Evolution of Complex Emergent Behaviour in Multi-State Cellular Automata

Michal Bidlo
Brno University of Technology
Faculty of Information Technology
Centre of Excellence IT4Innovations
Božetěchova 2, 61266 Brno
Czech Republic
bidlom@fit.vutbr.cz

ABSTRACT

The paper presents a special technique, called conditionally matching rules, for the representation of transition functions of cellular automata and its application to the evolutionary design of complex emergent behaviour. The square calculation in one-dimensional cellular automata and the problem of designing replicating loops in two-dimensional cellular automata will be treated as case studies. It will be shown that the evolutionary algorithm in combination with the conditionally matching rules is able to successfully solve these tasks and provide some innovative results in comparison with the existing solutions. The results represent successful solutions of problems in cellular automata discovered using conditionally matching rules, for which the utilisation of conventional techniques has failed. Original publication: M. Bidlo, "On Routine Evolution of Complex Cellular Automata," in IEEE Transactions on Evolutionary Computation, vol.PP, no.99, doi: 10.1109/TEVC.2016.2516242, URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7377086&isnumber=4358751

Keywords

cellular automaton; transition function; conditional rule; evolutionary algorithm

1. INTRODUCTION

Since the introduction of cellular automata (CA) in [5], researchers have dealt with the problem of how to effectively design transition functions for CA to solve a given task. Cellular automata have been studied both theoretically (e.g. see the extensive work of Wolfram [6]) and practically (e.g. in image processing [3] or design of arithmetic circuits in nanoscale [4]). The process of determining a transition function for a given application represents a difficult task, especially due to an enormous growth of the solution space in dependence on the number of cell states, and due to the fact that the process of "programming" the CA is not intuitive. Therefore, the aim is to automate this process. In this case, we will consider evolutionary algorithms (EA) to design the transition functions.

The goal of this paper is to present a method, called Conditionally Matching Rules (CMRs or conditional rules for short), that allows designing complex emergent behaviour in CA working with more than two cell states. It will be shown that transition functions for the CA, represented by the CMRs, can be designed automatically by EA and innovative solutions can be obtained in some cases in comparison with the existing solutions.

GECCO'16 Companion July 20-24, 2016, Denver, CO, USA

ACM ISBN 978-1-4503-4323-7/16/07.

DOI: http://dx.doi.org/10.1145/2908961.2930947

2. CONDITIONALLY MATCHING RULES

The concept of conditional rules and their evolutionary design is described in detail in [1]. For the purposes of this paper let us consider an example of a 1D CA working with 3-cell neighbourhood. A conditional rule for this CA is defined as $(cond_1\ s_1)(cond_2\ s_2)(cond_3\ s_3) \rightarrow s_{new}$, where $cond_x$ denote condition functions (e.g. $=$, $\leq$, $\neq$ etc.) and s_y denote state values. Each CMR contains a pair *condition and state value* that corresponds to (is evaluated with respect to) a specific cell in the neighbourhood. A finite sequence of CMRs represents a transition function that, for example, contains a rule $(\neq 1)(\neq 2)(\leq 1) \rightarrow 1$. Let c_1, c_2, c_3 be cells in states $2, 3, 0$ respectively, and a new state of $c2$ needs to be determined. The CMRs are evaluated sequentially until a rule is found whose all conditions are true with respect to the states of c_1, c_2, c_3. According to the aforementioned rule, $c_1 \neq 1$ is true as $2 \neq 1$, similarly $c_2 \neq 2$ is true $(3 \neq 2)$ and $c_3 \leq 1$ $(0 \leq 1)$. Therefore, this CMR is said to match, i.e. $s_{new} = 1$ on its right side will update the state of c_2. If no matching rule is found, then the cell keeps its current state. The goal of our experiments was to design CMR-based transition functions by means of evolutionary algorithm. The evolved functions can be transformed to the conventional table rules for the comparison purposes.

3. EXPERIMENTAL RESULTS

The first case study concerned the problem of squaring in 1D CA, i.e. calculating the function $y = x^2$ for arbitrary whole number $x \geq 2$. The value of x is represented as a continuous sequence of cells in state 1, whole length corresponds to x, all the other cells possess state 0. For example, $x = 3$ can be encoded as 000111100000. The result y is required to be a continuous sequence of cells in a single state

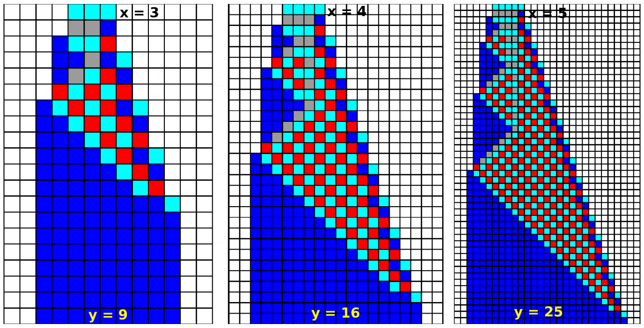

Figure 1: Example of squaring cellular automaton for $x = 3, 4$ and 5 respectively. This result is considered as generic (verified for up to $x = 100$ – in this case the CA needs 29,404 steps to produce the result).

different from state 0, whose length corresponds to x^2, with all the other cells in state 0. The resulting CA state, that is required to be stable, emerges after a finite CA development from the initial state. The evaluation of candidate solutions during evolution was performed for x from 2 to 6, the resulting CA were verified for larger values of x in order to identify generic solutions. Figure 1 shows a CA that was designed for the square calculations. This result exhibits a significant innovation in comparison with the solution published in Wolfram's work [6], page 639. Whilst Wolfram's transition function consists of 51 table rules and needs 78 steps for squaring 5^2, our solution possesses 35 table rules and needs 49 steps only (the difference further grows for larger x). Moreover, Wolfram's CA works with 8 cell states whilst our CA needs 5 states only. Note that other innovative results have been obtained using the CMRs [1].

The second case study investigated the design of non-trivial replication of loop-like structures in 2D multi-state CA. Several benchmark loops of various sizes and arrangements of their replicas were treated. The initial CA state contains a given loop, the goal is to design transition rules for the CA development after which (at least) a given number of copies of the loop emerges from the initial instance. Although the evaluation of candidate solutions was performed on a limited number of CA steps, it is expected that some of the results that fulfil the given requirements will be able to produce more replicas during the further CA development. Several successful solutions were obtained including potentially innovative and still unknown replication processes. For example, a simple rectangular loop whose replication creates a chaotic pattern in the cellular array (see a cutout of the CA in Figure 2) or a structure, called Bidlo's loop or "Bidloop", with the size and complexity similar to Byl's loop [2], which is able to replicate several times faster than Byl's loop (see [1] for details).

4. CONCLUSIONS

The paper presented evolutionary design of complex emergent behaviour in multi-state cellular automata with the transition functions represented by means of conditionally matching rules. Successful (and in some cases also innovative) results were achieved for the generic square calculations in 1D CA and solving non-trivial replication tasks in 2D CA.

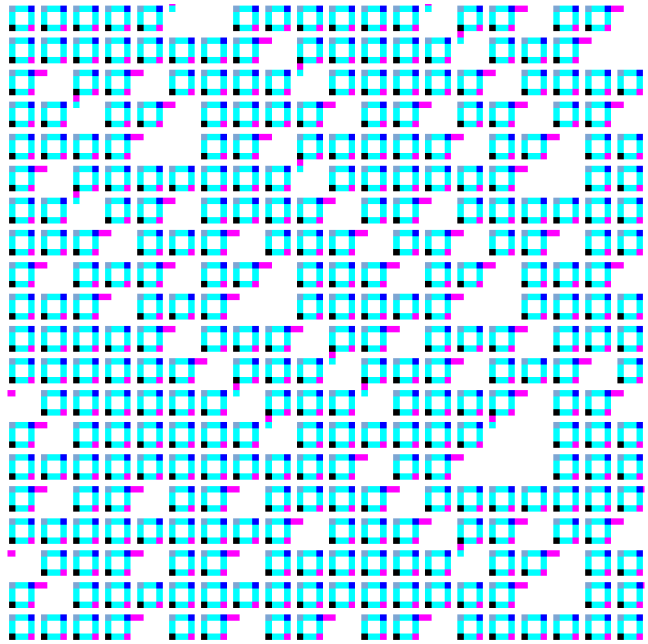

Figure 2: A sample of CA that replicates a simple rectangular loop whose copies exhibit a chaotic arrangement.

These results represent solutions to problems using cellular automata, for which the conventional representation techniques have failed. The concept of conditionally matching rules in combination with evolutionary algorithms showed an ability to design both the transition rules and sequences of steps of the CA in order to achieve the behaviour according to a given specification. We believe that this approach will allow us to solve other (both benchmark and real-world) problems in the future.

5. ACKNOWLEDGMENTS

This work was supported by the Czech science foundation project 14-04197S.

6. REFERENCES

[1] M. Bidlo. On routine evolution of complex cellular automata. *IEEE Transactions on Evolutionary Computation*, PP(99), 2016.

[2] J. Byl. Self-reproduction in small cellular automata. *Physica D: Nonlinear Phenomena*, 34(1–2):295–299, 1989.

[3] P. Rosin, A. Adamatzky, and X. Sun. *Cellular Automata in Image Processing and Geometry*. Springer, 2014.

[4] K. Sridharan and V. Pudi. *Design of Arithmetic Circuits in Quantum Dot Cellular Automata Nanotechnology*. Springer International Publishing Switzerland, 2015.

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

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