

# Generative Art from One-Dimensional Chip-Firing Automata

Gary R. Greenfield

University of Richmond, Virginia, USA; ggreenfi@richmond.edu

## Abstract

We consider a generative art scheme based on one-dimensional chip-firing cellular automata. Using a variety of search techniques, we identify chip-firing automata with interesting transient phase behavior. Our artworks are visualizations of this transient behavior. We also consider mathematical questions related to the dynamics of one-dimensional chip-firing automata.

## Introduction

Chip-firing cellular automata are derived from the so-called “parallel chip-firing game” which was described by Pavel Etingof in a survey article about student research in the September, 2015 American Mathematical Notices [7] as follows: “Every day each Martian gives each of his friends one Martian peso if he is sufficiently rich to do so. What will happen?” This is a reformulation of a well-studied problem concerning dynamics on graphs. That is, given a finite, undirected graph with non-negative integer weights — the number of chips — assigned to each vertex, simultaneously for all vertices whose weights are at least as large as their degree (i.e., the number of neighbors they have) distribute one chip to each of their neighbors. Since the number of chips is conserved and the graph is finite, it is clear that as the chip-firing process is iterated periodic behavior must occur. The survey article provides a figure showing a graph with six vertices and seven edges and a weight assignment such that after the fifth iteration of chip firing a periodic sequence of length four appears. The reason the problem is mentioned in the survey article is to highlight a journal article by Jiang et al. [13] where the *periodic* behavior of chip-firing games is characterized. Chip-firing games arise in the study of critical group graphs, two-register machines and sandpile models. Propp [15] traces the origins of the chip-firing game to problems first posed by Albert Engel in the 1970s. A recent book by Corry and Perkinson [4] is wholly devoted to chip-firing and its applications.

We will restrict our attention to a very simple case, namely the chip-firing game on the graph consisting of a cycle with  $n$  vertices. Thus every vertex has exactly two neighbors, and the chip-firing game can be realized as a one-dimensional, circular cellular automata where each cell corresponds to a vertex and each cell’s state, or color, corresponds to the number of chips the associated vertex has. Alternatively, we can just write the number of chips in the cell. To formally realize the chip-firing game, using  $m$  chips on a graph that is a cycle with  $n$  vertices, as a one-dimensional, radius one, circular, cellular automata, we define the state transition function  $\tau$  on (l)eft, (c)enter, (r)ight neighborhoods of the form  $\{(l, c, r) : l, c, r \geq 0 \text{ and } l + c + r \leq m\}$  such that the central cell that currently has  $c$  chips has  $\tau(l, c, r)$  chips after the simultaneous chip-firing process has occurred by letting

$$\tau(l, c, r) = \begin{cases} c + 2 & \text{if } l \geq 2, c < 2, r \geq 2 \\ c + 1 & \text{if } l \geq 2, c < 2, r < 2 \\ c + 1 & \text{if } l < 2, c < 2, r \geq 2 \\ c & \text{if } l < 2, c < 2, r < 2 \\ c & \text{if } l \geq 2, c \geq 2, r \geq 2 \\ c - 2 & \text{if } l < 2, c \geq 2, r < 2 \\ c - 1 & \text{if } l < 2, c \geq 2, r \geq 2 \\ c - 1 & \text{if } l \geq 2, c \geq 2, r < 2. \end{cases}$$

If we let  $\sigma_k$  denote the configuration of the cellular automata after  $k \geq 0$  iterations, then the period of a chip-firing automata is the smallest positive integer  $p > 0$  such that  $\sigma_{k+p} = \sigma_k$  for all  $k$  sufficiently large. The transient  $t$  is the smallest nonnegative integer  $t \geq 0$  such that  $\sigma_{t+p} = \sigma_t$ . That is,  $t$  is the number of iterations required until a configuration that eventually repeats occurs and  $p$  is the number of iterations required until the repetition itself is encountered. It is clear that if every cell in the initial configuration has at least two chips or if no cell in the initial configuration has at least two chips, then  $t = 0$  and  $p = 1$ . Based on the observation that once a cell has fewer than four chips, under subsequent iteration it will never have more than three chips, Dall’Asta [6] argues that the number of chips in each cell of an initial configuration should be 0, 1, 2 or 3. We shall adopt this convention here also.

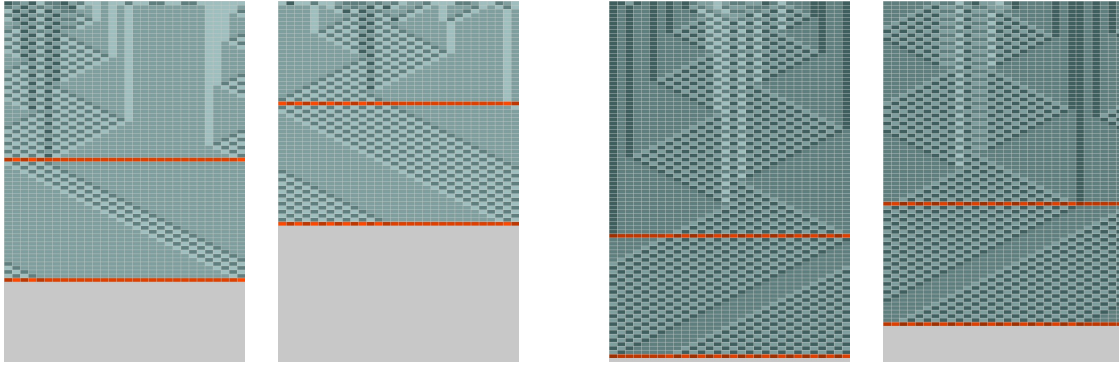
There is a simple construction of an initial configuration such that  $t = 0$  and  $p = n$ . Using  $m = n$  chips, place zero chips in the first cell, two in the second cell, and one chip in all of the remaining cells. With the understanding that the initial configuration  $\sigma_0$  is the first row, and for  $k > 0$ ,  $\sigma_k$  is the  $(k + 1)$ -st row, then for  $n = 6$ , the following chip-firing, or iteration, “history” shows how this works.

0	2	1	1	1	1
1	0	2	1	1	1
1	1	0	2	1	1
1	1	1	0	2	1
1	1	1	1	0	2
2	1	1	1	1	0
0	2	1	1	1	1

To obtain an  $m = 2n$  example such that  $t = 0$  and  $p = n$ , add one chip to every cell in the initial configuration just given. The results proven by Jiang et al. [13] suggest that analyzing periodic chip-firing in general can be reduced to identifying interacting “gliders”. The side by side chip pair 0 2 is the glider in our generic  $m = n$ ,  $t = 0$ ,  $p = n$  example above. Unlike the (chaotic) one-dimensional cellular automata where least upper bounds for the periods are open questions [16], it is known that the periods of chip-firing automata derived from cyclic graphs are bounded by the number of cells [6]. The generic example having period  $n$  notwithstanding, surprisingly, when we tested 10,000 instances of randomly generated initial configurations where  $n = 30$  and  $30 < m < 60$  ALL had period two! Further searching suggested that if  $m < 30$  or  $m > 60$  the automata eventually settle into a “frozen” state where  $p = 1$ . Thus period  $p = 30$  automata could only be found when  $m = 30$  or  $m = 60$ . Note that in these two cases there are unique frozen states that occur when all cells have exactly one chip or all cells have exactly two chips, respectively. By using a guided search to find  $m = 30$  and  $m = 60$  examples with  $p = 30$ , we discovered that the periodic behavior itself for these two cases is visually uninteresting as the four examples shown in Figure 1 suggest. This explains why, for artistic purposes, we focus on the transient phase of chip-firing automata rather than the periodic phase. Our goal is to find initial configurations with interesting transient behavior when  $n = 30$ .

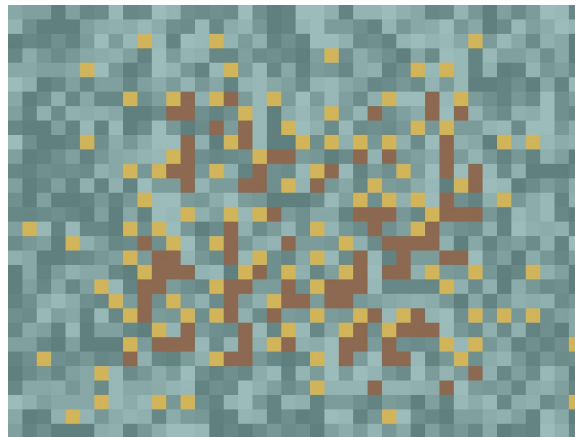
## Cellular Automata and Generative Art

A survey of visual art that incorporates the use of cellular automata, both continuous and discrete, can be found in Adamatzky *et al.* [1]. Bosch has used what is arguably the most famous cellula automata of all, Conway’s Game-of-Life, to create mosaics [2] and animations [3]. Cruz *et al.* [5] used interleaved one-dimensional automata in architectural design. Subsequently, Greenfield modified the concept to evolve two-dimensional grid art [10] [11] (see Figure 2). In these two instances the transition function for the two-dimensional cellular automata are never written down explicitly because cells have sixteen possible states and the transition rules are not arithmetic. Matsumodo *et al.* [14] use Rule 150 cellular automata to identify



**Figure 1:** Four  $n = 30$ ,  $p = 30$  one-dimensional, chip-firing cellular automata. The rows in shades of red highlight repetitions i.e., represent  $\sigma_t$  and  $\sigma_{t+p}$ . The two at the left use  $m = 30$  chips and the two at the right use  $m = 60$  chips. Cell chip counts are coded light blue (zero) to dark blue (three). Left to right, the transient values are  $t = 39, 25, 58, 50$ .

configurations (binary strings) with  $n = 114$ ,  $t = 0$  and  $p = 1022$  such that the resulting iteration history has certain complementarity and symmetry properties in order to produce Mobius scarves (see Figure 3). Feijs and Toeters design one-dimensional cellular automata for generating weave patterns for garments [8]. Fisher uses a variant of one-dimensional automata she refers to as “cellular automata on a staggered grid” as the starting point for her “Pixel Paintings” series (see Figure 4). The mathematical details are proprietary, but a description of her artistic process can be found online [9]. Torrence has used a sandpile model which is, in fact, the analogous two-dimensional version of our one-dimensional, chip-firing automata iterated over a large grid to make “Sandpile Fractiles” (see Figure 5).



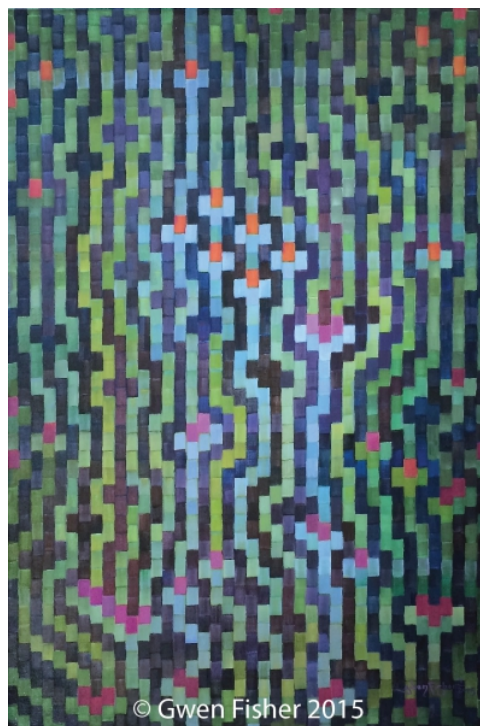
**Figure 2:** G. Greenfield, *Harnessing Chaos #5-9-53023*,  $15 \times 20$  cm, Digital print, 2017.

### Random Search for Transients

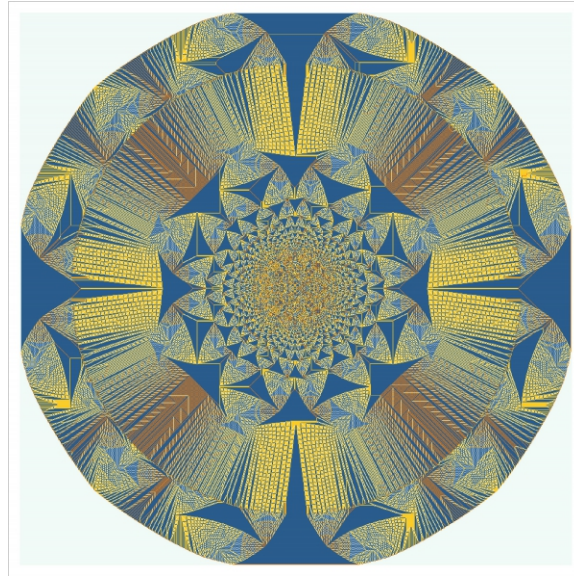
There are many ways to conduct a random search (i.e., a random walk though the space of all our initial configurations for cyclic graphs with fixed  $n$  and  $m$  where  $0 < m \leq 3n$  and no vertex contains more than three chips) to find chip-firing histories with large transients (i.e., large values of  $t$ ). For example, one could generate random initial configurations from scratch by starting with a graph whose vertices have no chips and



**Figure 3:** *Matsumodo, Segerman and Serriere, Mobius Cellular Automata Scarf, 70 × 80 × 5 cm, Merino wool, 2018. Reprinted with permission.*



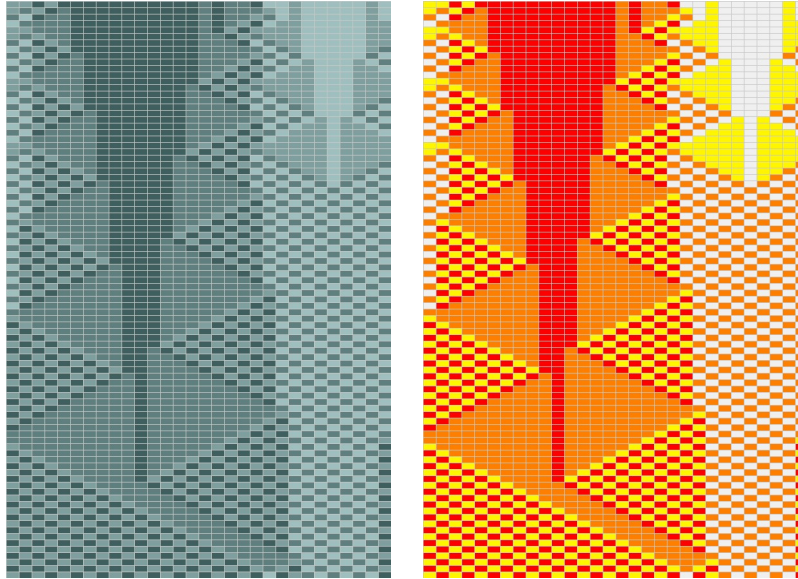
**Figure 4:** *G. Fisher, Pixel Painting I, Water Lilies, 76 × 51 × 2 cm, Acrylic Painting on Canvas, 2015. Reprinted with permission.*



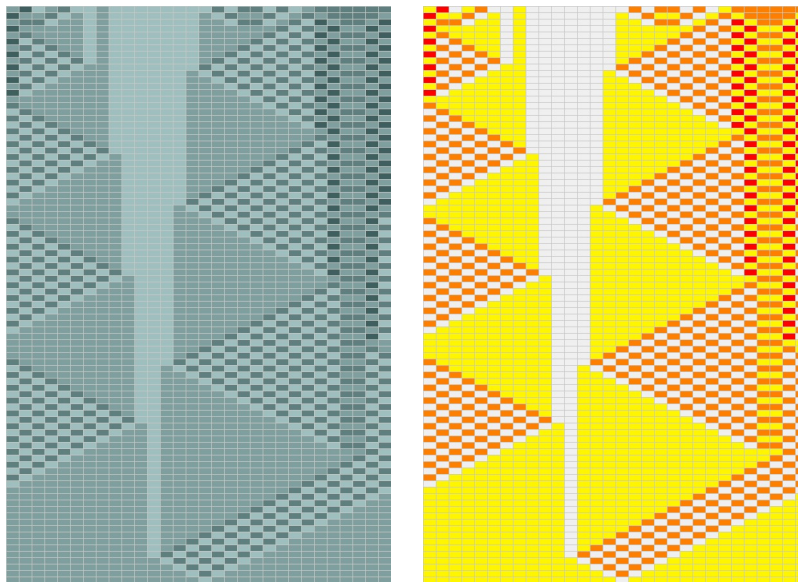
**Figure 5:** *B. Torrence, Sandpile Fractile Disk 60 × 60 cm, Archival Digital Print, 2018. Reprinted with permission.*

invoking an algorithm which performs the following step  $m$  times: Randomly choose a vertex until one is encountered that does not have three chips and increase its chip count by one. Note that this algorithm could be modified so that between one and three chips is assigned to the vertex selected in accordance with how many chips it already has and how many chips remain to be distributed. Or, one could assign chips vertex by vertex according to a probability distribution and then fix things up at the end by adding or subtracting chips from vertices as necessary. A less immediately obvious option is to start with *any* valid initial configuration and then successively “mutate” it using the following step: Randomly choose vertices until one is found that has chips and subtract one chip from it, then randomly choose vertices until one is found that has fewer than three chips and add one chip. It is not too hard to prove that for fixed  $n$  and  $m$ , given any two initial configurations, there exists a sequence of mutation operations that mutates one to the other. In fact, the minimum number of mutations, or “edits”, required to do so defines a distance function that makes the space of all configurations with  $n$  cells and  $m$  chips into a metric space.

We used a combination of these random search methods to sample 100,000 initial configurations with  $n = 30$  and  $30 < m < 60$  and capture the iteration histories for some of the largest transients that we encountered. More precisely, for each batch of 5,000 random initial configurations, interactively, we made a decision about whether to accept or reject the visualization history of the initial configuration that had the largest transient encountered within that batch. Figure 6 is representative of the more visually interesting transient phases that we were able to identify using this naive technique. For  $n = 30$  and either  $m < 30$  or  $m > 30$ , it can be argued that iteration will eventually lead to a frozen state, whence  $p = 1$ . For values of  $m$  near 30, the observation that  $p = 1$  or  $p = 2$  does not affect the ability to identify visually interesting transient phase examples using random search as shown in Figure 7 which has  $m = 29$  and  $t = 89$ . For both figures, on the left we show the example in the “working version” blue color scheme that emphasizes the dynamics, and on the right we show it in our presentation color scheme that emphasizes the chip counts in the cells.



**Figure 6:** The first 90 rows of the iterative history of an  $n = 30$ ,  $m = 53$ ,  $t = 88$ ,  $p = 2$  chip-firing cellular automata (i.e.,  $\sigma_0, \dots, \sigma_t, \sigma_{t+1}$ ) shown (left) in the light blue to dark blue coloring and (right) in an enhanced coloring scheme where white, yellow, orange, red code for 0, 1, 2, 3 chips respectively.



**Figure 7:** The first 90 rows of the iterative history of an  $n = 30$ ,  $m = 29$ ,  $t = 89$ ,  $p = 1$  chip firing cellular automata (i.e.,  $\sigma_0, \dots, \sigma_t$ ) shown (left) in the light blue to dark blue coloring and (right) in the enhanced coloring scheme where white, yellow, orange, red code for 0, 1, 2, 3 chips respectively. Notice how the last row in the image on the right codes for the frozen state by showing all cells in yellow except for one in white.

## Mathematical Considerations

For  $n = 30$  and  $m = 30$ , it is clear that possible periods for one-dimensional, chip-firing cellular automata, with chip counts not allowed to exceed three, include 1, 2 or 30 since we can realize these three possibilities using the initial configurations whose “regular expressions” are  $1^{30}$ ,  $(02)^{15}$  and  $021^{28}$  respectively. Similarly, by using  $2^{30}$ ,  $(13)^{15}$  and  $132^{28}$  we can realize these same three values if  $n = 30$  and  $m = 60$ . More generally, if  $m = n$  or  $m = 2n$ , then there exist initial configurations of period  $d$  for all positive divisors  $d$  of  $n$  as follows. For  $m = n$ , if  $d = 1$  use  $1^n$ , and if  $d > 1$  use  $(01^{n/d-2}2)^d$ . For  $m = 2n$  just increase every digit in the strings for the  $n = m$  case by 1. An argument that the converse is true (i.e., if  $m = n$  or  $m = 2n$ , then the period  $p$  must divide  $n$ ) is given in Dall’Asta [6]. Dall’Asta also explains why  $p = 1$  if  $m < n$  or  $m > 2n$ , and why  $p = 2$  for the remaining possibilities for  $m$ . What about transients? We first make a definition.

**DEFINITION.** Two initial configurations  $(a_1, \dots, a_n)$  and  $(b_1, \dots, b_n)$  are *h-equivalent* if the latter can be obtained from the former either by (right) circular shifting it or by (right) circular shifting its reversal.

Since circular shifting and reversal operations respect iteration histories, it is immediate that if two initial configurations are h-equivalent then their  $t$  and  $p$  values coincide. If  $n = 30$  and the chip count  $m$ , where  $31 \leq m \leq 59$ , is fixed what is the maximum transient value  $t$  that can occur? For each value of  $m$  we made two runs, each considering 10,000 initial configurations, using a simple optimization routine that mutated a random initial configuration until it could be replaced by one that had a larger transient value. The resulting transient values  $t$  we obtained were impressive. They seemed to indicate initial configurations should converge to expressions which when written as regular expression were h-equivalent to  $3^q r 0^{n-(q+1)}$  where, using the division algorithm,  $m = 3q + r$  with  $0 \leq r < 3$ . This makes sense since if we appeal to the one-dimensional sandpile model this is the highest single pile we can make using  $m$  chips when cells can have at most three chips. However, when we computed the transient values for these expressions for  $29 \leq m \leq 61$ , sometimes they matched the best values from our experimental runs, sometimes they exceeded them, and sometimes they fell short. Due to space considerations, the details of this work, as well as a comprehensive definition-theorem-proof treatment of the theoretical claims made herein, can not be included here. For the record, the largest transient value we ever encountered was  $t = 133$  during an optimization run with  $n = 30$  and  $m = 59$ . It was realized by an initial configuration whose regular expression is h-equivalent to  $13^{19}010^8$ .

It is also a daunting challenge to determine the iteration dynamics, up to h-equivalence, for fixed  $n$  and  $m$ . That is, to determine the rooted trees such that the vertices of the trees are h-equivalence classes, the roots are h-equivalence classes whose representatives generate periodic phases, and the paths from the leaves to the roots represent transient phases. In fact, even though the cardinality of an h-equivalence class must divide  $2n$ , just counting the number of h-equivalence classes appears to be a non-trivial problem.

## Summary and Conclusions

We have introduced one-dimensional chip-firing cellular automata and surveyed the chip-firing literature. We reviewed the use of cellular automata in mathematical art. We established techniques for searching for configurations that had large transient phases. We explained how we used visualization to turn large transient phase examples into artworks. We discussed mathematical questions arising in the study of one-dimensional, chip-firing cellular automata.

## References

- [1] A. Adamatsky and G. Martinez. *Designing Beauty: The Art of Cellular Automata*. Emergence, Complexity and Computation, vol. 20, Springer International Publishing, Cham, Switzerland, 2016.
- [2] R. Bosch and J. Olivier. “Game-of-Life mosaics.” *Bridges 2014 Conference Proceedings*, Gary Greenfield, George Hart, and Reza Sarhangi (Eds.), Tesselations, Phoenix Arizona, 2014, 325–329.

- [3] R. Bosch. “Two-frame animations in Conway’s Game of Life,” *Bridges 2015 Conference Proceedings*, Kelly Delp, Craig S. Kaplan, Douglas McKenna, and Reza Sarhangi (Eds.), Tesselations, Phoenix Arizona 2014, 355–358.
- [4] S. Corry and D. Perkinson. *Divisors and Sandpiles: An Introduction to Chip-Firing*. AMS Press, Providence RI, 2018.
- [5] C. Cruz, M. Kirley, and J. Karakiewicz. “Generation and exploration of architectural form using a composite cellular automata.” *Third Australasian Conference, ACALCI 2017, Geelong VIC, Australia, January 31 - February 2, 2017, Proceedings*, M. Wagner et al. (eds.), LNAI 10142, Springer International Publishing, Cham, Switzerland, 2017, pp. 99–110.
- [6] L. Dall’Asta. “Exact solution of the one-dimensional deterministic fixed-energy sandpile.” *Physics Review Letters*, vol. 96, no. 5, 2006, 058003.
- [7] P. Etingof, S. Gerovitch and T. Khovanova. “Mathematics research in high school: the PRIMES experience.” *Notices of the American Mathematical Society*, vol. 62, no. 8, November 2015, pp. 910–918.
- [8] L. Feijs and M. Toeters. “A Cellular Automaton for Pied-de-poule (Houndstooth).” *Bridges 2018 Conference Proceedings*, David Swart, Carlo Séquin, and Kristóf Fenyvesi (Eds.), Tesselations, Phoenix Arizona, 2017, 403–406.
- [9] G. Fisher. <http://gwenbeads.blogspot.com/2015/10/pixel-painting-4-tom-davis-cellular.html>.
- [10] G. Greenfield. “Hex-chaos compositions and equivalence classes of packing problems.” *Bridges 2018 Conference Proceedings*, Eve Torrence, Bruce Torrence, Carlo Séquin, and Kristóf Fenyvesi (Eds.), Tesselations, Phoenix Arizona, 2018, 245–252.
- [11] G. Greenfield. “Interleaved cellular automata, evolved art and packing problems.” *2018 IEEE Congress on Evolutionary Computation (CEC) Conference Proceedings, July 8 – 13, Rio de Janeiro*, Gary C. Yen (Ed.), IEEE, Piscataway, NJ, 2018, 1377–1383.
- [12] J. Guzman and C. Klivans, “Chip-firing on general invertible matrices.” *SIAM Journal on Discrete Mathematics*, vol. 30, no. 2, 2016, pp. 1095–1101.
- [13] T.-Y. Jiang, Z. Scully and Y. Zhang. “Motors and impossible firing patterns in the parallel chip-firing game.” *Siam J. Discrete Math*, vol. 29, no. 1, 2015, pp. 615–630.
- [14] E. Matsumodo, H. Segerman and F. Serriere. “Möbius cellular automata scarves.” *Bridges 2018 Conference Proceedings*, Eve Torrence, Bruce Torrence, Carlo Séquin, and Kristóf Fenyvesi (Eds.), Tesselations, Phoenix Arizona, 2018, 523–526.
- [15] J. Propp. “Prof. Engel’s marvelously improbable machines.” *Math Horizons*, vol. 26, no. 2, 2018, pp. 5–9.
- [16] S. Wolfram. “Universality and complexity in cellular automata.” *Phys. D Nonlinear Phenomena*, vol. 10, no. 1, 1984, pp. 1–35.