The Effects of Boundary Conditions on Cellular Automata

Brian J. LuValle
Algorithmic Nature Group
LABORES for the Natural and Digital Sciences
Paris, France

In this paper, we explore how the behavior of cellular automata (CAs) is affected by different boundary conditions and what types of systems it is affected by. Six experiments are presented. The first determines the most effective boundary condition for optimizing rule-output (in)equality related to Shannon entropy. The second experiment determines the effect of the boundary conditions compared to the size of the initial conditions. The third experiment determines the variation of systems for a specific Li–Packard and Wolfram class. The fourth experiment generalizes the principles of the third experiment into a rule space involving four binary cells rather than three. The fifth experiment determines the change in complexity of systems with different boundary conditions. Finally, the sixth experiment generalizes the concepts of the fifth experiment into the same rule space used in experiment four. Most of the time, boundary conditions do not change the classification, thus do not change the behavior. But some boundary conditions have been found to change the qualitative behavior. However, it is only constant boundary conditions that have been found to change the complexity of a system, and the constant boundary conditions only decrease the complexity. The largest change in the compression ratio was found to be for rule 110 when the boundary conditions change from cyclical to constant. The degree of this change is approximately 25%. Distributed and cyclical boundary conditions can be relied upon to maximize the rule-output equality of systems in general, while null boundary conditions can be relied upon to increase the overall complexity of a system.

Keywords: boundary conditions; Wolfram classes; Li–Packard class; predicting cellular automaton behavior; uncompressibility

1. Introduction

Boundary conditions are a medium that an automaton utilizes as a substitute for a lack of infinite length of the lattice of conditions. Five types of boundary conditions are tested to see which type of boundary condition has the highest impact on rule-output (in)equality.

https://doi.org/10.25088/ComplexSystems.28.1.97

The inputs of a rule represent the values of both a cell and its neighbors required for the rule to transform the active cell so that its value is the same as a corresponding output.

Constants are the values an arbitrary cell possesses. The values of the constants are limited to the scope of the rules; therefore, in an elementary cellular automaton (ECA), the constants are either 1 or 0. If the initial conditions are to be excluded from the automaton, only the scope of the outputs of the rules can be constants.

The proportions of constants in the outputs of the rules will be equal to the proportion of constants in a single evaluation of the rules on initial conditions, with a proportion of constants of 1:1. This fact has many applications, including being the reason that an automaton of the second Li–Packard class [1] has near-perfect rule-output equality.

Rule-output equality is when the density of nonzeros of the rule output matches the proportion of nonzeros in the output rows or after k steps. The rule-output equality of a particular system can be measured by taking the density of nonzeros in the outputs of the rule evolution divided by the density of nonzeros in the rule output of a system such as a cellular automaton (CA).

Although rule-output equality displays high variation, it does have the ability to be predicted with 100% accuracy. Generally for a system to maximize its rule-output equality over time, the proportions of the initial conditions must be 1:1.

The first type of these boundary conditions examined in this paper is constant boundary conditions. Constant boundary conditions utilize a single constant and remain static through the length of the evaluation. The next type is random boundary conditions, which select a constant in the scope of the rules at random. The third type is pseudorandom or distributed boundary conditions—they generate the boundary conditions based on the proportions of the constants in one evaluation of the rules, with the proportions of the constants in the initial conditions being 1:1. Next, cyclical or periodic [2] boundary conditions are used. The final boundary condition tested as a medium for changing the rule-output equality is null boundary conditions or boundary conditions that are outside the scope of the rules, so information becomes lost over time. All the boundary conditions are shown in Figures 1 through 6 for ECAs.

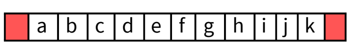


Figure 1. The lattice (values a through k) on which an automaton will be evaluated. The boundary conditions are shown in red.

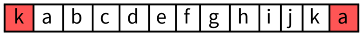


Figure 2. Periodic boundary conditions in which the values of the cells correspond with the name.

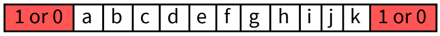


Figure 3. Random boundary conditions are displayed.

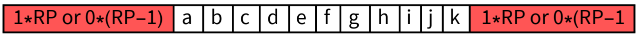


Figure 4. Distributed boundary conditions where the weights on the generation of a particular constant are equal to the weights of randomly selecting that constant from the list of the outputs of the rules.

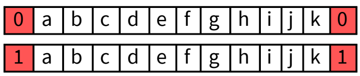


Figure 5. The top graphic depicts constant boundary conditions (red) where the boundary condition has a value of 0 (the constant). Constant boundary conditions could also have a constant value of 1, as in the bottom graphic.

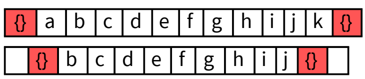
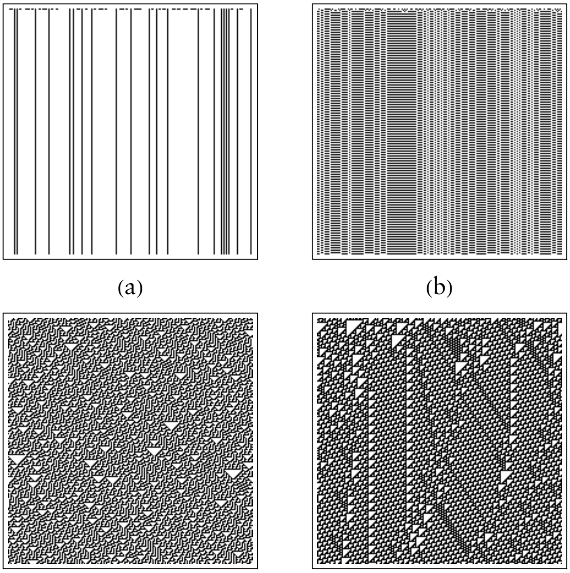


Figure 6. Null boundary conditions. Two evolutions are shown to display the loss of information.

2. Methods

The Wolfram classes [3] are a series of classes used to classify cellular automata (CAs) (Figure 7). The first Wolfram class is described as reaching stability and remaining in that state. An example of this Wolfram class would be ECA rule 4. The next Wolfram class, or Wolfram class 2 systems, are systems that have simple repeated change and no rapid growth and decay. An example rule would be rule 33. Class 3 systems are changing constantly and have no permanent structures, contrary to class 4 systems, which include both rapid change and permanent structures. An example of a class 3 Wolfram system can be seen in rule 30, and an example of a class 4 Wolfram system can be seen in rule 110.

https://doi.org/10.25088/ComplexSystems.28.1.97



(a) (b) (c) (d)

Figure 7. Examples of the Wolfram classes: (a) rule 4, (b) rule 33, (c) rule 30 and (d) rule 110.

The Li–Packard classes [1] (Figure 8) are similar to the Wolfram classes [3], though the Li–Packard classes offer more specification for nonchaotic or noncomplex systems than the Wolfram classes. The first Li–Packard class describes uniform or null systems that have a homogenous configuration of their cells. This can describe the first type of class 1 systems, although only two ECA systems can be classified in this way. The second Li–Packard class describes nonhomogenous but stable systems as well as the second type of class 1, under which many ECA systems are classified. The third Li–Packard class can be thought of as encompassing both class 1 systems such as rule 254 and class 2 systems such as rule 16. This class of Li–Packard systems can be thought of as two-cycle systems. These systems, after two evolutions, become invariant—this includes spatial-shift rules such as 16.

The next Li–Packard classes are known as periodic, in which the configuration becomes invariant after a certain cycle length, dependent on the number of cells. These can be thought of as rules such as rule 3. The fifth Li–Packard class of system is complex, in which the time required to reach the limiting condition is often extremely long (rule 30). The final Li–Packard classes are known as chaotic and are exponentially divergent in cycle length with respect to the number of cells.

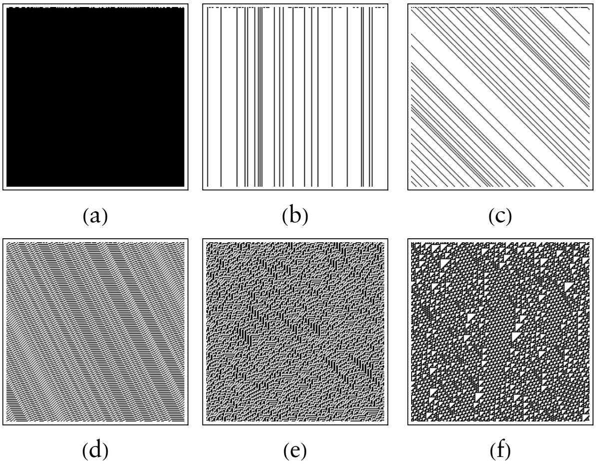


Figure 8. Examples of the Li–Packard classes: (a) rule 255, (b) rule 4, (c) rule 16, (d) rule 3, (e) rule 75 and (f) rule 110.

The entropy of the space-time evolution is calculated using the Shannon entropy equation as performed in [4] and is equivalent to rule-output equality. An estimation of the Kolmogorov complexity of a CA evolution with different boundary conditions is estimated by measuring the length of the lossless compression of that series. To calculate this length, the Wolfram Language function [5] `Compress` is used, which is based on Lempev–Ziv–Welch algorithm. The ratio of the size of the compressed version of the string to the standard size of the string is taken as first used in [6] on ECAs and CAs. This is the ratio of lossless compression and is used to measure the lossless compression of an object [4].

Rule-output equality can be quantitatively described as the proportion of ones compared to the proportion of ones in its global rule description. For example, the rule output of ECA rule 110 is $\{0, 1, 1, 0, 1, 1, 1, 0\}$ and its nonzero proportion is 5/8. The rule-output equality of a system measures how close the proportion of the output of the program would be to the proportion of the rule density. Using this measure, we can represent the general distribution of inputs throughout the system. For similar rule-output equality values for different initial conditions, we can determine that the programmability of that system is very low, as it seems to yield the same distribution of inputs of the rule.

Binary rule-output equality can be predicted using the outputs of the rules as well as their positions and the length of the evaluation. First, to predict rule-output equality, the probabilities of the specific combinations of the inputs of the rules are examined. Next, the probabilities are added and the predicted proportions are determined. The predicted proportions are then applied as initial proportions, and this process is repeated for the length of the evaluation.

However, the rule-output equality of a system can change based upon the initial condition of a system. The rule-output equality of a specific proportion of a specific initial condition is called specific rule-output equality.

Rule-output equality can be generalized to more symbols/states.

3. Results

3.1 The First Experiment: Rule-Output Equality per Boundary Condition

The first experiment measures rule-output equality through the proportions of the constant value 1 to the length of the entire system. This is done for all of the ECA rules. The experiment is constructed by creating a list of each rule and a corresponding automaton function. Each of these rules is run with random initial conditions for 200 steps. This is performed for each possible boundary condition. The results are compared in Figure 9.

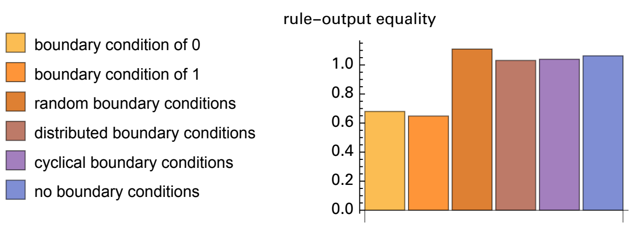
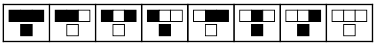


Figure 9. All of the ECAs aggregated together, showing the distribution of rule-output equality with respect to each boundary condition.

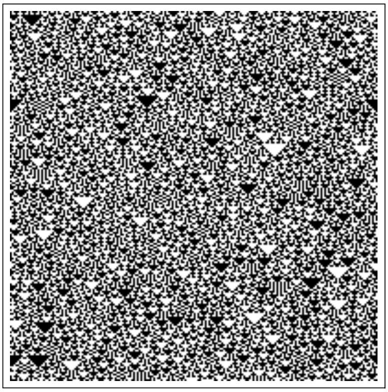
Random boundary conditions maximize the rule-output equality, but it exceeds the rule-output equality by 10%. Because of this, it will be the equivalent of 90% and will not work as well for optimizing the predictability of rules using random dot propagation as, for instance, the distributed boundary conditions. Distributed boundary conditions are more effective at maximizing the rule-output equality than initial conditions of the same size. This is, however, only due to field of influence, as boundary conditions have two directions for fields of influence rather than one.

Distributed boundary conditions have the closest to perfect rule-output equality. Cyclical boundary conditions are very close—to the point of the difference being statistically insignificant given the randomization of the boundary conditions as well as the initial conditions. Because of this, we can hypothesize that the distributed boundary conditions and the cyclical boundary conditions will be equivalent, and both have near-perfect rule-output equality.

In Figure 10, an example of this is demonstrated on rule 150, by default (to refer to random initial conditions). Rule 150 is chaotic. Because of the chaotic nature of this rule, permanent structures will have trouble existing. For this reason, many chaotic systems approach rule-output equality.



(a)



(b)

Figure 10. (a) A rule plot of rule 150 and (b) an array plot of the CA for that rule using random initial conditions and cyclical boundary conditions.

The distribution of the rule-output equality of this system is in Figure 11. Rule 150, from the lack of diversity of rule-output equality, seems to indicate that rules possessing high complexity are more rigid in their rule-output equality and often will nullify the effects of the environment.

Next, an example of the fourth Wolfram class is shown in Figure 12. Using the definition of complexity as defined in this paper, this is less complex than rule 150. Because of that, we can expect the system to possess less rigidity in its rule-output equality.

As predicted, there is less rigidity of the rule-output equality present (Figure 13). It can now be hypothesized that the variation of the rule-output equality of a system will be inversely proportional to the amount of variation in the rule-output equality with different boundary conditions.

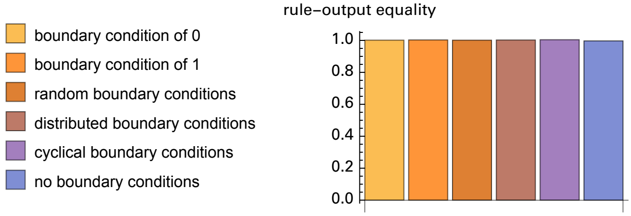
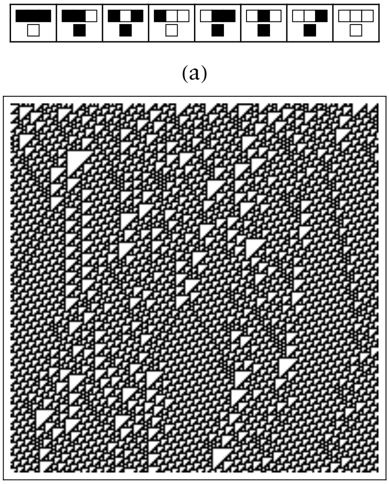


Figure 11. The distribution of the rule-output equality with respect to each boundary condition for ECA rule 150.



(a)

(b)

Figure 12. (a) A rule plot of rule 110 and (b) an array plot of the CA for that rule using random initial conditions and cyclical boundary conditions.

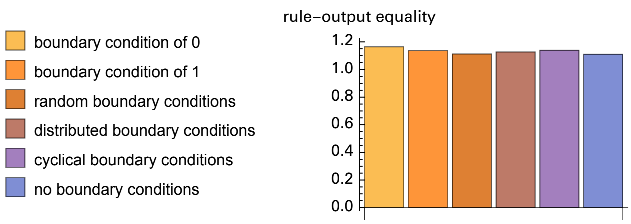
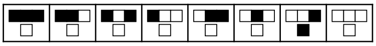
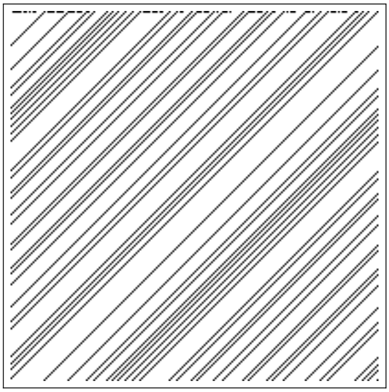


Figure 13. The distribution of the rule-output equality with respect to each boundary condition for ECA rule 110. More variation can be seen in this system than the previous.

To test this, rule 2 will be used (Figure 14). According to this recently formed theory, rule 2 should have more variation of rule-output equality because of its low complexity.



(a)



(b)

Figure 14. (a) A rule plot of rule 2 and (b) an array plot of the CA for that rule using random initial conditions and cyclical boundary conditions (left).

This theory has so far been correct in describing the inverse proportion of the variance of the set of rule-output equality with boundary conditions and the complexity of a system. Figure 15 shows a test of the variance of rule-output equality for rule 4.

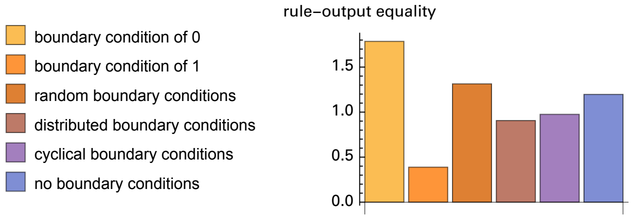


Figure 15. The distribution of the rule-output equality with respect to each boundary condition for a spatial-shift system (ECA rule 4).

Systems of lower complexity often have more rigidity of rule-output equality than more complex systems such as rule 110. This disproves the theory that complexity is inversely related to the variation of the rule-output equality with boundary conditions, and this relationship goes both ways. Figure 16 shows a test of the variance for rule 255.

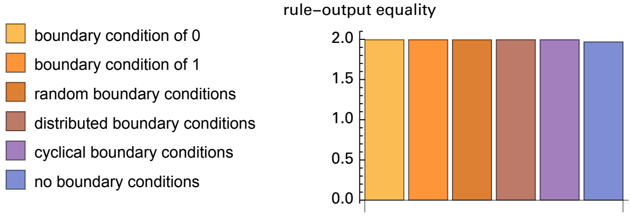


Figure 16. The distribution of the rule-output equality with respect to each boundary condition for ECA rule 255. This system is far less complex than rule 4, but it has less variation.

From this data it would seem to make more sense that the higher variation of the rule-output equality would represent a more spatial-shift-based system. Evidence of this can be seen in the high variation of the spatial-shift system rule 2, and the slightly lower variation in rule 110 with fewer spatial shifts occurring.

3.2 The Second Experiment: Variance of Rule-Output Equality per Size of Input

When using field of influence, the idea becomes apparent that the impact of the boundary conditions is related to the size of the initial conditions. This holds true, as the two are inversely proportional. With larger initial conditions, the boundary conditions affect the system much less. In Figure 17, a graph is plotted of the size of the input and the variance for all boundary conditions. This is done for rule 2 and performed using 200 evaluations.

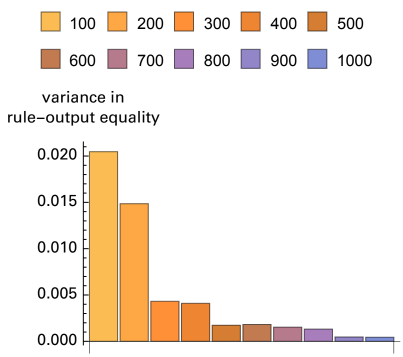


Figure 17. A bar graph depicting the relationship between variance in the rule-output equality and the effect of the boundary conditions, that is, the length of the evaluation.

The question is posed, How impactful are initial conditions compared to boundary conditions? An experiment similar to the previous is composed (Figure 18), though instead of changing the length of the initial conditions, the length of the evaluation is changed. We might expect the relationship between boundary conditions and initial conditions to also be inversely proportional, but from the current graph it seems to be more of a direct relationship. As the length of the evaluation (the color) changes, the statistical variance increases.

The increase in variance is due to the higher number of critical points that will affect a system. Unlike the initial conditions, the boundary conditions will continue to affect the system as the influence of the initial conditions becomes lower, due to the farther distance from the first evaluation.

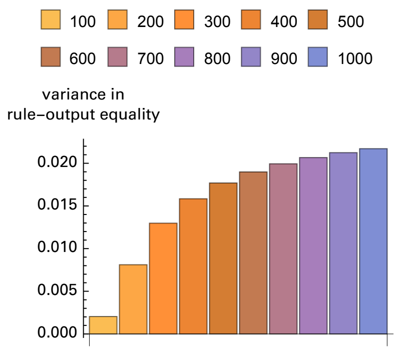


Figure 18. The relationship between the variance in rule-output equality and the number of evolutions a system undergoes.

3.3 The Third Experiment: Relation of the Li–Packard and Wolfram Classes to Rule-Output Equality

As found from the disproof of the theory presented in the first experiment, there is no dual correlation between complexity and variation of rule-output equality. The third experiment examines the potential of a one-way relationship between the two.

In the third experiment, Li–Packard classes [1] and specific Wolfram classes [3] are tested for the effects of boundary conditions. For the first Li–Packard class as well as for a representation of the first Wolfram class, ECA rule 255 is used. The next Li–Packard class is shown by rule 4 (which is also in Wolfram class 1). To demonstrate the third Li–Packard class and also to provide an example of the second Wolfram class, rule 2 is used. Rule 3 is used to provide an example of the fourth Li–Packard class and another example of the second Wolfram class. Representing both the fifth Li–Packard class and the third Wolfram class is rule 30, while rule 110 represents the sixth Li–Packard class and also the fourth Wolfram class.

Each of the Li–Packard class examples uses 200 cells for the initial conditions with random constants and is evaluated for a length of

200 steps. The rule-output equality is measured using the methods previously defined. A table of this is shown in Figure 19.

Rule	1	0	Distributed	Null	Cyclical	Random
255	1	1	1	1	1	1
4	1.0088	1.0244	1.0224	1.0192	9.08896	1.0164
2	1.82907	0.491644	0.77312	0.9852	7.79144	1.14696
3	1.80747	1.81467	1.81297	1.7537	7.24617	1.81137
30	0.997996	1.0029	1.00033	0.999961	2.00249	1.00058
110	0.90875	0.91338	0.911688	0.903282	1.44742	0.909183

Figure 19. The rule-output equality for each boundary condition and a system of each Li–Packard class.

If we omit rules 30 and 110, the Wolfram rules, then we will find that a graph of this shows definite increasing complexity (Figure 20). The boundary conditions of 1 and the boundary conditions of 0 are found to be complementary, with random evenly distributed boundary conditions being between both on the graph.

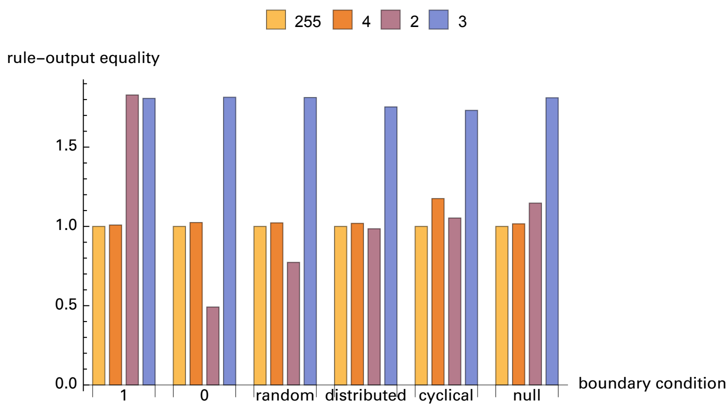


Figure 20. The rule-output equality for the rule and conditions it corresponds with. In this graph, the x axis represents the boundary conditions in order of increasing estimated Kolmogorov complexity (lossless uncompressibility).

When examining the graph in Figure 20, it becomes clear that there are only a few select types of systems that have boundary-condition-dependent rule-output equality (rule 2). All of the systems with the exception of the reasonably close null conditions display low variance in position, except for rule 2.

The reason for the high variance in rule-output equality is the matter of spatial shift. These spatial-shift systems will move off screen, assuming a boundary value does not interrupt. A spatial-shift system is shown in Figure 21.

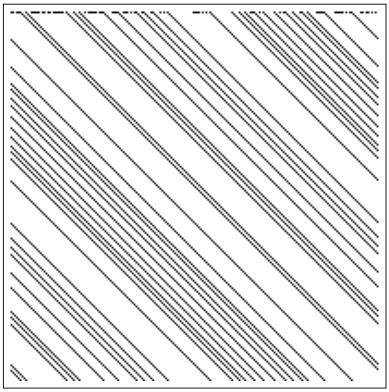


Figure 21. An example of a spatial-shift system.

However, in the graph of the rule-output equality of the Li–Packard classes in Figure 22, it becomes apparent that the spatial-shift system is the only Li–Packard class system that experiences high variation in rule-output equality.

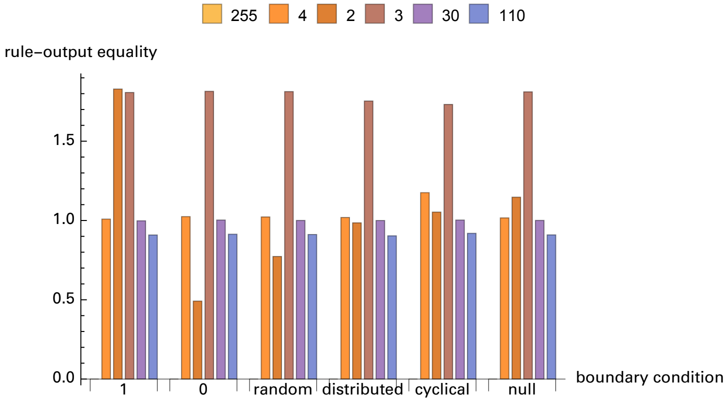


Figure 22. The distribution of rule-output equality for each boundary condition and the first four Li–Packard class examples.

It has been seen that the system possessing the greatest variation in rule-output equality is a spatial-shift system. Most other systems have a constant rule-output equality, which will deviate slightly given changes in the estimations of Kolmogorov complexity by uncompressibility of the series of boundary conditions.

As has been seen, it is possible to increase the rule-output equality of an object using its boundary conditions. As the distributed boundary conditions are the closest to perfect rule-output equality (of 1), the distributed boundary conditions have been seen to maximize the rule-output equality of a system across all levels.

3.4 The Fourth Experiment: Relation of the Li–Packard and Wolfram Classes to Rule-Output Equality for Higher Rule Spaces

The fourth experiment generalizes the principal measures of rule-output equality to a CA or CAs with larger rule spaces. In this experiment, the rules with a larger rule space than four use random number generation to determine the value of the output constants of the rules.

Similarly to the previous experiments, the system of each Li–Packard classification is used (Figure 23). The rules are generated using an output constant-based structure. As a representation of the first Li–Packard class, the 16th rule is used. To represent the next Li–Packard class is rule 13108. Representing the spatial-shift systems is 843, followed by the more complex 3213. To represent the chaotic classification of the Li–Packard systems is rule 32421. Finally, rule 5832 stands as a representation of the chaotic rule classes.

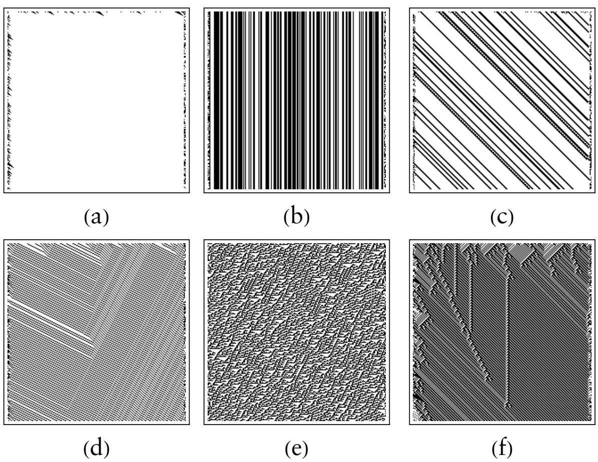


Figure 23. An example of each system using random initial and boundary conditions: rules (a) 16, (b) 13108, (c) 843, (d) 3213, (e) 32421 and (f) 5832, in order of increasing complexity according to the Li–Packard classifications.

Rule 32421 appears to be a spatial-shift system, but it falls short on the matter of its variation of shifting. Contrary to conventional spatial-shift systems, rule 32421 sometimes shifts by 0 to 2 cells each evolution. These variable spatial-shift systems behave much more like the chaotic systems than spatial-shift systems.

Due to the configuration of the preceding systems, the impact of the boundary conditions is visible, and because of this, no system can truly be a completely stable system using random boundary conditions. Therefore, even for the system of the first Li–Packard class, there will be variation in the rule-output equality.

Figure 24 shows a table of the rule-output equality for each system and boundary condition.

Rule	1	0	Distributed	Null	Cyclical	Random
16	2.081	0.008600	0.06320	0.02880	0.01260	0.06367
13108	0.9200	0.8200	1.055	0.9408	0.9600	0.9273
843	0.7405	0.2401	0.5160	0.4077	0.2828	0.7536
3213	1.297	0.4165	1.208	1.152	1.164	1.189
32421	0.8558	0.8289	0.8400	0.8385	0.8426	0.8432
5832	1.485	0.5521	1.202	1.246	1.167	1.178

Figure 24. Each CA rule and boundary condition and the corresponding rule-output equality.

In this experiment, the rule-output equality is distributed drastically differently than the previous three cell input systems (Figure 25). This is most likely due to the chaotic effect of the boundary conditions and their greater extent over the system.

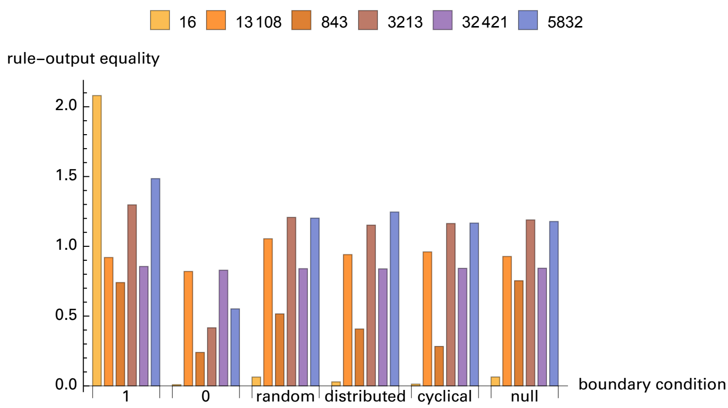
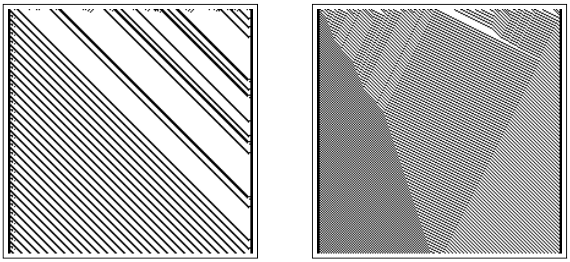


Figure 25. The distribution of the rule-output equality for each system using Li–Packard example systems of the fourth rule space.

Unlike the previous ECAs, the variation of these CAs is extraordinarily high. This is because of the extent of the boundary conditions and their influence. Unlike the ECAs with the single influence of the boundary conditions, systems using the fourth rule space develop two layers of cells that are impacted by the boundary conditions, rather than one.

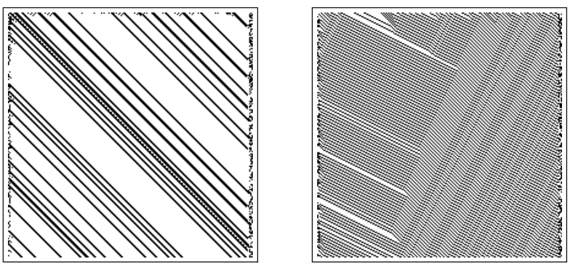
Although many more systems possess high variation, the spatial-shift system possesses the most variation once again (Figure 26). The relative levels of rule-output equality are similar, with the exception of rule 16 with singular boundary conditions, given that it exceeds the expected proportions, because rather than the completely homogenous configuration, a boundary of 1-valued cells is found.



(a) (b)

Figure 26. (a) The spatial-shift and (b) the dual-spatial-shift systems are shown using constant boundary conditions.

Similarly to how chaotic systems such as rule 32421 form simple structures when exposed to constant boundary conditions, the class 2 systems that have higher complexity in their spatial-shift patterns also form simple permanent structures (Figure 27).



(a) (b)

Figure 27. (a) The spatial-shift and (b) the dual-spatial-shift systems are again shown, this time using random boundary conditions.

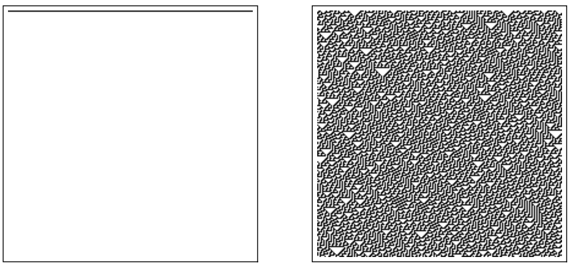
The complex or sixth example Li–Packard class rule seems to not be affected by the change in boundary conditions, with the exception of the constant of 1 being the boundary condition.

3.5 The Fifth Experiment: Examination of the Change in Entropy per Boundary Condition

The fifth experiment determines the change in complexity for the change in boundary conditions. The measure of complexity is determined by the use of Shannon or information entropy. The boundary conditions are arranged in order of increasing estimated uncompressibility for their series; that is, constant, random, distributed, cyclical and null.

When the entropy of a system is increased, that system comes closer to being a Wolfram class 3 system, and therefore the predictability of that system using random dot propagation is increased.

Given that initial conditions can change a system from homogenous to complex, it is assumed that boundary conditions can do the same thing (Figure 28). However, as experiment 2 has proven, the length of the evaluation is directly proportional to the rule-output equality, while the size of the boundary conditions is inversely proportional.



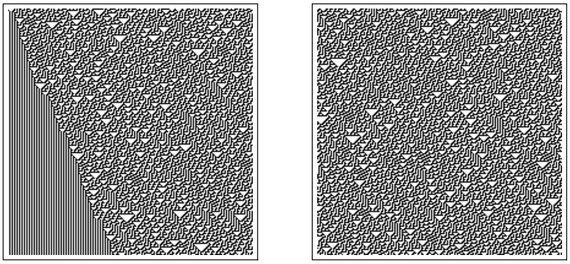
(a) (b)

Figure 28. An array plot of rule 30 showing (a) constant initial conditions as opposed to (b) random initial conditions.

Similarly to the previous experiment, a system of each Li–Packard class is used. Each of these systems is tested for increase in complexity as the uncompressibility of the boundary conditions is increased.

For systems that possess class 1 Li–Packard behavior, it is seen that the boundary conditions or lack thereof has no effect on the overall structures present in the system. This would seem to indicate that class 1 Li–Packard systems would all have to be induced by the outputs of the rules rather than the conditions, and therefore their behavior is unchangeable.

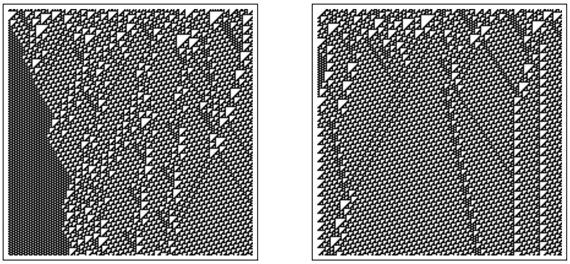
Constant boundary conditions seem to create permanent structures (Figure 29). Rather than disappearing, these structures seem to overtake the existing conditions in place. Therefore these systems will eventually become class 2 systems and become stable after the permanent structures have overtaken the chaotic ones. Although the structures presented by the system would appear to be class 1, at the very end a slight oscillation appears, making it so that the system is not Wolfram class 1 but Wolfram class 2. Therefore, constant boundary conditions cause a decrease in the overall complexity of a system.



(a) (b)

Figure 29. Two array plots of ECA rule 30: (a) constant boundary conditions are used, while (b) uses random boundary conditions.

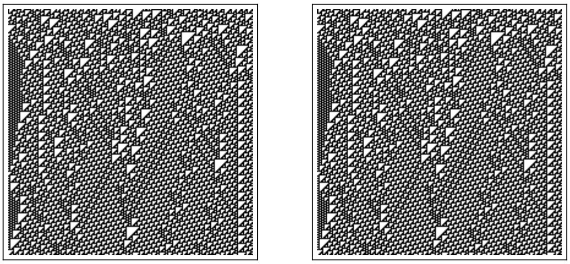
However, constant boundary conditions do not always create expanding permanent structures. In Figure 30, rule 110 is shown with permanent boundary conditions; the size of the experiment structure on the end, however, remains constant. With rule 110, the impact of constant boundary conditions often varies significantly, as in Figure 30.



(a) (b)

Figure 30. ECA rule 110 with (a) constant boundary conditions as opposed to (b) random boundary conditions.

This can be one of two things—either the result of random conditions in the initial conditions or the boundary conditions. To determine this, we can examine the presence of this structure under constant initial conditions and random boundary conditions or vice versa to determine the change. An example of this is shown in Figure 31 for two systems of the same initial conditions but separate boundary conditions. It can be noted that the nonpermanent structures on the side are similar if not the same. Therefore, it is the initial conditions that the boundary conditions depend on for the creation of permanent structures in this rule.



(a) (b)

Figure 31. Two copies of rule 110 with the same initial conditions but random separate boundary conditions. This figure illustrates that the structures of rule 110 depend more upon the initial conditions than the boundary conditions.

Figure 32 shows the Shannon entropy of each system for each boundary condition and rule—corresponding to a Li–Packard class. In the graph shown, the noncomplex systems seem to be impacted by the change in uncompressibility of the boundary conditions series, while systems possessing entropies of 0.6 or greater are not affected by such changes.

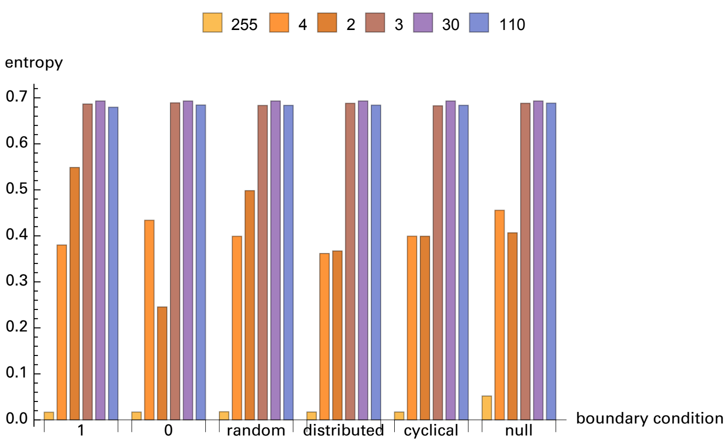


Figure 32. Shannon entropy behavior. The boundary conditions change in order of increasing uncompressibility across the *x* axis.

The variation of rule-output equality for a system with changing boundary conditions is similar to the variation of the entropy of that same system with boundary conditions. However, the constants involved in both systems differ quite drastically.

The connection of entropy to the rule-output equality of a system seems to only be in the variation of the system rather than the constants of the system. However, systems do seem to have similar changes for both rule-output equality and entropy when the boundary conditions are changed (Figure 33).

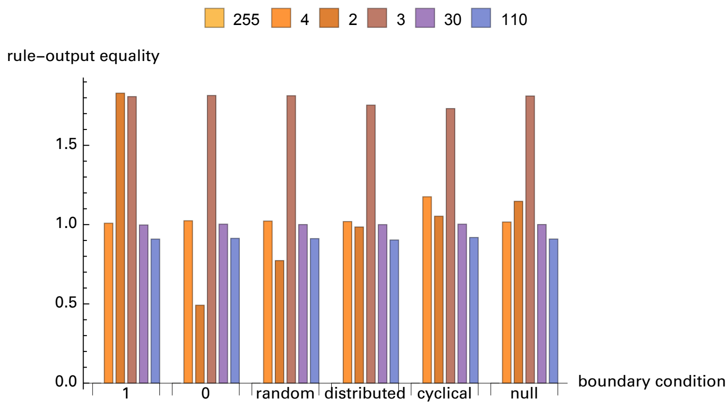


Figure 33. The rule-output equality and its variation for each boundary condition.

3.6 The Sixth Experiment: Examination of the Change in Entropy of Systems per Boundary Conditions Generalized for Higher Rule Spaces

Experiment six generalizes the principles of the previous experiment into a general CA. The rule spaces used for this CA will be using four binary cells rather than the three used in ECAs. Similarly to the fourth experiment, the sixth experiment utilizes the same systems of the fourth rule space to represent each of the Li–Packard classes.

To create the rules for these CAs, a general constant-based rule function is used. After the tables are created, they are flattened into a list containing 16 elements or the number of combinations for four binary cells. The number represents at which position in the list the rule was created.

Similarly to previous systems using the fourth rule space, the boundary conditions also increase the variation due to the larger size. Figure 34 shows a table of the entropy of each of the rules for each boundary condition.

From this it is shown that systems of larger rule spaces that require larger boundary conditions will have more variable complexity than systems that have smaller sets of boundary conditions.

Previously for the ECAs, the variation of the rule-output equality was shown to be similar to the variation of the entropy of that system. That principle still holds true for the CAs, though there is more variation of the rule-output equality (Figure 35).

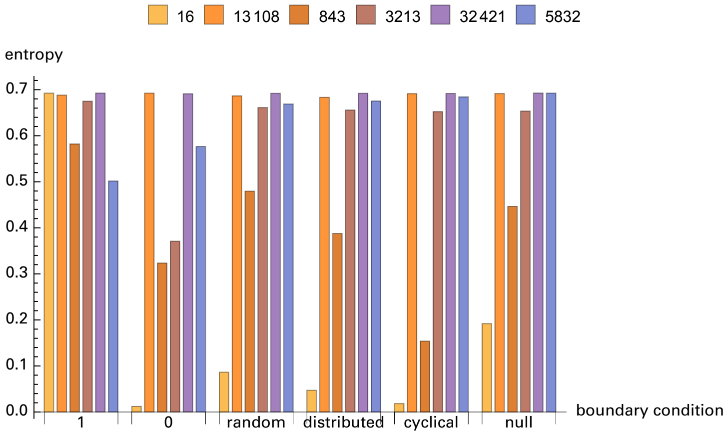


Figure 34. The rule-output equality and its variation for each boundary condition.

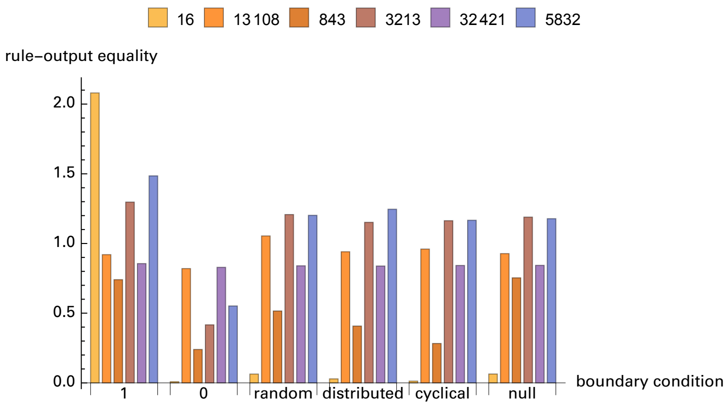


Figure 35. The rule-output equality and its variation for each boundary condition.

The general predictability of ECAs seems to be very similar to the predictability of CAs. This can be deduced by the similarity between either the entropy or the rule-output equality of a system. The rule-output equality increases the chance of generating a value of a cell randomly and having it match the results of an actual CA, while increasing the entropy decreases structuring and therefore makes it more likely that a particular cell is to be found in a certain region (Figure 36).

In this experiment it has been seen that the two factors that impact predictability by random dot propagation, rule-output equality and the entropy of a system are both changed in a way that allows them to reach the optimized levels for predictability.

The only difference between Figure 36 and Figure 37 of the entropy per Li–Packard class is the entropy of the first Li–Packard class and the second Li-Packard class. These systems would possess class 1 entropy similar to Wolfram class 1 systems if the boundary conditions did not create structures near the boundaries due to the increased rule space.

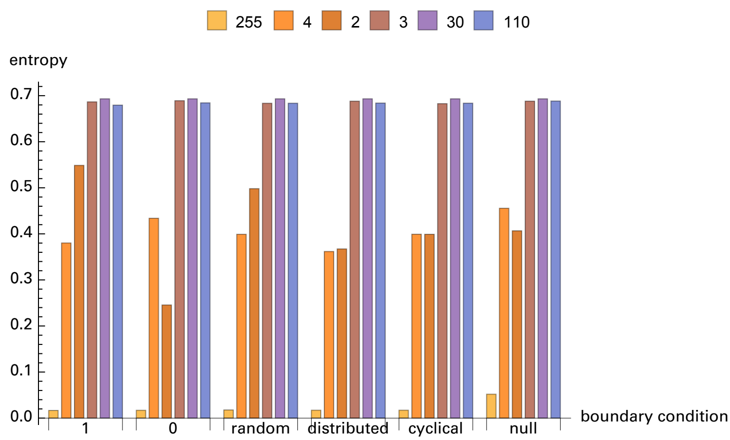


Figure 36. The distributions of the ECAs with respect to the boundary conditions and example Li–Packard systems.

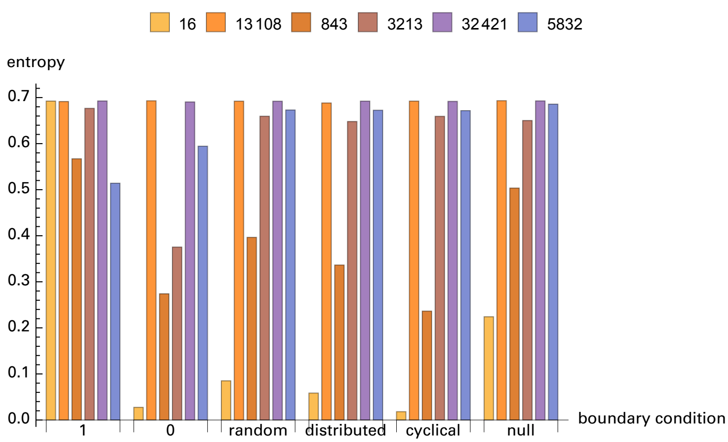


Figure 37. The distributions of the CAs with respect to the boundary conditions and example Li–Packard systems.

An example of boundary conditions impacting the complexity of a system can be seen in Figure 38 for rule 110. For all systems, the same initial conditions are utilized. The area can be divided into zones of function based on the common perseverance of certain structures in that area as well as the structuring in the initial conditions. The first zone in the boundary conditions is the beginning, where the constants of 0 dominate the constant of 1. Next a transition to zone 2 occurs, where the majority of the constants are 1 rather than 0. Then there is zone 3, which persists until the end of the initial conditions, where very large structures are present.

Figure 38. The initial conditions of the CA.

Figure 39 shows an array plot utilizing the constant of 1 as the boundary conditions; a part of the first zone is populated with a simple permanent structure. Zone 3 behaves in a rather chaotic and erratic way, which also exhibits minor spatial shifting.

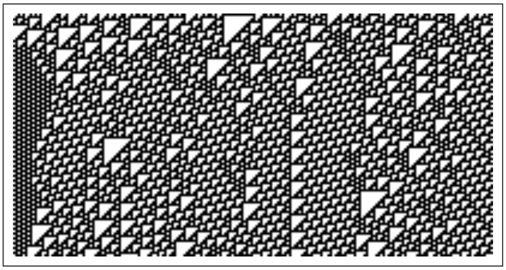


Figure 39. Rule 110 with constant boundary conditions where the constant is 1.

Figure 40 shows an array plot where the first zone lacks the permanent structure seen previously. The second zone is also more erratic than previously; however, certain elements are preserved. The third zone seems quite stable and possibly behaving like a Wolfram class 2 system in that region.

The second system of boundary conditions and the first system of boundary conditions seem to be complementary even when impacting the automaton. In the first set of boundary conditions, zone 1 is non-complex, while zone 3 is complex. This is reversed for the second set of boundary conditions, in which zone 1 is complex, while zone 3 is not.

Random boundary conditions display zones 1 and 3 becoming quite complex and even impacting zone 2 a bit. Random boundary conditions, as we see in Figure 41, act as a middle ground between the two, and at the same time also increase the complexity.

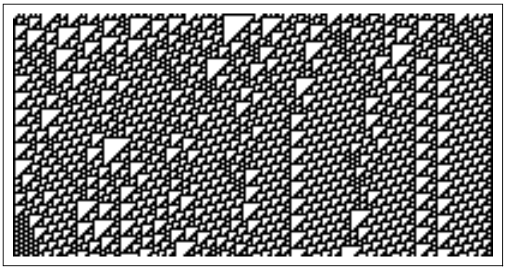


Figure 40. Rule 110 with constant boundary conditions where the constant is 0.

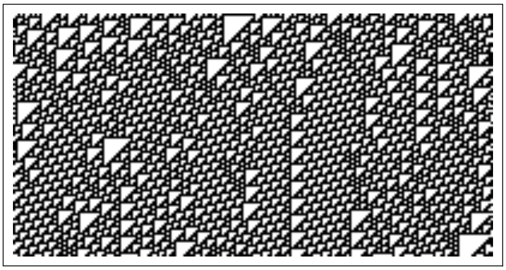


Figure 41. Rule 110 with random boundary conditions.

Distributed boundary conditions display the structuring from both the boundary condition of 1 and the boundary condition of 0 (Figure 42). Then this structuring disappears and evolves into the boundaries seen in the random boundary conditions.

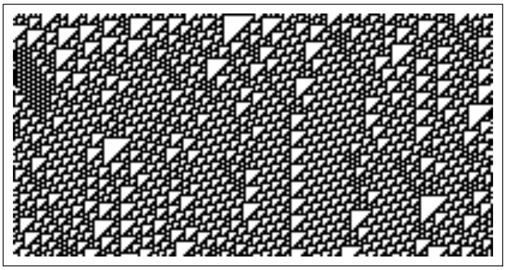


Figure 42. Rule 110 with distributed boundary conditions.

Cyclical boundary conditions display a similar lack of complexity as the start of the distributed boundary conditions; however, this continues thoughout the length of the evaluation (Figure 43).

Aside from the obvious lack of information, the null boundary conditions seem to have high complexity on both ends, similar to the random boundary conditions (Figure 44). However, this change does not last for the length of the system and eventually is lost.

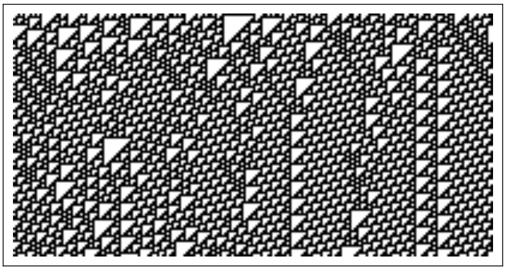


Figure 43. Rule 110 with cyclical boundary conditions.

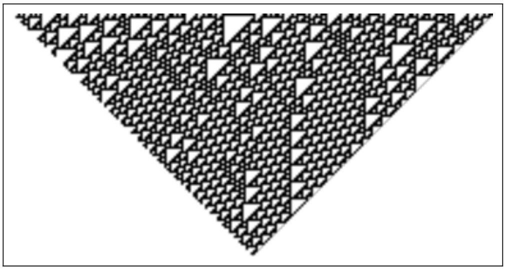


Figure 44. Rule 110 with null boundary conditions.

4. Conclusion

It has been seen from this paper that distributed or cyclical boundary conditions are most effective at maximizing the rule-output equality for cellular automata (CAs). This only affects a very specific type of system (although this type of system is quite common). The systems that possess variations in their rule-output equality dependent on boundary conditions have been found to be class 3 when using the Li–Packard definition, and specifically systems that use spatial shifts. This would actually make quite logical sense, as the spatial-shift systems will have their shift based on the limited-space automaton in which information is lost.

Two-cycle Li–Packard systems are affected the most by the boundary condition, though other rules are as well. Systems such as rule 110 are unaffected by these boundary conditions, even though rule 110 spatially shifts the pattern as well. All other systems that are not two-cycle Li–Packard systems seem to have relatively the same rule-output equality. For the second rule used in the specific rule-output equalities, there is one exception, but this can be dismissed due to the fact that it is a fixed-point system according to the Li–Packard classification, in which it is stable after applying an updating rule once. With decreasing size, the proportions will also decrease and certain regions will be cropped.

The data from the null boundary conditions must be more carefully considered given the nonuniform compression ratio with the first Li–Packard class. Because of the homogenous configuration, the compression of such an object is exactly the same size. However, the initial size of the system is much smaller, and when the ratio between the two sizes is taken, the effect of the much smaller piece of data divided by something shows and therefore it is larger.

From the fourth experiment it has been determined that an increase in the size of a rule space and therefore a boundary condition space will create higher amounts of variation in the rule-output equality. The constant boundary condition of 1 seems to always increase the rule-output equality more than the constant boundary of 0 decreases it.

From the fifth experiment it has been found that the boundary conditions of a cellular automaton (CA) can increase or decrease the complexity of system similarly to the initial conditions. The null boundary conditions have been found to increase the complexity of a system the most, while the constant boundary conditions (1 and 0) have been found to decrease the complexity of a system the most. Random boundary conditions seem to increase the complexity, yet not as much as null boundary conditions. Finally, distributed and cyclical boundary conditions display rather uniform properties in which change in complexity is between constant and null. This is displayed in Figure 30.

The sixth experiment shows that systems of the same Li–Packard classes will have similar behavior under different boundary conditions. This allows us to generalize which systems would be the most predictable. The systems that currently display this pattern are the class 4 and 5 Li–Packard systems.

Boundary conditions have been found to change the Wolfram classification of behavior. However, it is only constant boundary conditions that have been found to change the complexity of a system, and the constant boundary conditions only decrease the complexity. One such example would be rule 30, which can transform from class 3

Wolfram behavior conditions to class 4. As previously discussed, this is not an increase but a decrease in complexity.

Most of the time, boundary conditions do not change the classification, thus do not change the behavior. The largest change in the compression ratio is found for rule 110 when the boundary conditions change from cyclical to constant boundary conditions, specifically the boundary condition of 0 or vice versa. The degree of this change is approximately 25%.

When the success of changing the boundary conditions on each classification of system is analyzed, it is found that this is not an effective method of increasing rule-output equality and it only works for a select type of system. When analyzing this form, the perspective of the general classification system is quite insignificant.

In conclusion, the general predictability of CAs is increased by using distributed or cyclical boundary conditions, that is, the most stable boundary condition. This can address the variability of the predictability of class 2 systems seen in prediction of CA random or partial random prediction. This will not be able to address such prediction variability as is seen in the machine learning-based methods seen in the works of Toole and Page in prediction of CAs [7]. The variance in their predictive methods is due to the variance in the distance of the spatial-shift system and the fact that there will be no set distance, depending on size-based values.

Distributed and cyclical boundary conditions can be relied upon to maximize the rule-output equality of systems in general, while null boundary conditions can be relied upon to increase the overall complexity of a system.

Acknowledgments

The author thanks Lila Onigman for assistance in the editing of the final draft of the paper.

References

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

[2] J. R. Weimar. "2.3 Boundary Conditions." (Feb 27, 2018) www.jcasim.de/main/node6.html.

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

[4] H. Zenil and E. Villarreal-Zapata, "Asymptotic Behavior and Ratios of Complexity in Cellular Automata Rule Spaces," *International Journal of Bifurcation and Chaos*, **23**(9), 2013 1350159. doi:10.1142/S0218127413501599.

[5] Wolfram Mathematica, Release Version 11.3, Champaign: Wolfram Research, Inc., 2018.

[6] H. Zenil, "Compression-Based Investigation of the Dynamical Properties of Cellular Automata and Other Systems," *Complex Systems*, 19(1), 2010 pp. 1–28. complex-systems.com/pdf/19-1-1.pdf.

[7] J. Toole and S. E. Page, "Predicting Cellular Automata," working paper. sites.lsa.umich.edu/scottepage/wp-content/uploads/sites/344/2015/11/CAWorkingPaper_Edit.pdf.