

A Robust Game of Life

Tom Portegys and Janet Wiles

Abstract. Biological life is characterized by a robustness to noise often lacking in digital models. Although noise in the form of mutations has an important evolutionary purpose, it is also a threat that living systems are particularly adept at defending against. Artificial Life systems are frequently state-machines of some sort, requiring robustness mechanisms aimed at preserving and correcting state information. We explore a class of mechanisms based on robustness in transition dynamics rather than robustness in the emergent patterns of complex systems. We propose a Game of Life enhancement in which disrupting noise in the form of random state changes is corrected using a “light cone” of spatial and temporal context information. This enhancement does not change the original rules: a noiseless enhanced run is identical to an unenhanced run. We believe this general scheme could be applicable to other parallel computing systems in which contextual information determines state transitions.

Introduction – biological robustness and the challenge for digital models

A characteristic feature of biological life is its robustness to many types of perturbations. The genetic code precisely specifies a mapping between DNA and proteins, yet inbuilt redundancy and error correction mechanisms make genomes stable over millions of years. In contrast to biological systems, digital systems have some areas of robustness, but they are frequently brittle with respect to many types of damage, including random noise.

The process of creating a model of a biological system necessarily abstracts and simplifies components. In the past, models have frequently omitted any stochastic components in the search for other insights from natural computation. Many properties of biological systems are emergent, and computational models are a powerful method for studying how complexity emerges from simple processes. The Game of Life is one such model for studying the emergence of dynamic propagating patterns from simple local rules (Gardner 1970). Models of the emergence of different forms of complex structures are now an established area of study of computational modelling and insights are being integrated into systems biology.

A major challenge for computational modellers in general, and systems biologists in particular, is how to model robustness in biological systems. Precise definitions of robustness differ depending on the source of variation. The first step in modeling the robustness of living systems lies in identifying which features a system are most significant for robust dynamics and what kinds of abstractions provide informative analogues. Models such as the Game of Life provided powerful analogies for studying emergent dynamics, but the patterns of gliders, glider guns and other structures are very brittle, in the sense that any non-zero probability of noise destroys the intricate dynamics. Similarly, in a different domain, genetic regulatory networks have been modeled by Random Boolean Networks (RBNs), which show attractors that provide interesting analogies with cell types (Kauffman, 1995). Some attention has been given to the nature of the updating schemes in model systems and synchronous updating has been shown to play a role in maintaining complex structure in many systems. A critical phase transition occurs in the Game of Life as the fraction of cells that are updated changes from low to high synchrony (Blok and Bergerson 1999) and has been shown to affect the number and length of limit cycles in genetic regulatory networks (Harvey and Bossomaier 1997; Di Paolo 2001; Hallinan and Wiles 2004). Random noise has some similarities to stochastic updating in destroying structures that depend on synchronous transitions.

A variety of approaches have been taken to studying robustness in computational systems. Reliable computing with unreliable components is a subject that has drawn interest since Von Neumann's time (von Neumann 1966). Probabilistic cellular automata (PCA) have been used to study both innately probabilistic phenomena, such as the behaviour of human populations. For example, models have been developed of the disappearance of the Native American tribe called the Anasazi in the American South West (Dean, Gumerman et al. 1998). Maintaining information with unreliable components has been shown to be possible in automata (Gacs 1986). Other models have been developed to investigate the properties of error-correcting (non-ergodic) systems. For example, Toom (Toom 1974) proved a method that stabilizes a two dimensional automaton of cells having bi-polar spin values that has found application in the development of ferromagnetic devices such as computer disk drives (Grinstein, Jayaprakash et al. 1985). The study we describe is in this rich tradition of using digital systems to study fundamental issues. We

are specifically interested in preserving the original rules of the Game of Life, and exploring issues that are relevant to biological complexity.

Some aspects of biological systems have the property of acting as digital systems. The 4-letter alphabet in DNA is one such digital code, and yet biological systems have robustness to many types of damage. In the next section we review one such mechanism for robustness in the face of stochastic noise, that of the mapping from DNA to protein known as the Genetic code.

Life's robust mechanisms

The genetic code has fascinated and amazed both biologists and computational modellers since the detective work that revealed its subtleties half a century ago. The genetic inheritance in DNA is in the form of a double stranded string of nucleotides. Each nucleotide triplet codes for a specific amino acid and is called a codon. There are 64 codes (4 nucleotides in three positions in the codon) and 20 amino acids, which allows for considerable redundancy in the mapping. Each amino acid is recruited by a tRNA, which functions like a tugboat that guides the amino acid and attaches it in the order specified by the genetic sequence. The molecular biology of tRNA could in theory attach to any amino acid, functioning like a symbolic label. However, in practise, one genetic code is shared by almost all living organisms with a few minor variations known for some bacteria and yeast (Alberts, Johnson et al. 2002).

The fact that variations occur in the code is taken as evidence that the code is not in fact arbitrary, but has probably evolved. Initial models of the mapping were hypothesized based on error correction between codons. From an information theoretic perspective, an elegant correction mechanism would make sense. Biological organisms are highly evolved, and damage to the pathway from gene to protein can occur at many stages, including mutations to the genes caused by nucleotide substitution when the DNA is copied during cell division, errors in the transcription, editing or other damage to RNA, or mistranslation from codons to amino acids. Error correction for these misadventures is a natural hypothesis to make, but it proved inadequate to explain the structure of the code.

It turns out that the properties of the code point to robustness of a highly functional form. See (Freeland and Hurst 2004) for a lay summary of the properties of the genetic code as summarised in this section, and (Freeland and Hurst 1998) for a detailed description of the methods they used. In particular, the mapping seems to be optimised to reduce the functional impact of substitution errors in the amino acids. The functional form of proteins is governed by their 3-dimensional structure. Some amino acids are hydrophobic, and face towards the centre of a protein, while others are hydrophilic and face the watery environment of the cell. The genetic code has the property that substitutions in the first two positions tend to code for amino acids with similar hydrophobic properties, and the third position (which is known to “wobble” due to mistranslation errors) frequently codes for identical amino acids.

The biological summary is a simplification of the process, and many other mechanisms also contribute to reducing the effects of errors from a variety of sources. At the level of the DNA sequence itself, multiple types of damage and repair mechanisms are known, including spontaneous hydrolysis estimated to occur at the rate of 10^4 events per cell per day, ultraviolet (UV) damage that can affect transcription and replication, and ionising radiation and oxidative damage from normal cell metabolism causing DNA double stranded breaks (DSBs) (Rouse and Jackson 2002). The past decade has produced major insights into the cell's repair mechanisms although it is still unclear how primary DNA damage is detected (Rouse and Jackson 2002).

For the purposes of this study, the essential insight from studying the robustness of the genetic code is that information measures are insufficient for understanding its structure. The best explanation of the mapping to date is that it is governed by the effect of errors on the *functional* form of proteins. Proteins can be viewed as emergent structures created by the dynamic processes of tRNA assembly. In this sense, we can view the tRNA as local mechanisms that assemble simple amino acids to create the complex emergent structures of proteins (emergent compared to the functionality of individual amino acids). The system's properties can be seen as emergent in the sense that the genetic code does not "know" what structures its amino acids create. The robust design of the code is not based on stored templates for the allowable proteins and checking the factory line after production (although cells may also use this form of correction). The robustness is emergent from the inherent properties of the assembly process, just as the proteins themselves are emergent.

Dynamic vs static forms of error correction

In this study, noise is implemented as randomly changing the value of every cell on a grid with a small probability, p . To an extent the rules of Life provide for a certain level of self-correction, for example, a cell becoming alive due to noise in a neighbourhood of dead cells will immediately die. However, in general the game is quite fragile to stochastic noise.

We considered a variety of ways in which robustness could be introduced to the Game of Life. As a computational mechanism, the normal Game of Life is very precise, but not robust. One approach is for each cell to take the average value of its neighbours. Such a model may be robust, and has analogies to spin glass systems, but it loses the preciseness that enables the computational dynamics of the classic model.

We wanted an approach that preserves the original rules so that in the absence of noise, the system behaves exactly the same as the classic model. We developed a discrete approach to robustness that uses information from the history of a system to constrain the allowable dynamics. This approach is consistent with the idea of adding robust mechanisms to the generators of a pattern, rather than to the emergent patterns themselves.

Because the updating rules in the Game of Life are localised to a small neighbourhood, there are a restricted set of allowable transitions in any region of the grid. Consider a 3x3 grid in which the value of the central cell is unknown. If the states of the 8 neighbours are known for the previous time step, then in many cases the state of the central cell will be completely determined. In a 3x3 grid, there are 2^9 possible states. After the initial burn in, many of these states do not occur. If the number of active neighbours was 0 or more than 3 at the previous time step, then the central cell must be off. If the number of active neighbours was exactly 3, then the central cell must be on. It is only the cases where the number of active neighbours was 1 or 2 that the value of the central cell is uncertain. Such conditions constitute ${}^8C_1 + {}^8C_2 = 32$ cases, or 6.25% of the 512 possible states.

Methodology

We enhanced the Game of Life by adding a "light cone" (to borrow a term from astrophysics) of spatial and temporal context information to correct erroneous values. This enhancement does not change the original rules: a noiseless enhanced run is identical to an unenhanced run.

The error-correction scheme is based on propagating state information across the grid. Chemical signals diffusing in a medium would be a plausible natural analogy. Cells in the grid

transmit their states to each other via signals sent at the "speed of light" (one cell per step), limited by a maximum range. Noise probabilistically causes signals to be misread by destination cells in proportion to the distance traveled. A cell uses this cone of information to construct a set of state histories for its locale. The most recent history, that of the previous step, is the conventional 3x3 Life neighbourhood. The cone can then be used for error-correction. A neighbour's state is compared against its neighbours' states during the previous step. If they agree according to Game of Life rules, then the neighbour is considered to be valid. If they disagree, then an error must have occurred either in the neighbour or its neighbours. The checking then continues back in time. Once a cell's neighbourhood is corrected, limited by the number of available histories, the next state can be determined.

A glider is a pattern that recurs, moving across the grid and repeating every four steps. Corruption to any one of the active cells causes the pattern to disintegrate. The cone provides a way of correcting such dynamic patterns (see Figure 1).

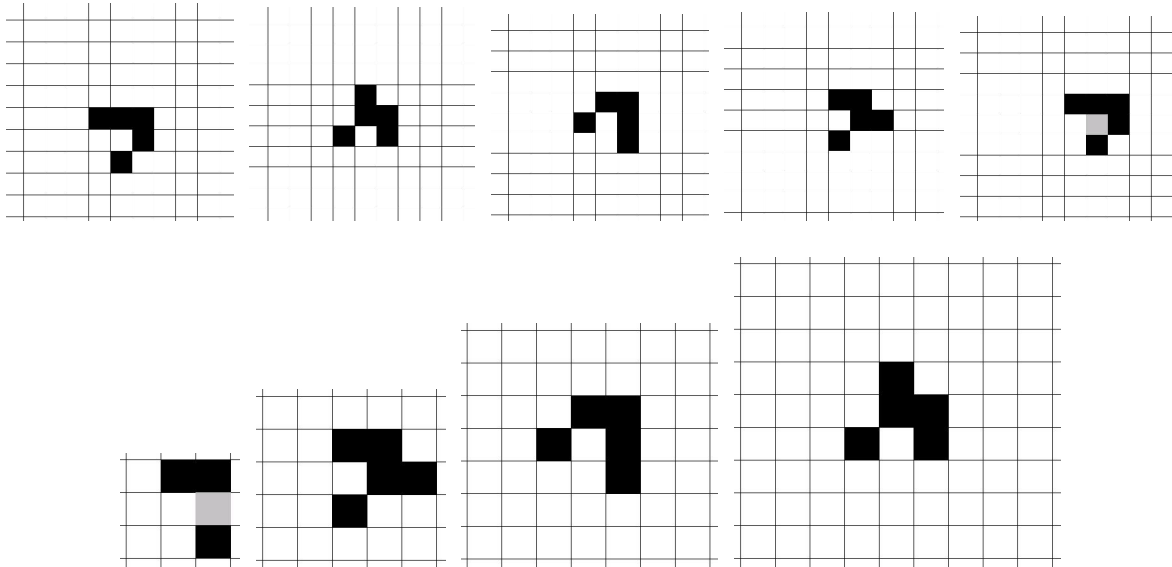


Figure 1. Glider and cone of correction. In the top row, the glider is shown in successive steps crossing the grid. The last image shows an error occurring as the gray cell. The cone for the central cell adjacent to the error is shown going back 3 steps in time (bottom row). As can be seen, the gray cell contradicts the prediction of its previous neighbourhood. But the cells in the neighbourhood agree with their respective neighbourhoods going back 2 more steps in time. So it is likely that the gray cell is an error.

Definitions:

Let a *grain* g be defined as a cell c in a cellular automaton at time t :

$$g = \text{grain}(c, t)$$

Also define *cell*, *time* and *state* functions to extract these quantities from a grain g :

$$c = \text{cell}(g), t = \text{time}(g), s = \text{state}(g)$$

A *cone* for a grain g is a pyramidal set of *layers* of grains, up to height h , each successive layer containing increasing numbers of increasingly older contiguous grains that are centered about g :

$$cone(g) = \{ layer_0(g), layer_1(g), \dots, layer_h(g) \}$$

where the top $layer_0(g)$ contains the 3x3 Moore neighbourhood of g , and successive layers are grains that are neighbours of the previous layer:

$$layer_i(g) = \bigcup_{j=1}^k grain(N_j, time(g) - i)$$

where N is the set neighbour cells in the Moore neighbourhoods of the cells in the previous layer:

$$N = neighbours(cells(L_{i-1}(g)))$$

Thus the radius of each successive layer increases by 1.

Error-correction:

Assign the initial probability that a grain x in $cone(g)$ has an erroneous value caused by noise to be proportional to its spatial distance from g :

$$error(x) = \max(dist(cell(g), cell(x)) * noise, 1)$$

where $dist$ is a city-block distance, and $noise$ is a unit of noise probability.

The probability that a set G of grains contains an error is:

$$error(G) = 1 - \prod_{j=1}^k (1 - error(G_j))$$

Define the $context(x)$ of grain x to be the set of grains in $cone(g)$ containing x 's Moore neighbourhood in the successive layer, i.e., at the previous $time(x) - 1$.

The recursive error-correction algorithm for x is:

correct(x):

if (x in deepest layer) return

for (each grain y in context(x))
do

```

        correct(y)
    enddo

    if (state(x) = gol(context(x)))
    then
        error(x) = 1 - (error(x) * error(context(x)))
    else
        if (error(x) > error(context(x)))
        then
            state(x) = gol(context(x))
            error(x) = error(context(x))
        endif
    endif
endif

```

where *gol* computes a state from a given context neighbourhood according to the Game of Life rules.

The rule checks the state of a grain against what its context (neighbourhood in the previous time step) predicts that it should be. If they agree, then the reliability of the grain is increased by virtue of the support it receives from the reliability of its context. If they disagree, then the state can be corrected by its context only if the context's error probability is lower than that of the grain; otherwise, the context is assumed to be in error and the state of the grain remains unchanged. Optimally, by applying the rule from the deepest layer upward the reliability of the grains in the upper portions of the cone will improve.

Results

The algorithm was coded in Java and a variety of comparison runs made on a 50x50 grid. Each run consisted of 50 trials of 50 steps. For each trial, 10% of the cells in the grid were randomly made live. Two independent variables were selected: range, defined as the maximum cell state signal propagation distance, and hence the cone depth; and noise, a unit of signal distortion probability that multiplies with city-block distance from the central cell. The dependent variable is the number of successful trials compared to an identical noise-free trial. Figure 2 shows the results. A range of 1 represents the Game of Life with no error correction. It can be seen that even at a noise setting of .00005, which translates to an average of only one misread neighbour in the entire grid per time step, there is very little chance of a trial completing successfully, testifying to the fragility of the game.

Increasing the range above 5 effectively corrects for noise settings of .00005 and .0001. However, at higher levels of noise, it appears that error correction cannot be accomplished by increasing the range. This is likely because as a cone grows larger, it encompasses more and more errors within it.

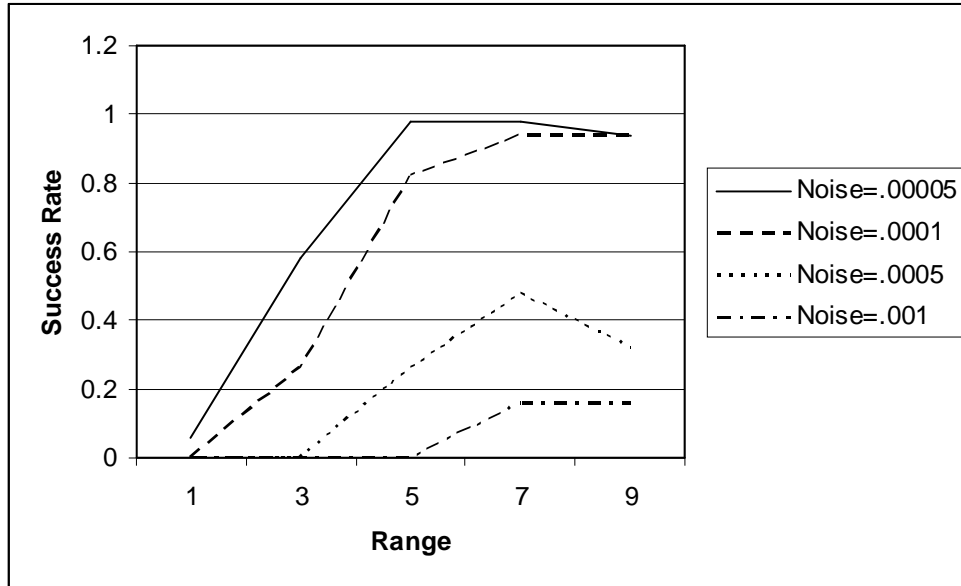


Figure 2. Error-correction performance

In the previous trials, cells noise applies only to cells reading the states of other cells; a cell's own state is read error-free. A situation in which noise also affects a cell's reading of its state can be modeled by adding an "internal distance" to the city-block distance from the central cell in a cone. Thus even the central cell becomes subject to noise errors. The graph in Figure 3 shows the results of an internal distance of 1. The graph shows an expected diminishment of error-correction performance with the additional noise.

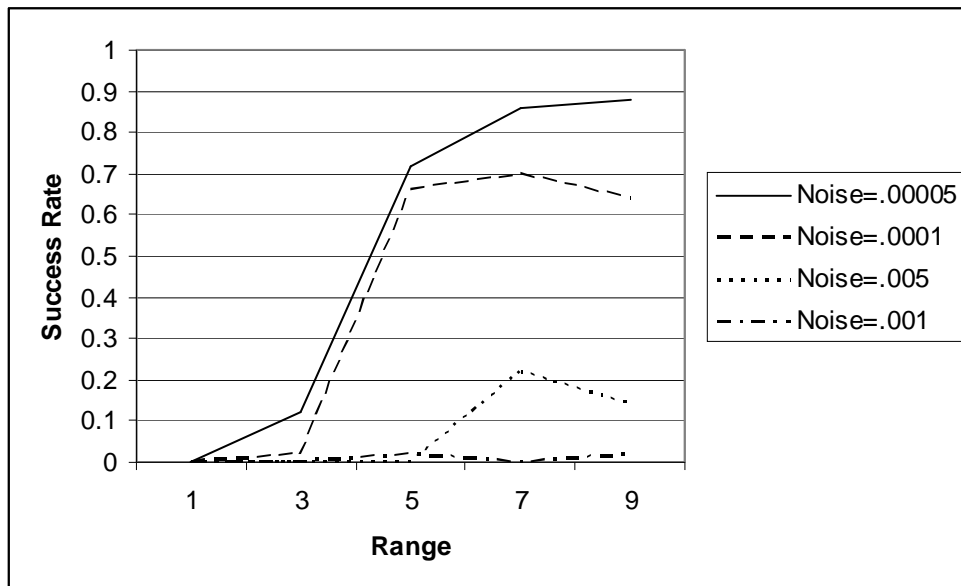


Figure 3. Error-correction performance with internal distance = 1

Discussion and Conclusions

The mechanism we propose for robust behaviour in the Game of Life is based on error-correction of the allowable dynamic trajectories in the system rather than error-correction on the emergent patterns themselves. This approach provides a new way of thinking about the study of robust behaviour in complex systems. The Game of Life has been used as a way of thinking about emergent complexity in biology as well as other systems. Simple transition rules give rise to space time patterns that have interesting properties at a larger scale of space and time. In this study, we have been using the Game of Life to investigate how a complex emergent system could correct for stochastic noise. The method is not based on correcting the possible static patterns at any point in time, but rather correcting the transitions between local components of patterns – the generators that give rise to the emergent complexity. The model provides a way of thinking about the power of context for dealing with stochastic noise.

In biological systems and other real-world domains, the state spaces are typically very large. The overhead of applying error-correction to the dynamic trajectories in a large state transition diagram are prohibitive. However, if the complex behaviour of interest is emergent from a small set of rules, as it is in the Game of Life, then error-correction of the possible dynamic trajectories within local neighbourhoods is a fraction of the size of the emergent state space. From the perspective of a viewer at the global level, the light cone of past admissible states can appear as a way of probabilistically correcting emergent patterns without any knowledge of what those emergent patterns are.

Insights from advanced technology, such as the design of Boeing 747's, are that protocols are more important than modularity for multi-level systems design and maintenance, and this insight has been claimed to apply also to higher level biological organisms (Csete and Doyle 2002). Every model is by nature a simplification and serves to highlight some aspects of a system under discussion, and finesse other aspects. A conjecture arising from our project is that one of the contributing mechanisms to biological robustness is likely to be error correction of the allowable transitions governed by protocols, not just the resultant system states.

In conclusion, we have explored how one particular mechanism of robustness may be studied in the Game of Life. The dynamics of the Game of Life mean that only a subset of the possible patterns occur after the initial burn in. The essential principle of dynamic systems are their dynamics, and in this paper we have considered error correction based on the allowable transitions within a local neighbourhood.

The Java code is available at www.itk.ilstu.edu/faculty/portegys/research/robustlife.zip

References

- Alberts, B., A. Johnson, et al. (2002). The Molecular Biology of the Cell, Garland Science.
- Blok, H. J. and B. Bergerson (1999). "Synchronous vs. asynchronous updating in the Game of Life." Phys. Rev. E **59**(3876).
- Csete, M. E. and J. C. Doyle (2002). "Reverse Engineering of Biological Complexity." Science **295**(5560): 1664-1669.
- Dean, J. S., G. J. Gumerman, et al. (1998). Understanding Anasazi Culture Change Through Agent-Based Modeling. Santa Fe, Santa Fe Institute.

- Di Paolo, E. A. (2001). "Rhythmic and non-rhythmic attractors in asynchronous random Boolean networks." Biosystems **59**: 185-195.
- Freeland, S. J. and L. D. Hurst (1998). "The genetic code is one in a million." Jnl Molecular Evolution **47**(3): 238-248.
- Freeland, S. J. and L. D. Hurst (2004). "Evolution Encoded." Scientific American **April**.
- Gacs, P. (1986). "Reliable computation with cellular automata." Journal of Computer and System Sciences **32**(1): 15-78.
- Gardner, M. (1970). "Mathematical Games: The fantastic combinations of John Conway's new solitaire game "life". " Scientific American **223**: 120-123.
- Grinstein, G., C. Jayaprakash, et al. (1985). "Statistical Mechanics of Probabilistic Cellular Automata." Phys. Rev. Lett. **55**(23): 2527-2530.
- Hallinan, J. and J. Wiles (2004). Asynchronous Dynamics of an Artificial Genetic Regulatory Network. ALife IX The 9th International Conference on Artificial Life, Boston, Ma, MIT Press.
- Harvey, I. and T. Bossomaier (1997). Time out of joint: Attractors in random Boolean networks. Proceedings of the Fourth European Conference on Artificial Life (ECAL97). MIT Press.
- Rouse, J. and S. P. Jackson (2002). "Interfaces Between the Detection, Signaling, and Repair of DNA Damage." Science **297**(5581): 547-551.
- Toom, A. (1974). "Nonergodic multidimensional systems of automata." Problems of Information Transmission **10**: 239-246.
- von Neumann, J. (1966). Theory of Self-Reproducing Automata (edited by A. Burks), University of Illinois Press.