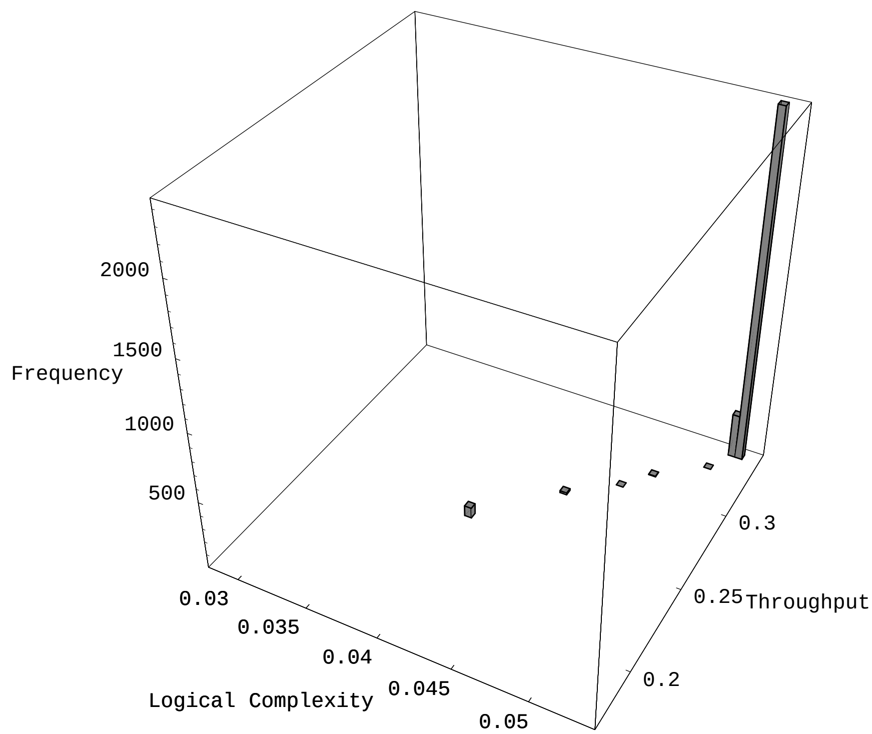


Figure 13.4: The three dimensional distribution of complete 2-3-4 evolutions with respect to compressed state complexity and throughput using mobile carriages, which includes frequencies of evolutions at blocks of values. The proportional scaling of logical and temporal evolution lengths is evident in this distribution, since most evolutions occupy a small range of logical complexity values.

In a system with three shafts, a valid configuration must consist of a single shaft of one direction and two in the other direction. While variations in the connectivities of the two shafts with a common direction are possible with sufficient queues or magazines, the single uni-directional shaft has complete connectivity to all queues and magazines. With its complete connectivity, the lone shaft is common to all carriages, making parallel operations impossible. For any sort of parallelism, there must be at least four carriages (and greater than one queue and magazine), permitting simple 1-2-1 subsystems to arise.

As with conventional systems, true parallelism - completely independent subsystems - never results in complete evolutions, assuming queues are initially identical with respect to the items they contain. However, significant parallel operations are possible in mobile carriage systems, if specific connectivity links the subsystems in the context of item variety. Figure 13.5 shows the temporal evolution history or configuration 14364390 2-4-3 with a (20-40-40) queue distribution. The connectivity of configuration 14364390 is presented in Figure 13.6 and shows that the first queue and magazine and the second queue and magazine are partly isolated. Along with de facto priority logic, Figure 13.5 shows that the carriages utilize this independence to operate without a phase lag. In order to be complete, present connectivity must be present between all queues and magazines via down shafts. This connectivity, along with common connectivity with respect to the third magazine, result in a configuration that operates in parallel until being "tripped" into a phase lag required for a complete evolution.

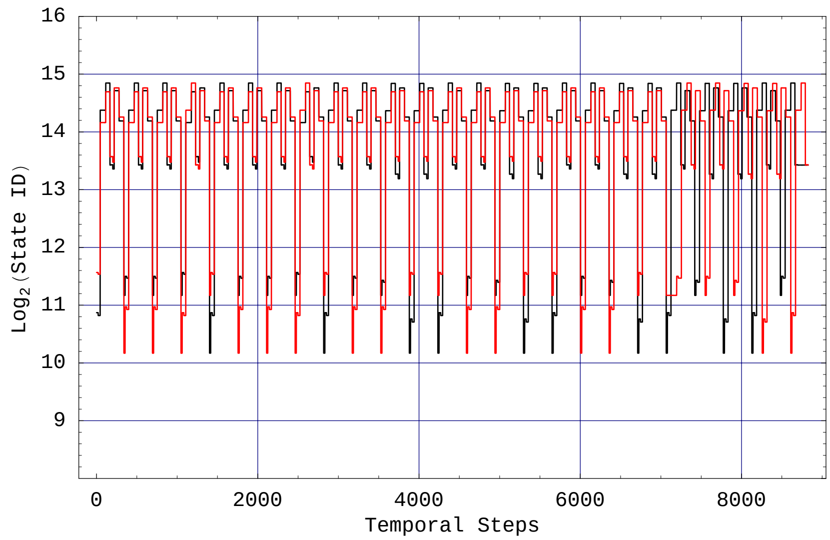


Figure 13.5: The temporal evolution history of configuration 14364390 2-4-3 with a (20-40-40) queue distribution. The two carriages in this configuration operate in parallel with no phase lag until being tripped into a phase lag by common connections.

Like the conventional systems, synchronous carriage operations without a phase lag mean that less state transitions per carriage cycle occur, resulting in lower logical complexity values. The relative amount of time that a system spends with and without a phase lag or, more accurately, the average number of state transitions per system cycle determine the equivalent number of carriages (logically) in the system, or the level of mimicry with respect to logical state transitions.

13.1 Robustness

The construction of mobile carriage systems results in similar throughput values for most complete evolutions for the set of logic described, because of the inherent immunity of car-

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

$$(SQ_{DOWN}) = \begin{pmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{pmatrix} \qquad (SM_{DOWN}) = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

$$(SQ_{UP}) = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 0 \end{pmatrix} \qquad (SM_{UP}) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

$$(QM_{DOWN}) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \qquad (QM_{UP}) = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

Figure 13.6: The incidence matrices for configuration 14364390 2-4-3 reveal that the first queue and magazine and second queue and magazine have some independence that can result in the absence of a phase lag between carriages. The subsystems must be ultimately connected in some manner for a complete evolution.

riages to halting (complete immunity if some form of reservation logic is employed). While the lack of halting does not provide any useful relationship between complexity and performance, complexity can still be applied towards robustness, particularly for measures related to the physical connectivity.

The tendency for carriages to have uniform utilization in complete evolutions in no way suggests that any configuration has maximum robustness because not all evolutions are necessarily complete. As for conventional systems, robustness is a function of the combination of physical connectivity in the context of variable input streams.

Figure 13.7 presents the robustness curves for the set of 2-3-4 evolutions, superposed with the normalized number of configurations at each populated level. Because each set of configurations at each level of robustness is composed only of complete evolutions, and complete evolutions have been shown to have maximal throughput, the robustness curve for throughput is, not surprisingly, an essentially flat line. The logical complexity robustness curve matches that of the throughput, because the logical and temporal evolution lengths scale in proportion in systems where parallel, in-phase carriages are impossible. For configurations in which phase lags may be absent (at least partially), such as the 2-4-2 systems with some degree of subsystem independence, a range of logical complexities may exist at the near constant throughput and their respective throughput and logical complexity robustness curves may begin to deviate.

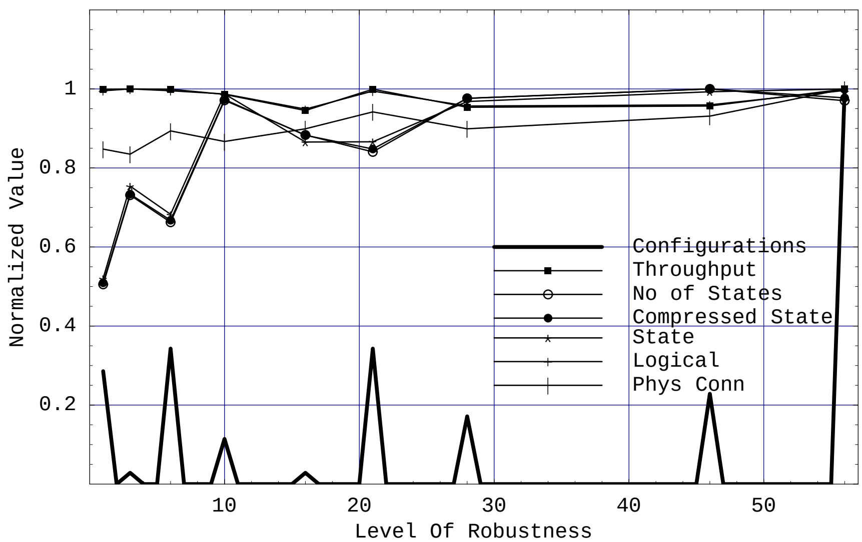


Figure 13.7: The robustness curves for throughput and various complexity measures for the 2-3-4 mobile carriage configurations, superposed with the configuration populations at each level. The throughput and logical complexity curves are nearly identical and share no practical correlation with robustness. The state-based complexity measures are also quite similar, since the logical and temporal evolutions lengths are fairly constant for all complete evolutions, and suggest a correlation with robustness.

The state-based complexity measures - the number of unique states, state complexity, and compressed state complexity - share nearly identical robustness curves, which are fundamentally different than those for the logical complexity and throughput. Although the distributions of evolutions with respect to state-based complexity measures reveal a range of possible values, the distributions alone provide no clue as to their relationship to robustness. However, there is a distinct trend in the state-based complexity measures with respect to robustness.

The explanation for lower state-based complexity measures at lower levels of robustness is largely a function of the definition of robustness and, more importantly, the attributes of a configuration that are responsible for complete evolutions. Because the number of complete evolutions, and therefore the level of robustness, is dependent on the number of magazines with absent connectivity to queues, configurations at lower levels of robustness are complete only for evolutions with less item variety in the queue distribution. A configuration with a level of robustness of 1 is therefore complete only when a single item type is present. As the level of robustness increases, corresponding configurations are able to completely transfer queue distributions with greater variety. But since item variety typically results in a greater number of states, and since throughput and evolution length remain relatively constant, then configurations that result in complete evolutions with a greater variety of item types will have higher mean numbers of states. That is, the number of states entered in an evolution of a configuration with a low level of robustness is approximately the same as that for a configuration with a high level of robustness for the common set of queue distributions. However, the complement of queue distributions will have greater item variety and increase the mean number of states for the robust configurations.

Looking with greater detail at the robustness curves in Figure 13.7 provides greater insight into the relationship between physical connectivity and robustness. For conventional systems, a configuration is complete for the set of queue distributions formed by the combination of item types equal to the number of relevant magazines - those magazines with present connectivity to all queues. With our constant 20% item distribution increment, there is therefore 1 queue distribution that results in a complete evolution for a configuration with 1 relevant magazine, 6 queue distributions that result in complete evolutions with 2 relevant magazines, 21 queue distributions with 3 relevant magazines, and, for configurations with 4 magazines, the greatest number of magazines for the systems analyzed, 56 queue distributions result in complete evolutions. For conventional systems, when considering completeness, and not uniqueness, a configuration exists only at the 1^{st}, 6^{th}, 21^{st}, or 56^{th} level of robustness, depending on the number of relevant magazines. However, Figure 13.7 for mobile carriage systems, reveals populations at additional levels of robustness. The explanation for these intermediate levels is related to the uni-directionality of shafts and the importance of distinguishing between sub-circuits with up and down shafts.

With respect to the sub-matrix describing connectivity between queues and magazines via down shafts, the relationship between the number of relevant magazines and the level of robustness is identical to that for conventional systems. If no downward shaft connects a magazine to a queue, then none of the items in that queue destined for that magazine can be delivered directly and all evolutions with queue distributions containing some fraction of this item type are incomplete. However, if an upward connection between a magazine and queue is absent, then an evolution may still be complete, although higher level patterns are required.

As an example, consider configuration 1019774 2-3-4, which has the incidence matrices presented in Figure 13.8 and a level of robustness of 46. The pathways and their directions for this configuration are illustrated in Figure 13.9. Inspection of the QM matrix, ignoring directionality, suggests that the configuration is capable of complete evolutions for any queue distribution, because of present connectivity between all queues and magazines. But the direction of the pathways is essential to robustness. Looking at the QM_{DOWN} matrix we again see present connectivity between all queues and magazines - a result of a single down shaft that must be completely connected for a valid configuration. The QM_{UP} matrix however, has an absent connection between the first queue and fourth magazine. This connectivity initially suggests that the configuration has three relevant magazines and any queue distribution containing item types corresponding to the fourth magazine results in an incomplete evolution because there is no return path from the fourth magazine to the first queue. And the evolution of this configuration with a queue distribution containing only items corresponding to the fourth magazine is certainly incomplete. However, if the number of items bound for the fourth magazine is sufficiently low, then complete evolutions result, as the system essentially utilizes a collateral circuit via higher order patterns.

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

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

$$(SQ_{DOWN}) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{pmatrix} \qquad (SM_{DOWN}) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

$$(SQ_{UP}) = \begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 0 & 0 \end{pmatrix} \qquad (SM_{UP}) = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$(QM_{DOWN}) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \qquad (QM_{UP}) = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Figure 13.8: The incidence matrices for configuration 1019774 2-3-4. There is no connection between the first queue and fourth magazine via up shafts, yet the configuration is capable of complete evolutions, when the number of items bound for the fourth magazine are sufficiently low, because higher order patterns return carriages to the first queue to deliver items to the fourth magazine.

Figure 13.10 presents the temporal evolution history of configuration 1019774 with an (80-0-0-20) queue distribution, with emphasis on the higher order patterns that enable the system to deliver items from the first queue to the fourth magazine. For this example evolution, for a carriage to transport items from the first queue to the fourth magazine, it must travel indirectly through the second queue and first magazine as it carries corresponding items. In order to complete the indirect circuit, there must be sufficient items corresponding to the first magazine (or second or third for this configuration) relative to the number of items bound for the indirectly connected fourth magazine, in the context of de facto priority logic. For configuration 1019774, queue distributions with more than 40% of item types corresponding to the fourth magazine result in incomplete evolutions. The ten of these queue distributions therefore result in the $56 - 10 = 46^{th}$ level of robustness. In general, a configuration with absent connectivity in the QM_{UP} sub-matrix must have sufficiently high ratios of item types

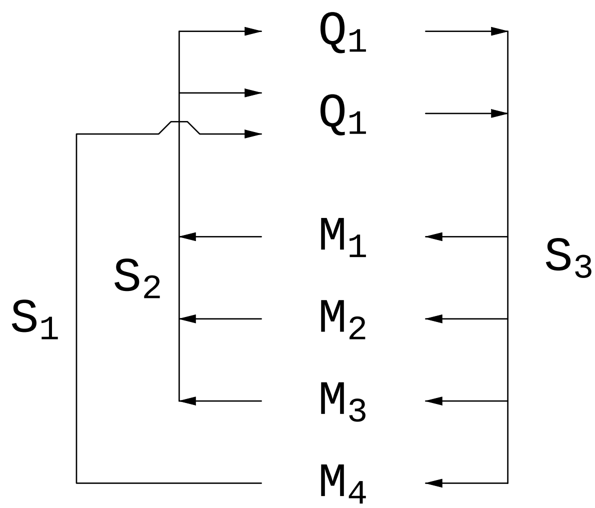


Figure 13.9: A schematic layout of configuration 1019774, showing the connections between queues and magazines and the directionality of shafts. The vertical layout of magazines and queues is not intended to suggest a physical reality - all queues are on one level and all magazines are on a common level.

required for collateral circuits with sufficient connectivity for repetition of the necessary higher order patterns.

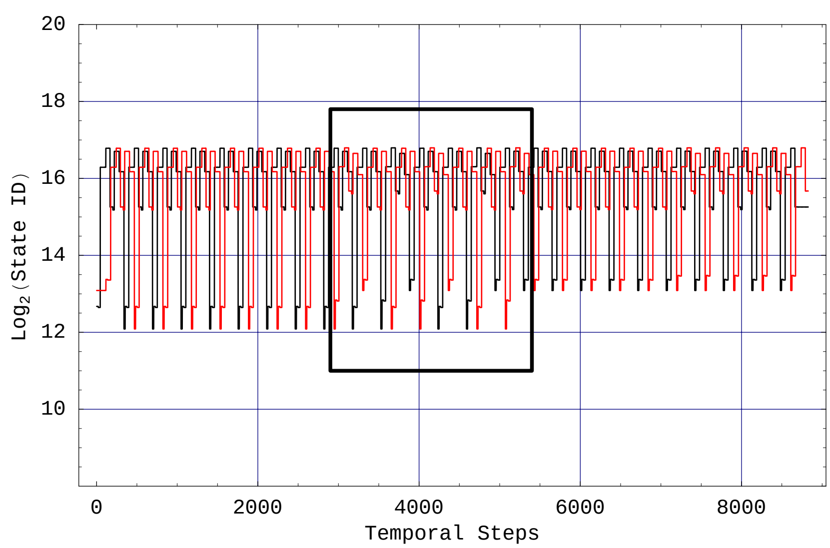


Figure 13.10: The temporal evolution history of configuration 1019774 2-3-4 with a (80-0-0-20) queue distribution. Higher order patterns are highlighted that describe carriages' circuits that result in a complete evolution

The relationship between connectivity and throughput indicates an important characteristic of these mobile carriage systems. Given that complete evolutions tend to have maximum throughput and *present* connectivity between queues and magazines with respect to both up and down shafts guarantees completeness, then additional connectivity does nothing to increase performance or adaptability. Increasing the number of connections between queues and magazines, while increasing average robustness and throughput in conventional systems, is inefficient in mobile carriage systems, only increasing the practical costs of additional connections.

Because we need to see how both the up and down sub-circuits behave in order to determine robustness, the use of average physical connectivity, when considering a configuration as a whole without regard to directional circuits, is not particularly useful for mobile carriage systems as a predictive measure of robustness (and, like other complexity measures, certainly not for throughput). However, understanding how the actual connectivity impacts robustness is useful and is applicable as a first-order filter for candidate configurations, given constraints on the control of the queue distribution.

A complete analysis of systems with mobile carriages, including investigation of the effects of parallel carriage/hatch operations and the absence of interlocks, is not performed here. Mobile carriage systems are intended only to illustrate how the associated assumed logic, in the context of physical connectivity and input streams, can significantly impact the relationships of complexity measures with performance and adaptability and therefore the practical applicability of these complexity measures in optimal searches of large design spaces. The lack of a practical relationship between complexity measures and performance or adaptability does not suggest that such useful relationships are inherently absent from systems with mobile carriages. Adaptation of the logic and system operation may introduce meaningful correlations, but require additional development not addressed here.

Chapter 14

Conclusions

Intuitively, we are able to distinguish between different types of behaviors. However, qualitative descriptions are too imprecise, especially in intra-class comparisons of behaviors. That is, can one thing be more chaotic than another? And to what extent? Based on theoretical measures such as algorithmic complexity and component counting and drawing from concepts from network topologies, we have created measures applicable to models of vertical material handling systems. Specifically, we examined deterministic models of conventional and unconventional naval weapons elevator systems with parallel, structured input streams that are simple enough to permit exhaustive characterization, yet large enough to exhibit a range of behaviors. Using these measures, we have demonstrated that relationships are present between behavior and performance and between behavior and robustness.

Algorithmic complexity, which is a significant basis for all dynamic measures and serves as a context for validating our intuitions regarding complexity, has been shown to exhibit little correlation to system performance. However, the practical implementations of algorithmic complexity do show correlations. For all dynamic measures, correlations result from the occurrence and timing of halting carriages, which is ultimately a function of physical connectivity and input streams. This effect is particularly apparent for the number of unique states, where no theoretical correlations are apparent and no other influences are present, such as normalizations by evolution lengths, which are inherently related to throughput.

Since model systems are defined in terms of physical connectivity, input streams, and control logic, it should not be a great surprise that the physical configuration play such an important role in the relationship between complexity and performance. However, it is extremely significant that a relatively simple potential complexity measure, such as the average physical connectivity, shows a strong relationship to performance. With a static, potential measure, the costs of simulations are avoided to a great extent, which can significantly reduce the time necessary to identify optimal solutions or can increase a search space.

In the design of complex systems with dynamic environments or, in the case of naval weapons elevator systems, an intractable state space of input streams, the concept of optimality is not limited to performance, but also must account for robustness. According to our interpretation of algorithmic complexity, optimal performance corresponds to the simplest evolutions, with little carriage interaction and delays. And, distributions of evolutions for our applied complexity measures often indicate this combination. However, this simple evolution/high performance point represents a very specific case for a combination of a certain physical connectivity and input stream, typically with little item variety. There is no guarantee that configurations with evolutions corresponding to this region will be good performers for other, more complex input streams. It is therefore necessary to consider the robustness of a configuration with a range of input streams. Our 'robustness curves', when superposed with configuration populations across possible levels of robustness, indicate that the most robust configurations generally have greater mean complexity, largely because robustness is a function of connectivity, specifically the presence and distribution of absent connections between queues and magazines and this connectivity dictates the amount of item variety that is possible for complete evolutions.

In recognizing the attributes of physical connectivity that relate to the level of robustness, we also found that optimality with respect to robustness by no means guarantees optimal performance. Given the constraints of connectivity imposed by a certain level of robustness, we must understand how specific connectivities relate to performance. While greater connectivity and more uniform distributions of connections tend to result in greater performance, the advantages of greater connectivity must be weighed against the practical costs and feasibility of more connections. Additionally, it is possible to obtain maximum or near maximum performance for all input streams without the necessity for complete connectivity. By avoiding redundant connectivity for shafts, we can obtain maximum performance for a given number of connections, while maintaining maximum robustness, yielding the most efficient configurations with respect to carriage utilization for a range of input streams.

The robustness of a configuration must also be considered in the context of the level of control of input streams. For the models used in this work, all queues were assumed to be identical and the set of input streams was constructed to consider a wide range of possibilities in order to simplify simulations while providing complete fundamental characterization of performance and behavior. Since physical connectivity and the set of possible input streams collectively determine the level of robustness, as the level of control of queues increases, connectivity can be decreased accordingly while still maintaining maximum robustness with respect to the set of input streams encountered, although this relationship indicates nothing about the resultant performance. However, the benefits associated with reducing connectivity based on the level of control of input streams comes at the price of reduced functionality - only those input streams suited to the physical configuration will result in complete evolutions, which may or may not raise any issues, depending on the life and intent of the system.

The relationships between complexity and performance and between complexity and robustness suggest that we design, or search for, configurations with maximum or near maximum complexity. However, in our deterministic system with the set of fixed evolution logic, chaotic behavior is never evident and the systems always operate in a simplistic regime, although the relative complexity in this regime varies. Evidence of operation in a simple regime lies in the values of the absolute complexity measures based on algorithmic complexity, such as logical and compressed state complexity, which indicate significant amounts of compression. Although our interpretation of the weapons elevator system results in operation strictly in a simplistic regime, the relationships in this regime are in agreement with evidence from complex systems, particularly from natural systems. The results therefore suggest a general applicability of behavior as part of an optimization method, where designed behavior is considered in the context of the environment. In evolutionary terms, systems with little environmental variability require simplicity in order to take advantage of environmental niches. With greater environmental variability, or even functionality, complex behavior is appropriate because of the associated mixture of adaptation and stability. Similarly, complex behavior is suitable in chaotic environments because of the self-organizing properties of complex systems. It is in these latter two situations that behavior-based optimization is intended to exhibit advantages over conventional methods, which do not consider behavior and provide little indication of performance and, more importantly, robustness beyond the limited set of simulations used in their application. But to validate these general principles, we require a formal "language" for complexity and behavior - a set of tools that enables identification of systems that would benefit from behavior-based optimization methods that, to date, does not exist. So, while the results of this work show that relationships can be present and offer the potential for behavior-based optimization techniques, they offer little with respect to general applicability beyond what we may imply from evidence in other complex systems.

The relationships for conventional systems show that physical connectivity and input streams, two of the three factors that define a system, have a collective impact on behavior, performance, and robustness. But the third factor, the operational logic, can also have a significant impact. Changes to the operational characteristics of shaft operations revealed no fundamental change in correlations, effectively changing only temporal effects, not logical. However, comparison of the effects of changes in operational logic in conventional systems on various complexity measures indicates interesting subtleties in the interpretation of algorithmic complexity which can affect the applicability of complexity as an optimization tool. Algorithmic complexity is defined as the length of the shortest program necessary for a complete description of an information stream. By this definition, the number of repetitions of a pattern is largely irrelevant, as the information required to describe the number of repetitions is typically negligible relative to the information required to describe the pattern. But if the intent of algorithmic complexity is to describe the amount of randomness or compression in an information stream, then the number of repetitions is relevant. The two interpretations of algorithmic complexity, as a relative or absolute measure or, for the model system, as a logical or temporal measure, indicates that a duality of algorithmic complexity is present that can affect its interpretation and application during comparisons of systems and their evolutions.

The control logic of mobile carriage systems, which is fundamentally different than that for conventional systems, resulted in a lack of correlations between behavior and performance or robustness. For these systems, an evolution is complete if all queues have at least one 'down' connection to each magazine corresponding to present item types and all of these magazines have at least one 'up' connection to each queue. Since carriages have no association with any particular shaft, the configuration that results in a complete evolution is effectively completely connected with respect to the input stream, meaning no carriage halts and therefore every complete evolution has maximum performance. So, despite variations in physical connectivity and input streams, the operational logic of mobile carriage systems effectively eliminates halting, the cause of correlations between complexity and performance and robustness in conventional systems, indicating that the operational logic must be suited to correlations if complexity-based optimization techniques are applicable.

The application of theoretical complexity measures and the identification of relationships between behavior and performance and between behavior and robustness in real systems, as well as their fundamental causes, represent preliminary steps towards the application of complexity to optimization. It is conjectured that complexity-based optimization is applicable as a stand-alone tool, filtering out configurations based on the behavior of evolutions in simulations, even as patterns emerge in the course of evolutions. It is envisioned that complexity can also be used as a quantitative measure of fitness in existing heuristic methods, such as genetic algorithms, as an additional optimizing filter. However, while heuristic methods rely on explicit simulation, static complexity measures offer the possibility of avoiding the costs associated with explicit simulations, which can significantly reduce the time necessary to find a suitable solution or, more importantly, can expand the size or detail of complex problems that can be solved.

Bibliography

[1] Reka Albert, Hawoong Jeong, and Albert-Laszlo Bararbasi. Error and attack tolerance of complex networks. *Nature*, 406:378–382, July 2000.

[2] Roger Anderson and Bob Sturges. Inherent behavior in complex systems. *Proceedings of 2004 Japan-USA Symposium on Flexible Automation*, 2004.

[3] Yaneer Bar-Yam. *Dynamics Of Complex Systems*. Westview Press, 1997.

[4] Albert-Laszlo Bararbasi and Eric Bonabeau. Scale-free networks. *Scientific American*, pages 50–59, May 2003.

[5] G Barney and S dos Santos. *Elevator Traffic Analysis Design and Control*. Peter Peregrinus Ltd, 1985.

[6] R. Beckers, O.E. Holland, and J.L. Deneubourg. From local actions to global tasks: Stigmergy and collective robotics. In *Proceedings of Artificial Life IV*, pages 181–189, 1994.

[7] Charles Bennett. How to define complexity and why. In *Complexity, Entropy, and the Physics of Information*, pages 137–148, 1990.

[8] Eric Bonabeau, Guy Theraulaz, Eric Arpin, and Emmanuel Sardet. The building behavior of lattice swarms. In *Proceedings of Artificial Life IV*, pages 307–312, 1994.

[9] R.F.I Cancho, C. Janssen, and R.V. Sole. Topology of technology graphs: Small world patterns in electronic circuits. *Physical Review E*, 64, 2001.

[10] R.F.I Cancho and R.V. Sole. The small world of human language. *Proceedings of the Royal Society of London, Series B: Biological Sciences*, 268:2261–2265, 2001.

[11] Gregory Chaitin. *Algorithmic Information Theory*. Cambridge University Press, 1987.

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

[13] Gregory Chaitin. *Exploring Randomness*. Springer, 2001.

[14] G. Clark, P. Mehta, and R. Frowse. Knowledge-based elevator controller. *IEEE Conference Publication*, 1(389):42–47, 1994.

[15] Robert Crites and Andrew Barto. Elevator group control using multiple reinforcement learning agents. *Machine Learning*, 33:235–262, 1998.

[16] James Crutchfield, Melanie Mitchell, and Rajarshi Das. The evolutionary design of collective computation in cellular automata. *Santa Fe Institute Working Papers*, 1998. Paper Number 98-09-080.

[17] Stephanie Forrest and T. Jones. Modeling complex adaptive systems with echo. In *Complex Systems: Mechanism of Adaptation*, 1994.

[18] M. Gardner. Mathematical games. the fantastic combinations of john conway's new solitaire game 'life'. *Scientific American*, pages 120–123, October 1970.

[19] Sylvie Geisendorf. Genetic algorithms in resource economic models: A way to model bounded rationality in resource exploitation? *Santa Fe Institute Working Papers*, 1999. Paper Number 99-08-058.

[20] James Gleick. *Chaos: Making a New Science*. Penguin Books, 1987.

[21] Takashi Hikihara and Shinichi Ueshima. Emergent synchronization in multi-elevator system and dispatching control. *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, E80-A(9):1548–1553, September 1997.

[22] Ming Ho and Brent Robertson. Elevator group supervisory control using fuzzy logic. In *Canadian Conference on Electrical and Computer Engineering*, volume 2, pages 825–828, 1994.

[23] Jessica Hodgins and David Brogan. Robot herds: Group behaviors for systems with significant dynamics. In *Proceedings of Artificial Life IV*, pages 319–324, 1994.

[24] John Holland. *Adaptation inNatural and Artificial Systems*. Ann Arbor: University of Michigan Press, 1975.

[25] Wim Hordijk, James Crutchfield, and Melanie Mitchell. Mechanisms of emergent computation in cellular automata. *Santa Fe Institute Working Papers*, 1998. Paper Number 98-04-034.

[26] Andrew Ilachinski. *Cellular Automata: A Discrete Universe*. World Scientific, 2001.

[27] H. Jeong, S.P. Mason, A.L. Barabasi, and Z.N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411:41–42, 2001.

[28] W. Jianru, L. Chunjiang, and L. Hongchi. Nn elevator group control method. *Elevator World*, 50(2):148–154, February 2002.

[29] Shareen Joshi, Jeffrey Parker, and Mark Bedau. Financial markets can be at sub-optimal equilibria. *Santa Fe Institute Working Papers*, 1999. Paper Number 99-03-023.

[30] Stuart Kauffman. *The Origins of Order, Self-Organization and Selection in Evolution*. Oxford University Press, 1993.

[31] Stuart Kauffman. *At Home in the Universe: the Search for Laws of Self-Organization and Complexity*. Oxford University Press, 1995.

[32] C. Kim, K. Seong, H. Lee-Hyung, and J. Kim. Design and implementation of fegcs: Fuzzy elevator group control system. In *New Frontiers in Fuzzy Logic and Soft Computing Biennial Conference of the North American Fuzzy Information Processing Society*, pages 109–113, 1996.

[33] C. Kim, K. Seong, H. Lee-Kwang, J. Kim, and Y. Lim. Fuzzy approach to elevator group control system. *IEEE Transactions on Systems, Man, and Cybernetics*, 25(6):985–990, June 1995.

[34] J. Kim and B. Moon. Adaptive elevator group control with cameras. *IEEE Transactions on Industrial Electronics*, 48(2):377–382, April 2001.

[35] Christopher Langton. Studying artificial life with cellular automata. *Physica D*, 22:120–149, 1986.

[36] Christopher Langton. Life at the edge of chaos. In *Artificial Life II*, pages 41–91, 1990.

[37] Bennett Levitan, Jose Lobo, Stuart Kauffman, and Richard Schuler. Optimal organizational size in a stochastic environment with externalities. *Santa Fe Institute Working Papers*, 1999. Paper Number 99-04-024 submitted to Organization Science.

[38] Wentian Li and Norman Packard. The structure of the elementary cellular automata rule space. *Complex Systems*, 4:281–297, 1990.

[39] Wentian Li, Norman Packard, and Chris Langton. Transition phenomena in cellular automata rule space. *Physica D*, 45:77–94, 1990.

[40] Harold V. McIntosh. Wolfram's class iv automata and a good life. *Physica D*, 45:105–121, 1990.

[41] K Mimaki, S Markon, H Kita, Y Komoriya, and Y Nishikawa. Modeling and analysis of complex traffic in buildings. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 4, pages 589–594, 1999.

[42] Melanie Mitchell. *An Introduction to Genetic Algorithms*. The MIT Press, 1998.

[43] John Nash. Equilibrium points in n-person games. In *Proceedings of the National Acadamy of Sciences*, 1950.

[44] G Newell. Strategies for serving peak elevator traffic. *Transportations Research Part B-Methodological*, 32(8):583–588, November 1998.

[45] Norman Packard. Adaptation towards the edge of chaos. In *Dynamic Patterns in Complex Systems*, pages 293–301. World Scientific, 1988.

[46] Grantham Pang. Elevator scheduling system using blackboard architecture. *IEE Proceedings. Part D, Control Theory and Applications*, 138(4):337–346, July 1991.

[47] D Pepyne and C Cassandras. Optimal dispatching control for elevator systems during uppeak traffic. *IEEE Transactions on Control Systems Technology*, 5(6):629–643, November 1997.

[48] Craig Reynolds. Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics*, 21(4):25–34, July 1987. ACM SIGGRAPH '87 Conference Proceedings, Anaheim, CA, July, 1987.

[49] Craig Reynolds. Competition, coevolution, and the game of tag. In *Proceedings of Artificial Life IV*, pages 59–69, 1994.

[50] M. Schlemmer and S. Argrawal. A computational approach for time-optimal planning of high-rise elevators. *IEEE Transactions on Control Systems Technology*, 10(1):105–111, January 2002.

[51] Manfred Schoeder. *Fractals, Chaos, Power Laws: Minutes from an Infinite Paradise*. W.H. Freeman, 1991.

[52] Joris Schroeder. Down-peak handling capacity of elevator groups. *Elevator World*, 33(11):48–53, November 1985.

[53] Claude Shannon and Warren Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, 1971.

[54] Benjamin Shargel, Hiroki Sayama, Irving R Epstein, and Yaneer Bar-Yam. Optimization of robustness and connectivity in complex networks. *The American Physical Society*, 90(6), February 2003.

[55] George Strakosch. *Vertical Transportation*. John Wiley and Sons, 1983.

[56] Steven Strogatz. *Sync: The Emerging Science of Spontaneous Order*. Hyperion Press, 2003.

[57] Kazuo Sugihara and Ichiro Suzuki. Distributed algorithms for formation of geometric patterns with many mobile robots. *Journal of Robotic Systems*, 13(3):127–139, 1996.

[58] T. Tobita, A. Fujino, K. Yoneda, and T. Ueshima. An elevator characterized group supervisory control system. In *IECON Proceedings (Industrial Electronics Conference)*, volume 3, pages 1972–1976, 2000.

[59] S. Valverde, R. Ferrer Cancho, and R.V. Sole. Scale-free networks from optimal design. *Europhysics Letters*, 60:512–517, 2002.

[60] John von Neumann. *Theory of Self-Reproducing Automata*. University of Illinois Press, 1966. ed. Arthur W. Burks.

[61] Duncan Watts and Steven Strogatz. Collective dynamics of 'small-world networks. *Nature*, 393:440–442, June 1998.

[62] Stephem Wolfram. Statistical mechanics of cellular automata. *Rev Mod Phys*, 55(8), July 1983.

[63] Stephen Wolfram. Universality and complexity in cellular automata. *Physica D*, 10:1–35, 1984.

[64] Stephen Wolfram. *A New Kind of Science*. Wolfram Media, Inc., 2002.

[65] William Wooters and Christopher Langton. Is there a sharp phase transition for deterministic cellular automata? *Physica D*, 45:95–104, 1990.

[66] Kenji Yoneda, Toshimitsu Tobita, Akira Onuki, and Hideo Kajiura. Multi-objective elevator supervisory control system with individual floor-situation control. *Elevator World*, 47(5):92–95, 1999.

[67] Q. Zong, X. Shang, Y. Yue, and X. Lei. Traffic pattern recognition of elevator group control systems. *Control and Decision*, 16(2):163–166, March 2001.

[68] Wojciech H. Zurek. Algorithmic information content, church-turing thesis, physical entropy, and maxwell's deamon. In *Complexity, Entropy, and the Physics of Information*, pages 73–90, 1990.

Appendix A

Baseline Operation Cycle

Table A.1: The strike-down sequence and cycle times for the modified baseline configuration consisting of a single queue, shaft, and magazine (located on the 7th deck). At the start of the cycle, the carriage is assumed to be unloaded at the main deck. The upper ballistic hatch is open and the lower ballistic hatch and magazine door are closed. In this sequence, the lower hatch is assumed to close over the carriage when it stops in the lower shaft. This operation represents a divergence from the actual sequence to reflect potential advances in system hardware. The individual cycle times have also been changed from existing baseline values to reflect these advances.

Step	Operation	Cycle Time (s)	Cumulative Time (s)
1	Load weapons at queue	20	20
2	Carriage travel from the main deck to the upper shaft, lower way point at the 3rd deck (2 decks at 7 seconds transit time per deck)	14	34
3	Upper ballistic hatch closes and locks	18	52
4	Lower ballistic hatch unlocks and opens	18	70
5	Carriage travel from the upper shaft, lower way point to the destination magazine level at the 7th deck (4 decks at 7 seconds transit time per deck)	28	98
6	Lower ballistic hatch closes and locks	18	116
7	Magazine door unlocks and opens	10	126
8	Unload weapons at destination magazine	20	146
9	Magazine door closes and locks	10	156
10	Lower ballistic hatch unlocks and opens	18	174
11	Carriage travel from the 7th deck to the upper shaft, lower way point at the 3rd deck (4 decks at 7 seconds transit time per deck)	28	202
12	Lower ballistic hatch closes and locks	18	220
13	Upper ballistic hatch unlocks and opens	18	238
14	Carriage travel from the upper shaft, lower way point to the main deck	14	252
15	Completion of one cycle with carriage ready to load at the main deck. Total cycle of 252 seconds		

Appendix B

Model Validation

B.1 Baseline Sequence

Validation of the weapons elevator model and simulation is performed with a comparison of the modified baseline sequence in Table A.1 to a trace of the evolution of states in time taken from the model. In the following single cycle of the evolution of the single queue, shaft, and magazine system (configuration 3 1-1-1 (100)), more system states than the 11 used to define system states in evolutions presented throughout the text are involved in order to provide the level of detail required for comparison with the baseline sequence. The 32 states used for the baseline cycle (and their bit position in the graphical representation) are enumerated in Table B.1. Any column containing colored cells describes the state of a binary variable, where white cells represent a false value and black cells represent a true value. Columns containing numbers represent the value of timers, location values, or other non-binary states.

When multiple carriages are present in an evolution, information is partitioned with respect to carriages such that the states in Table B.1 comprise a packet corresponding to a particular carriage. While the number of columns required to describe an evolution is clearly not constant and depends on the number of carriages, the size of packets for a particular configuration size is also not constant. Since the magazine door states corresponding to a particular shaft are included in each packet, the number of columns required to describe the state of a carriage is dependent on the number of magazines. However, since all respective magazine door states are included in a packet regardless of the connectivity of the corresponding shaft and magazines, the number of states in each packet is identical. The number of states required to describe a model in this format is therefore presented in Equation B.1, where s is the number of shafts/carriages and m represents the number of magazines.

$$States = s\,(26 + 6m) \tag{B.1}$$

Table B.1: The 32 states used in the trace of the baseline, single queue, shaft, and magazine system. In columns containing only color, a white cell indicates the state is false and black cells represent true. Numbered cells describe the value of a non-binary state such as a timer or location number.

Bit	State
1	Destination type
2	Destination number
3	Current location type
4	Current location number
5	Carriage loading
6	Loading timer
7	Upper ballistic hatch closed
8	Upper ballistic hatch opening
9	Upper ballistic hatch opening timer
10	Upper ballistic hatch open
11	Upper ballistic hatch closing
12	Upper ballistic hatch closing timer
13	Lower ballistic hatch closed
14	Lower ballistic hatch opening
15	Lower ballistic hatch opening timer
16	Lower ballistic hatch open
17	Lower ballistic hatch closing
18	Lower ballistic hatch closing timer
19	Magazine door closed
20	Magazine door opening
21	Magazine door opening timer
22	Magazine door open
23	Magazine door closing
24	Magazine door closing timer
25	Carriage unloading
26	Unloading timer
27	Up direction
28	Down direction
29	Up movement
30	Down movement
31	Shaft movement timer
32	Carriage level

Included on the right edge of the evolution are indicators of states changes corresponding to those in the baseline sequence, where the numbers describe the states in Table A.1. We can therefore establish the validity of the model by comparing the state of the system at each state change, knowing the order of states from Table B.1, with the corresponding number in Table A.1 as well as comparing the step in the evolution at each state change with the cumulative time in Table A.1. A cursory comparison of the steps corresponding to state changes in the model evolution and the baseline sequence reveals discrepancies that initially serve to invalidate the model. However, closer inspection reveals that the differences are trivial and irrelevant to model validity. For six of the state changes in the evolution, two time steps are required for a complete state change. This transition is a direct result of the serial nature of the programming used in the model. In a single scan of checks for state changes, if an updated value affects states checked earlier in the code, then an additional time step is required for the earlier code to respond to the change. The occurrence of these delays, described in Table B.2 for the baseline scenario, is a function of the sequence of code presented in Table B.3 that performs checks for changes in state values. For instance, when the simulation has detected that the lower ballistic hatch has finished opening (the end of operation 4 in Table A.1), the carriage does not simultaneously begin downward movement because the checks that permit downward movement are performed prior to the checks for updates in hatch states. Only in the subsequent check for carriage movement in the following cycle does the carriage movement state change and movement begin. Since the effects of the delays are cumulative and there are 6 instances in this evolution that result in delayed state changes, the length of a cycle in described by the simulation is 258 steps, compared with the 252 steps defined by the baseline scenario.

Table B.2: Scenarios that result in an apparent delay in state transitions. These delays are a function of the serial nature of programming used.

Step	Delayed transition
70	Lower ballistic hatch finishes opening, then carriage starts movement
117	Lower ballistic hatch finishes closing, then magazine door starts opening
128	Magazine door finishes opening, then carriage begins unloading
177	Lower ballistic hatch finishes opening, then carriage starts movement
224	Lower ballistic hatch finishes closing, then upper ballistic hatch starts opening
244	Upper ballistic hatch finishes opening, then carriage travels from the 3rd deck to the main deck

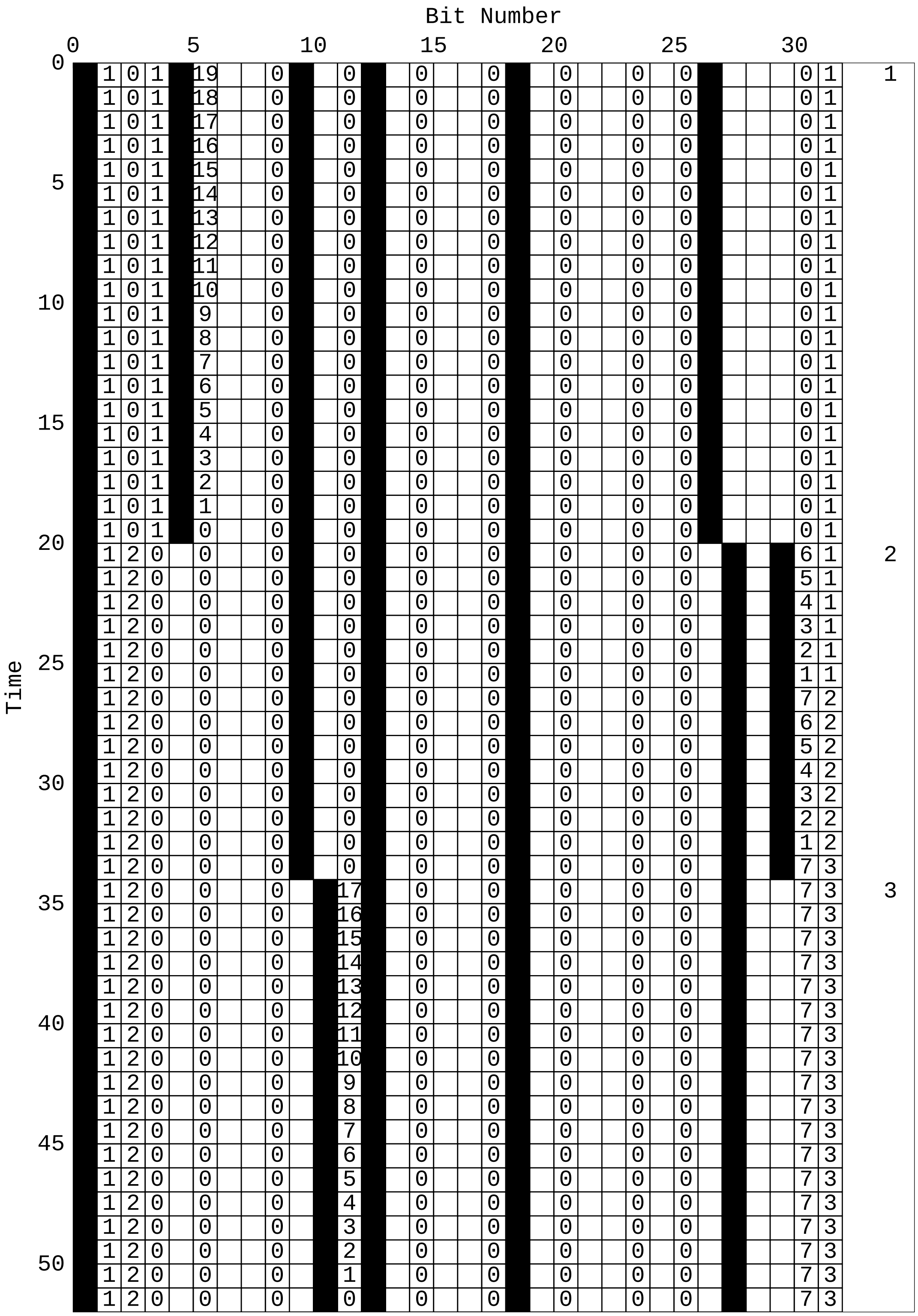
While the effects of serial programming that result in a discrepancy between “actual” and modeled sequences of state changes can be remedied in a number of ways, the differences are

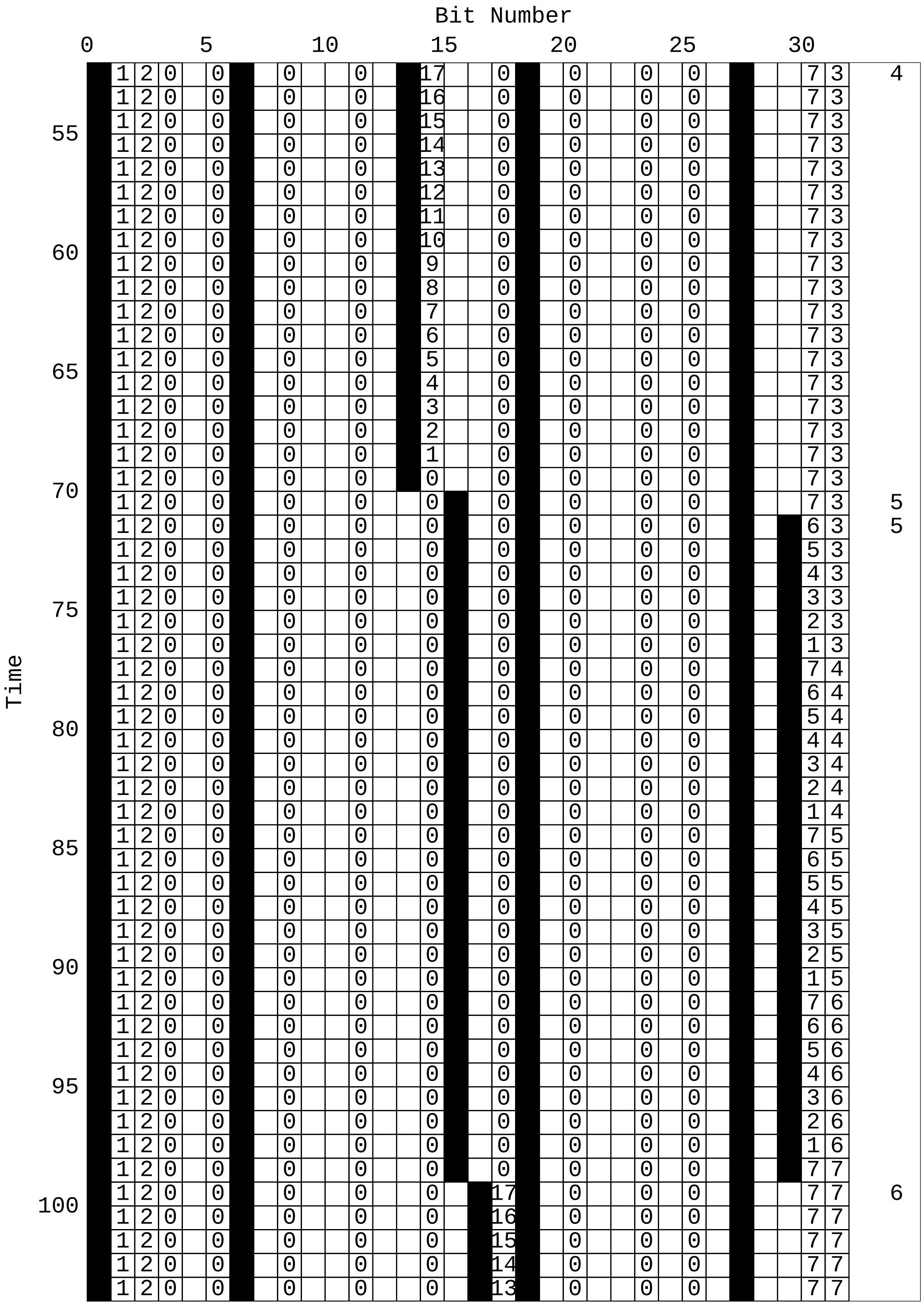
Table B.3: The serial sequence of states used in the programming for the elevator model. Two states that change simultaneously in the baseline sequence may change on different time steps in the model if a change made in a latter module affects a change in an earlier module.

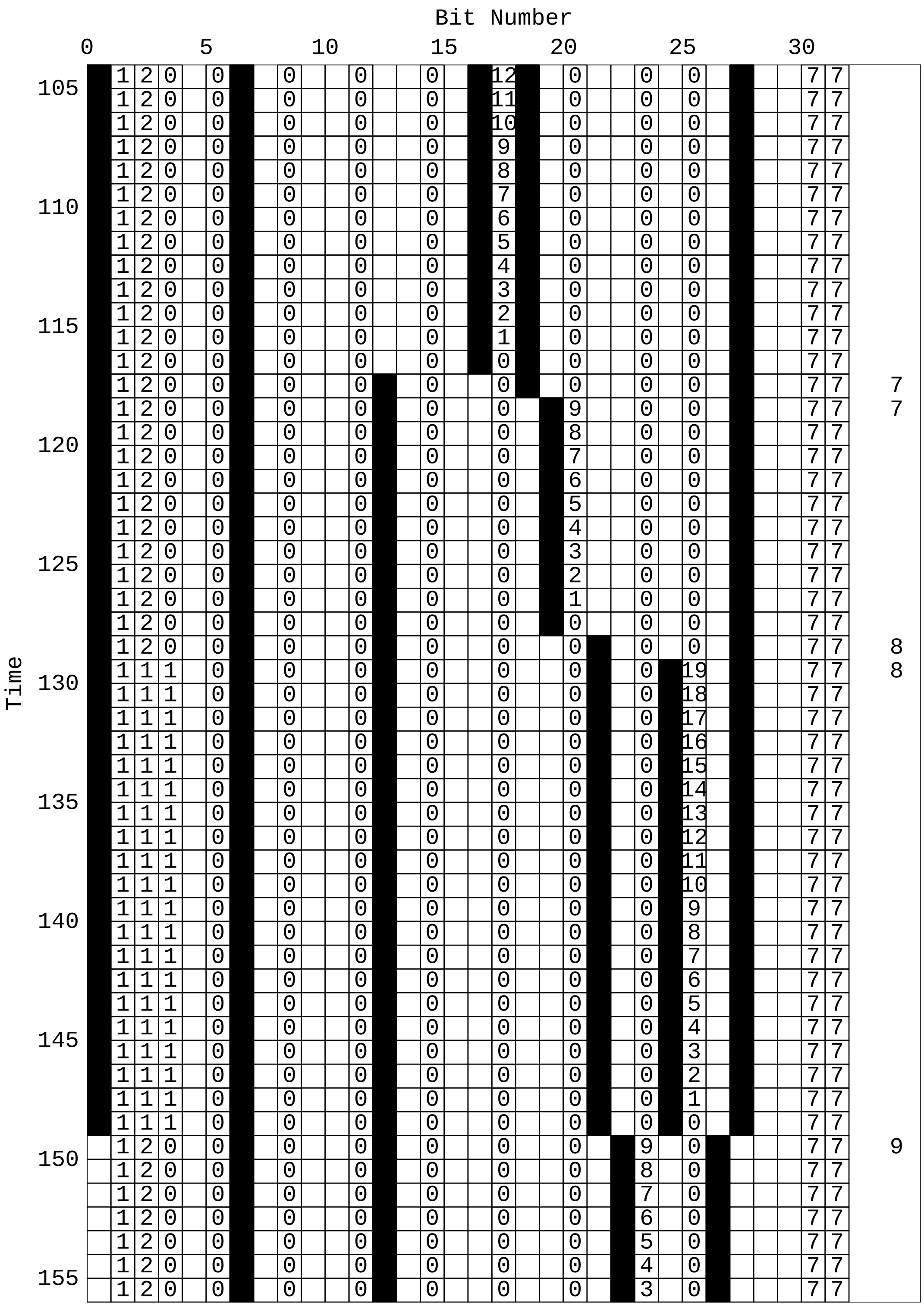
Checks for updating carriage states
CheckForStartLoading
CheckForFinishLoading
CheckForStartUnloading
CheckForFinishUnloading
CheckForUpMovement
CheckForDownMovement
CheckToReserveMagazine
CheckForSelectQueue
CheckForRequestOpenUpperBallisticHatch
CheckForRequestOpenLowerBallisticHatch
CheckForRequestOpenMagazineDoor
CheckIfMovingUpOrDown

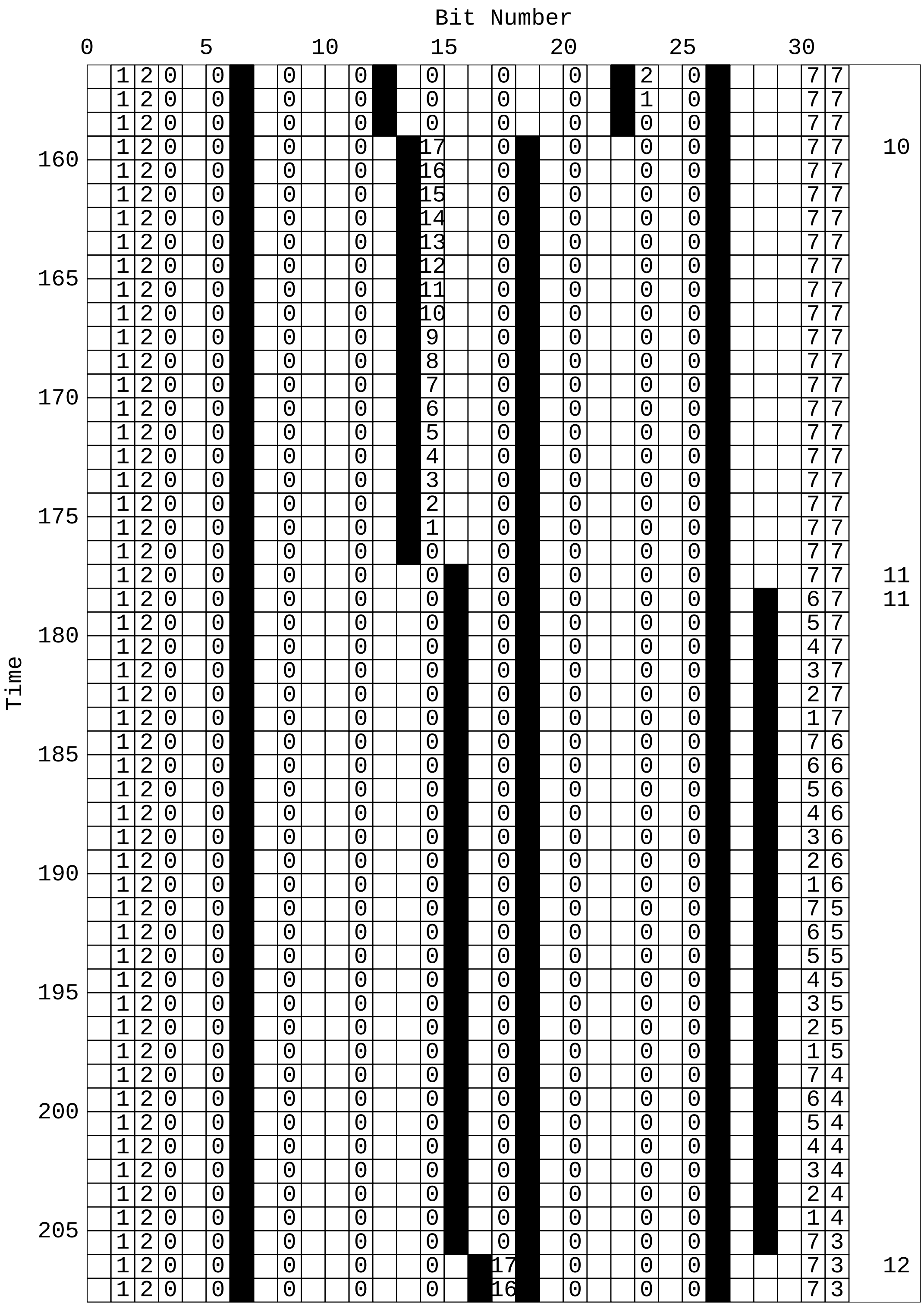
Checks for updating hatch/door states
CheckForMagazineDoorToStartOpening
CheckForMagazineDoorToFinishOpening
CheckForMagazineDoorToStartClosing
CheckForMagazineDoorToFinishClosing
CheckForUpperBallisticHatchToStartOpening
CheckForUpperBallisticHatchToFinishOpening
CheckForUpperBallisticHatchToStartClosing
CheckForUpperBallisticHatchToFinishClosing
CheckForLowerBallisticHatchToStartOpening
CheckForLowerBallisticHatchToFinishOpening
CheckForLowerBallisticHatchToStartClosing
CheckForLowerBallisticHatchToFinishClosing

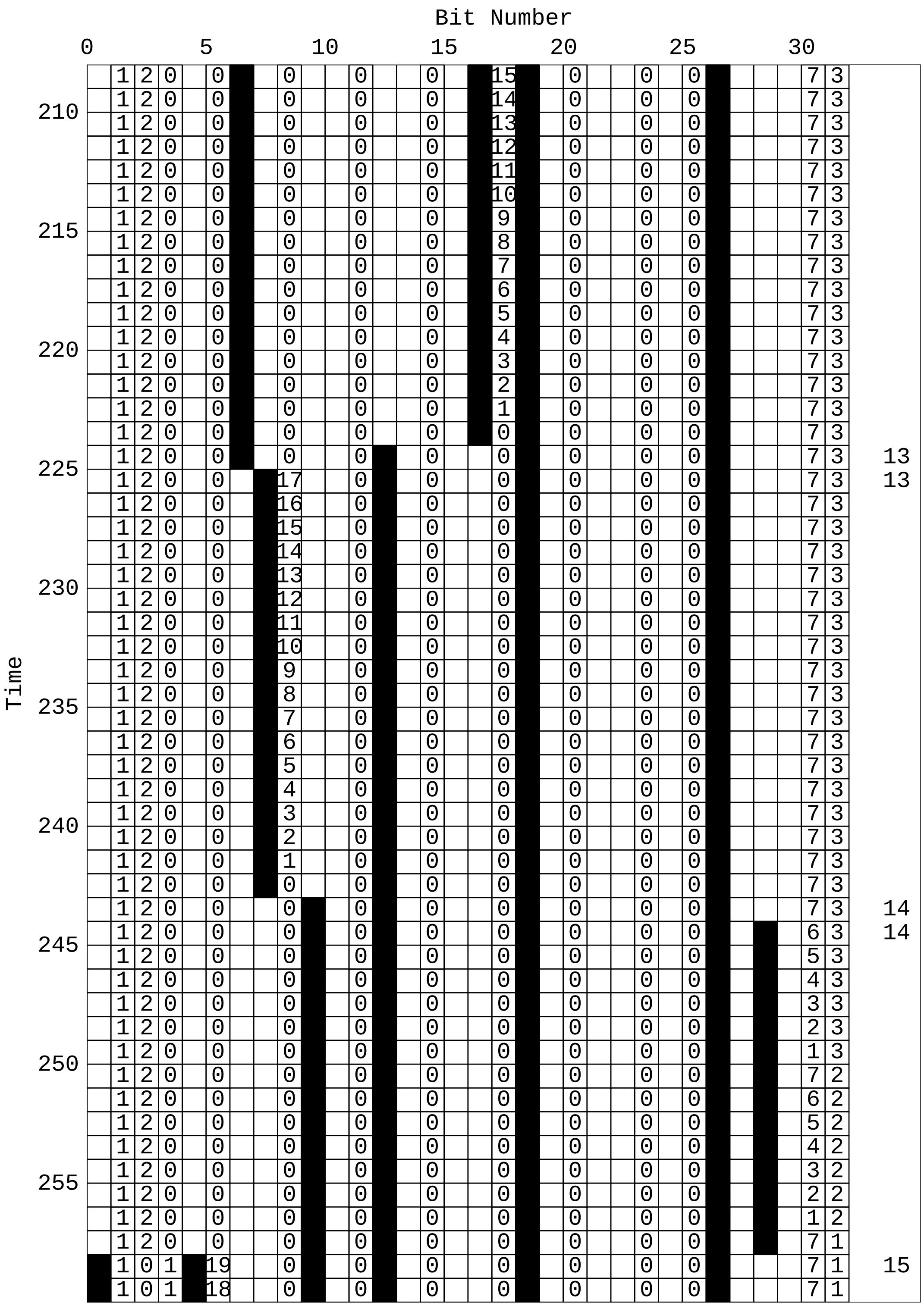
irrelevant with respect to the primary objectives of the work. Since the model is deterministic and phase lags between the sequences of carriage states remain fixed after being introduced, transition delays do not result in additional, or fewer system states. Since our intent is not to model the particular system defined by the baseline scenario, but to investigate relationships between complexity, performance, and robustness, these delays are therefore irrelevant to the validity of the model.


Bit Number
Time
1
2
3


Bit Number
0
5
10
15
20
25
30
Time
55
60
65
70
75
80
85
90
95
100
4
5
5
6

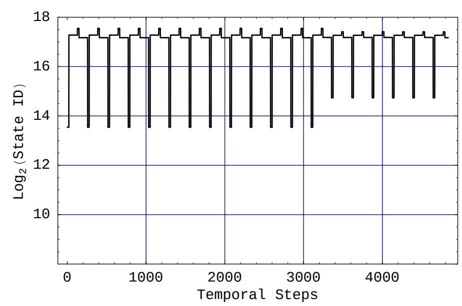

Bit Number
0
5
10
15
20
25
30
Time
105
110
115
120
125
130
135
140
145
150
155


Bit Number
Time
10
11
11
12

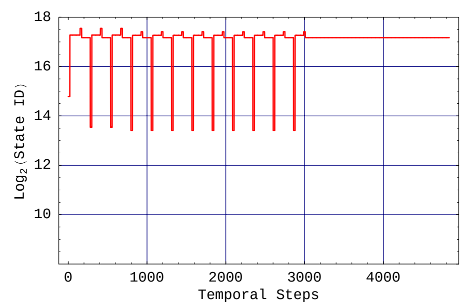

Bit Number
0
5
10
15
20
25
30
Time
210
215
220
225
230
235
240
245
250
255
13
13
14
14
15

B.2 Validation with Greater Complexity

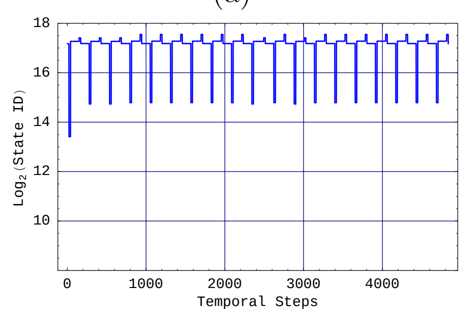
Using similar techniques for tracing the evolution of the baseline scenario, we can investigate more complex evolutions involving greater numbers of states that also allow us to validate the model logic. In particular, we can use the trace to identify problems associated with system operation not obvious or predictable from the rule sets, such as the "halting" problem identified in analyses, in which a carriage is "tricked" into selecting a queue that is emptied prior to the carriage's arrival. A view of the temporal evolution history for each carriage in this scenario is provided in Figure B.1. In this illustrative case (262143 2-3-4 (0-0-40-60)) of the completely connected 2-3-4 system, the second carriage halts at approximately the 3,100th step. However, an analysis of the following detailed evolution trace in Figure B.2 reveals that the cause for the halting of the second carriage is set up in the sequence of events prior to the halting. To observe these events, the evolution trace presents the states from the 2,965th step to the 3,120th step. As with the baseline evolution, key steps in the evolution relevant to the halting scenario are presented on the right margin of the evolution. For simplicity of the trace, descriptions of these steps are presented in Table B.4 that include the relevant bit numbers associated with state changes.



(a)



(b)



(c)

Figure B.1: The individual carriage state histories of the *(a)* first carriage, *(b)* second carriage, and *(c)* third carriage for configuration 262143 with a 0-0-40-60 queue distribution. The second carriage halts at approximately the 3,100th step, despite the complete connectivity of the configuration when the second carriage enters an empty queue. A detailed trace of the evolution illustrates the course of events and the logic that results in this scenario.

Since there are significantly more states involved in this evolution than in the baseline scenario because of the number of carriages and magazines, columns containing numerical values are difficult to interpret. As a result, the descriptions of these numbers have been replaced by shaded cells, where the intensity of the shading indicates the value of the state relative to the maximum value the state can assume. For instance, as a timer counts down from a value of 20, the cell will lighten steadily from a black cell through 18 shades of gray and result as a white cell.

Table B.4: A description of the sequence and timing of events that result in a carriage that enters an empty queue. The state numbers refer to key steps indicated in the detailed evolution trace.

State	Step	Event	Relevant bits
1	2968	First carriage begins unloading in the fourth magazine	3, 4, 43, 44
2	2988	First carriage finishes unloading and selects the first queue as its destination. At this time, there is a single item in the first queue, which corresponds to the third magazine	1, 2, 43, 44
3	2989	Second carriage starts unloading in the third magazine	53, 54 93, 94
4	3009	Second carriage finishes unloading and selects the first queue as its destination as well because the lone item on which the first carriage based its queue selection has not been picked up by the first carriage yet, so the second carriage is under the impression the item is available. The first carriage is waiting at the seventh deck while the magazine door to its shaft closes	47, 49 50, 51, 52, 93, 94
5	3097	The first carriage starts loading the last item in the first queue. The second carriage is at the third deck, waiting for its corresponding upper ballistic hatch to finish opening	3, 4, 5, 6, 58, 59, 95, 99, 100
6	3117	The first carriage finishes loading and the the second carriage arrives at an empty queue	5, 6, 100

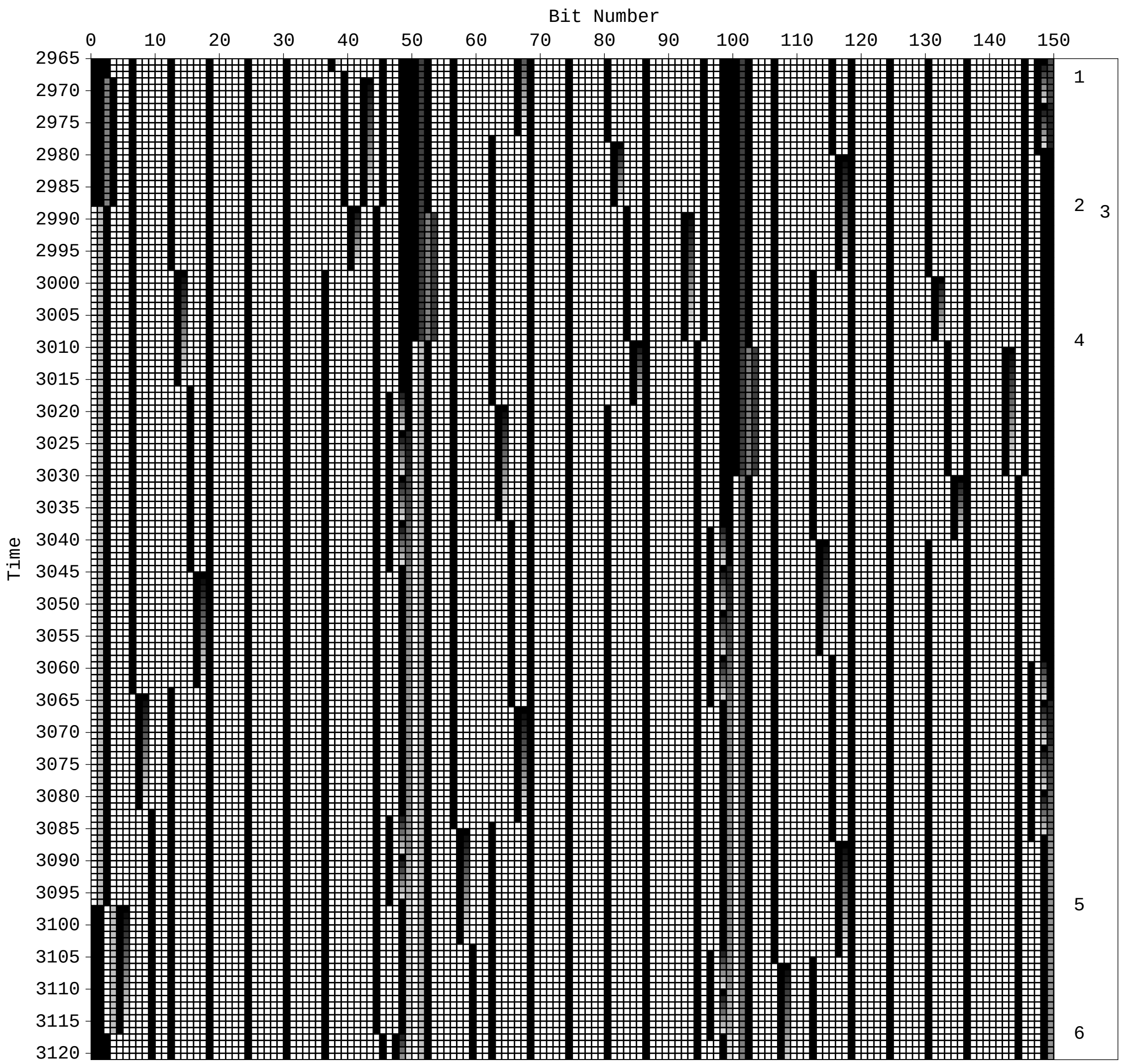


Figure B.2: The detailed evolution trace of the completely connected 2-3-4 size system with a (0-0-40-60) queue distribution from the 2,965th to the 3,120th evolution step. In this sequence, the events leading to a carriage entering an empty queue are illustrated.

Vita

Roger Anderson was born on September, 1976 to parents Robert and Ellen in Huntington, NY. He attended Webb Institute on a full scholarship and, in 1998, graduated with a bachelor's degree in naval architecture and marine engineering. In 1999, he received a master's degree in ocean engineering from the Massachusetts Institute of Technology. His master's thesis, *Experiments and Simulation of Line Heating of Plates*, investigated a model to predict angular deformations in plates from localized heating as an alternative to computationally intensive three-dimensional, non-linear finite element modeling.