A Genetic Approach to Search for Glider Guns in Cellular Automata

E. Sapin, L. Bull and A. Adamatzky,
Faculty of Computing, Engineering and Mathematical Sciences,
University of the West of England,
Bristol BS16 1QY, UK

***Abstract*— We aim to search for cellular automata candidate to an automatic system for the demonstration of collision-based universality and that can be able to simulate Turing machines in their space-time dynamics using gliders and glider guns.**

In this paper, we demonstrate a variety of novel glider guns discovered by genetic algorithms.

I. Introduction

The emergence of computation in complex systems with simple components is a hot topic in the science of complexity [32]. A uniform framework to study emergent computation in complex systems are cellular automata (CA) [29]. CAs are discrete systems in which a population of cells evolves from generation to generation on the basis of local transitions rules [30]. We are interested in the well-established problems related to the emergence of computation in complex systems with local interactions:

- Where are edges of computational universality ?
- How common is computational universality [31] ?

The universality of cellular automata has been tackled in the last thirty years [4], [22], [20], [17], [1] and remains a fruitful area where amazing phenomena at the edge of theoretical computer science and non-linear science can be discovered.

We are here only interested the universality of automata in the Turing sense, i.e., the capacity to simulate a Turing Machine [29] by automata. The most well known universal automaton is the Game of Life [14]. It was shown to be universal by Conway in 1982 [9], who employed *gliders*, mobile self-localized patterns of non-resting states.

Sapin et al [7], [25] discovered a range of universal CA which are universal because they can simulate the Game of Life. This was achieved by using two genetic algorithms (GAs) [19] in sequence searching for automata which exhibit gliders, named R, accepting a *glider gun*, i.e., a pattern which, when evolving alone, periodically recover their original shape after emitting with some gliders [3], [2]. This automaton is shown to be universal in [7], [8] thanks to the presence of the glider gun. Sapin et al.'s algorithm was demonstrated to be capable of discovering cell-state transition rules supporting three glider guns, as shown in figure 1.

In this paper, automata accepting glider guns that spontaneously emerge are searched for. Inspired by Sapin et al.'s approach, a GA is used and we show that, under certain operator combinations, a number of new examples can be discovered.

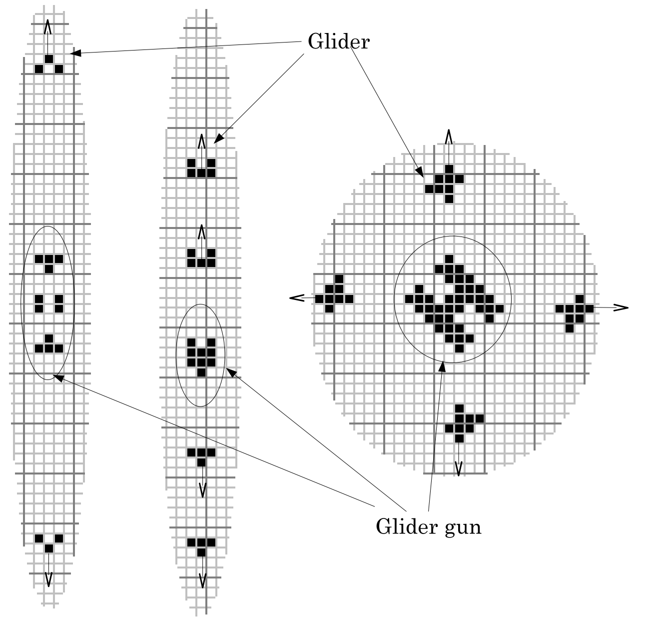

Fig. 1. Three glider guns discovered by Sapin et al.'s evolutionary algorithms.

The paper is arranged as follows: Section 2 describes previous related work. Section 3 sets out the characteristics of the evolutionary algorithms and the results are described in Section 4, along with the presentation of some interesting examples found by the GA. The last section summarizes the presented results and discusses directions for future research.

II. Previous work

In 1970, Conway discovered a special automaton (that he called the Game of Life) that was later popularized by Gardner in [14]. In [9], Conway, Berlekamp, and Guy show that the Game of Life can implement any function calculable by a Turing machine. Their proof of the universality of the Game of Life uses gliders, glider guns, and eaters. A glider gun is a pattern that emits a stream of gliders (used to carry information) and an eater destroys gliders. It is possible to combine glider guns and eaters in order to simulate logic gates and circuits. In [24], Rendell gives an explicit proof

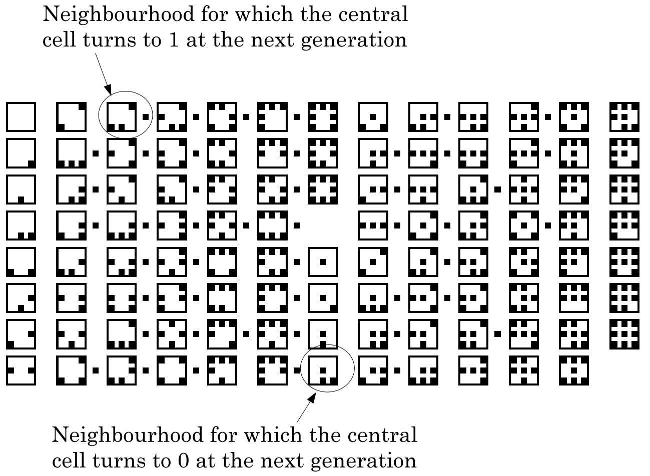

Fig. 2. An automaton of $\mathcal{I}$.The squares are the 102 neighbourhoods describing an automaton of $\mathcal{I}$. A black cell on the right of the neighbourhood indicates a future central cell.

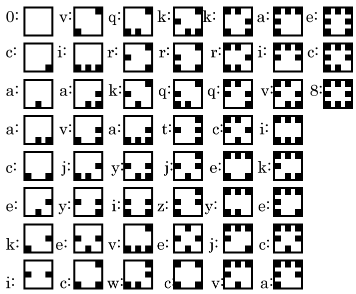

Fig. 3. Neighbourhood states.

of the universality of the Game of Life by showing a direct simulation of counter machines.

In the paper we constrain ourselves to the space $\mathcal{I}$ of 2D *isotropic* CA, with rectangular 8-cell neighbourhoods: if two cells have the same neighbourhood states by rotations and symmetries, then these two cells take the same state at the next generation [33]. There are 102 neighbourhood states that are different by rotations and symmetries, meaning that there are 2^{102} different automata in $\mathcal{I}$. A cellular automaton of $\mathcal{I}$ is shown figure 2.

For conciseness and readability, the following convention has been adopted for the automaton presentation. Each neighbourhood state is refered to by a letter, shown figure 3, and the number of cells from 0 to 8 among it. In the convention, the neighbourhood states that allow the central cell to survive is noticed in the first part after the letter s of the automaton notation and the neighbourhood states that allow the central cell to born is notice in the second part after the letter b. For exemple, the notation $s2iv3a8/b4i8$ means a cell survives for the negihbourhood states $2i$, $2v$, $3a$ and 8.

A. Evolving Cellular Automata

Previously, several good results from the evolution of cellular automaton rules to perform some useful tasks have been published. Mitchell et al. [10], [21], [11], [23] have investigated the use of evolutionary computing to learn the rules of uniform one-dimensional, binary CAs. Here a Genetic Algorithm produces the entries in the update table used by each cell, candidate solutions being evaluated with regard to their degree of success for the given task — density and synchronization.

Sipper [27] presented a related approach, which produces non-uniform solutions. Each cell of a 1D or 2D CA is viewed as a GA population member, mating only with its geographical neighbours and receiving an individual fitness. Sipper shows an increase in performance over Mitchell et al.'s work, exploiting the potential for spatial heterogeneity in the tasks. Koza et al. [6] have also repeated Mitchell et al.'s work, using Genetic Programming [18] to evolve update rules. They report similar results.

The manual search for gliders in cellular automata has been described by A. Wuensche who used his Z-parameter and entropy [34], A. Adamatzky et al. with a phenomenological search [13] and D. Eppstein [12]. Sapin et al.'s work [8], Ventrella [28] and Lohn et al. [16] have considered the emergence of glider-based universality in CAs via a GA.

B. R

As discussed above, in [3], in order to find computationally universal automaton other than the Game of Life, Sapin et al. used an evolutionary algorithm to look for automata in $\mathcal{I}$ exhibiting gliders. Among all the found automata with gliders, some automata also show glider guns. An automaton R, shown in figure 2, where a gun was found was proven to be universal in [7], [25].

The idea of the proof is to simulate the Game of Life by R. The logic universality of R is shown by implementing a NAND gate as well as intersections and synchronizations of patterns. The value of a cell in the Game of Life at the current generation is the result of a logic formula that accepts in entry the values, at the previous generation, of the simulated cell and of its eight neighbours. The automaton R can implement any logic circuit so it can simulate a cell of the Game of Life. The simulation of the Game of Life is then shown by carrying out a tilling of a surface with the identical simulation of cells.

III. Characteristic of the GA

The idea of the GA is to focus on an existing glider as one of the figure 4, named G, and to try to find a gun that emits this glider.

Different selection strategies and different types of crossover and mutation operators are tried. The first subsection describes these parameters. The results of the different parameters are discussed section 4.

A. Parameters

- Initialization
  The search space is the set $\mathcal{I}$ described in Section 2. An automaton of this space can be described by cell-state transition table. An individual is an automaton coded as a bit string of 102 Booleans representing the values

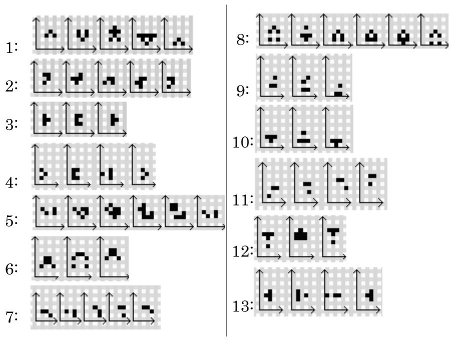

Fig. 4. Gliders.

of a cell at the next generation for each neighbourhood state.
The 102 bits of an automaton are divided into two subsets. The first subset is the neighbourhood states used by the glider G and their values are determined by the evolution of G. The process that determines these neighbourhood states is detailed for the third glider of figure 4, named glider 3, in the figure 5.
The second subset are initialised at random.
The population size is 100 individuals.

- Fitness Function
The choice of the fitness function is a main difficulty in this problem. Lohn et al. [16] have chosen a multiobjective function, Ventrella [28] employed a particle swarm interacting intimately with the cellular automata and Sapin et al. [7] use an image-filtering to detect structures for measuring fitness. The idea of the fitness function is inspired by the one of Sapin et al. [3].
The computation of the fitness function is based on the evolution of a random configuration of cells by the tested automaton A. The parameter of the fitness function is justified in [3]. A random configuration of cells of size 40 is generated. This configuration evolves by the transition rules of the automaton A during 100 generations. After this evolution, the presence of gliders G is checked by scanning the result of the configuration of the cells. For each generation of the glider G, the pattern and its symetrics is puted in a rectangle without living cells, as shown figure 6 for the glider 3. The algortihm count the number of times each pattern appears in the evolving configuration of cells, the number of appearences of the glider is then known. Figure 7 shows the evolution of a random configuration of cells and the result of the test of the glider. The value of the fitness function is the number of gliders that appeared divided by the total number of cells.
- Genetic Operators
The mutation function simply consists of mutating one bit among the second subset of the 102 bits. The rates 1,5 and 10 percent are tried, together with three crossover operators.
  – No crossover

Fig. 5. Detail of the construction of set of neighbourhood states that are used by a glider.

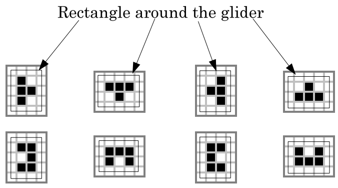

Fig. 6. Patterns of a glider used by the fitness function.

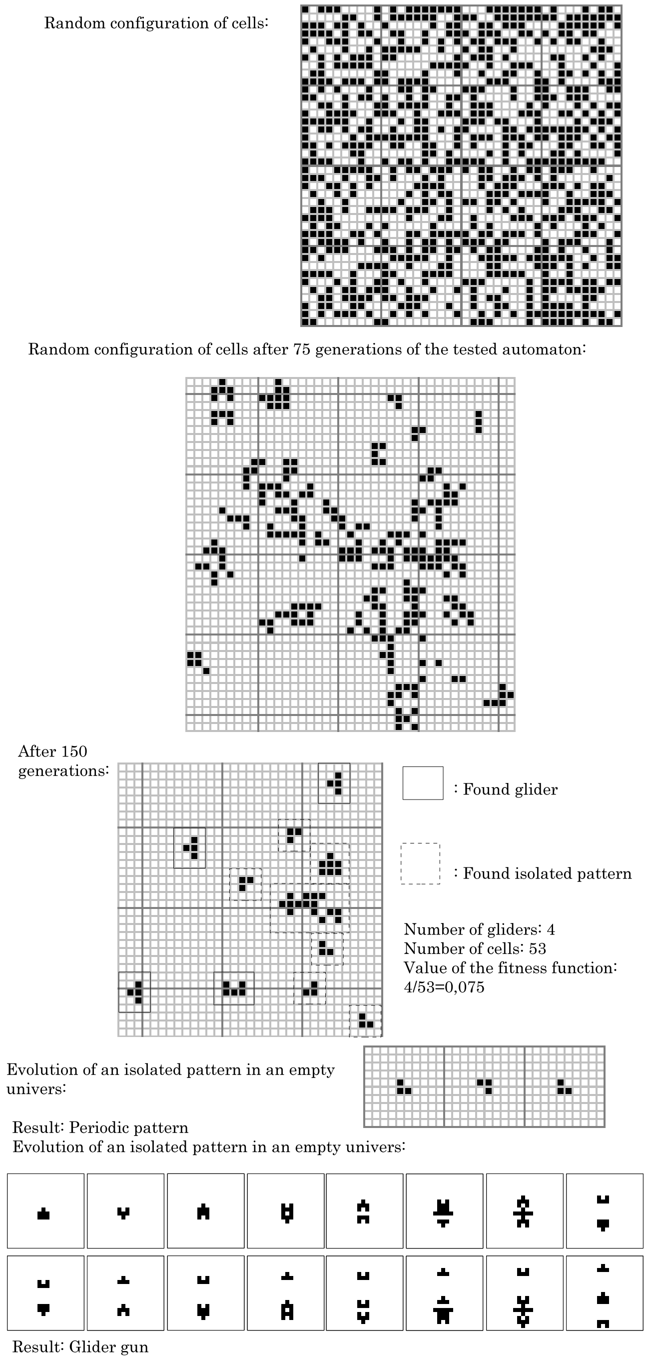

Fig. 7. Exemple of computation of the fitness function and the stopping criterion.

The genetic algorithm is tried without crossover.

  – Central point
    A single point crossover with a locus situated exactly on the middle of the genotype is tried.
  – Random point
    A last kind of recombination is tried with a single point crossover with a locus randomly situated.

A linear ranking selection and a binary tournament selection of size 2 are tried.

- Evolution Engine
  An elitist strategy in which the best half of population is kept and a non-elitist strategy in which the new population is made of only children are tried.
- Stopping Criterion
  The presence of a glider gun is continuously checked. The test is inspired by Bays' test [5] and also used in [2]. After the evolution of the random configuration of cells, the pattern is isolated [2] and tested in an empty universe during 100 generations. If a pattern P reappears at the same place with gliders around then the pattern P is a glider gun. When a glider gun is found the algorithm stops. An exemple of this test is shown figure 7.
  The value of the fitness function and the generation of the best rule are stored. If after ten new generations the algorithm has not found a better rule the algorithm stops.

IV. Results

A. Best Parameters

In order to determine the best parameters, one hundred executions of the evolutionary algorithm with each of the values of the parameters were realized. The algorithm tries to find the gun that generates some specific glider, we have chosen the glider 3.

For each of the values of the parameters, the number of executions which find a gun are shown in table I.

The best parameters, among the tested ones, are a mutation rate of 1, a non elitist strategy, a tournament selection and a central crossover. The good results of the central crossover can be explained by the fact that the first 51 neighbourhood states determine the birth of cells, while the other 51 determine how they survive or die. The elitist strategy that kept half of the population is worse than the non-elitist strategy. An elitist strategy that just kept one parent could be tried, however.

With the best parameters, as noted above, the average variation of the best value of the fitness function for 100 executions is shown table I.

B. First Results

971 different glider guns are found emitting the glider 3. In order to determine if a gun is new, the neighbourhood states used by the given gun is compared to the ones of the other guns. The found guns emit one, two or four streams — as shown in figures 9, 10 and 11 respectively. All the 971 guns can be found in [26] in Life format.

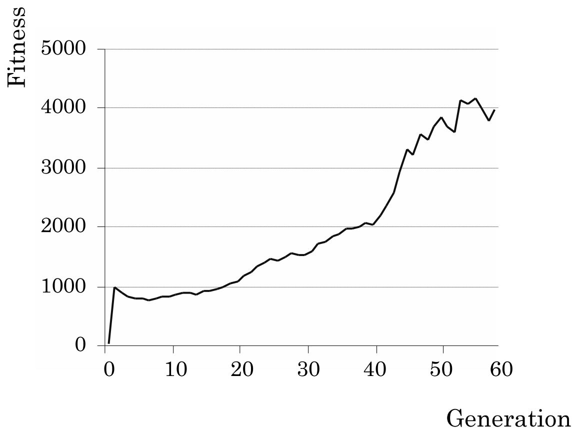

Fig. 8. Average variation of the best fitness for 100 executions with the best parameters.

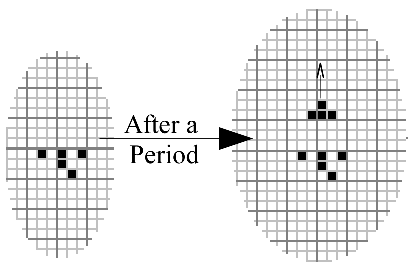

Fig. 9. A gun emitting one stream. This gun can be noticed: s01c2aki3ajec4aywkqjec5kyvi6a7ec8/b2civ3ae4aivrze5krci6viec7ec8.

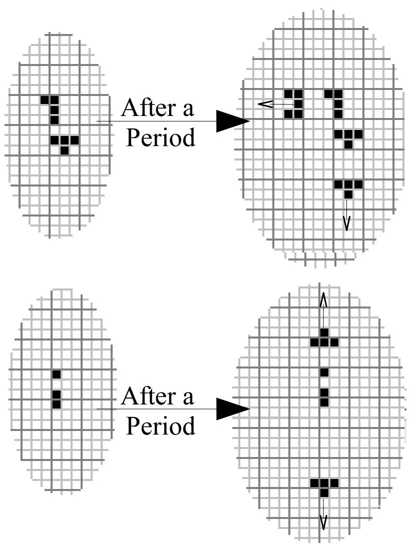

Fig. 10. Guns emitting two streams. These two automata can be noticed: s1c2av3vyek4avwrqtzec5rai6viec8/b2cki3aeqrk4yivwqzec5kejvi6ve7e and s1e2a3ayecqr4avrqtz5rqeji6vec/b2ceiv3ak4arqe5rcvi6kec8.

TABLE I
NUMBER OF EXECUTIONS FROM A TOTAL OF 100 PER EXPERIMENT THAT FIND A GUN UNDER A GIVEN COMBINATION OF PARAMETERS OR OPERATORS. THE THREE NUMBERS CORRESPOND TO THE 1,5, AND 10 MUTATION RATES.

Mutation rate : 1

	Elitist		Non Elitist	
	Tournament	Ranking	Tournament	Ranking
No Crossover	16	14	42	8
Middle Crossover	21	21	64	54
Random Crossover	3	3	4	1

Mutation rate : 5

	Elitist		Non Elitist	
	Tournament	Ranking	Tournament	Ranking
No Crossover	32	14	11	1
Middle Crossover	18	14	20	23
Random Crossover	2	3	2	2

Mutation rate : 10

	Elitist		Non Elitist	
	Tournament	Ranking	Tournament	Ranking
No Crossover	8	13	12	4
Middle Crossover	17	10	19	19
Random Crossover	2	2	0	1

C. *Found Guns*

From 50 different gliders, 10 executions of our algorithm per glider allow us to find guns for 26 gliders. Figure 12 shows 4 gliders and their corresponding guns.

The number of rules that exhibit a gun depends on the number of neighbourhood states used by the giving gun. For example, the first gun of figure 12 uses 37 neighbourhood states and the corresponding glider uses 13 neighbourhood states. So 2^{65} automata exhibit this gun among the 2^{89} automata that exhibit this glider in the space $\mathcal{I}$ described in Section 2 of 2^{102} automata. In the space $\mathcal{I}$, one automaton over 16777216 exhibit this gun among the automata that

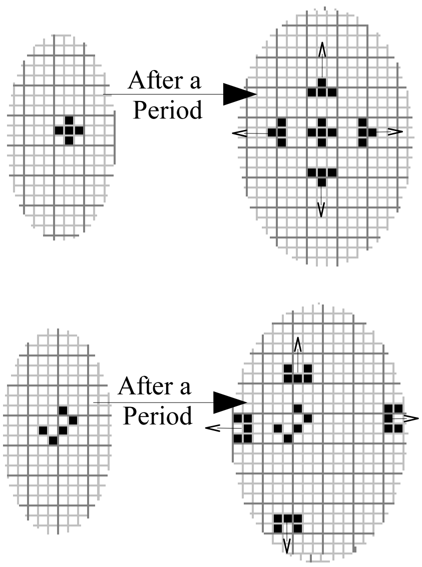

Fig. 11. Guns emitting four streams. These two automata can be noticed: s1c2ai3ajyecqr4yvkrtjzec5kejv/b2cki3aeq4yivwkrqtzec5kei6ike and s1c2aciv3ayecr4ayqtjzec5kqcev6ka7ec8/b2civ3aeq4yiktec5vai6ica7ec8.

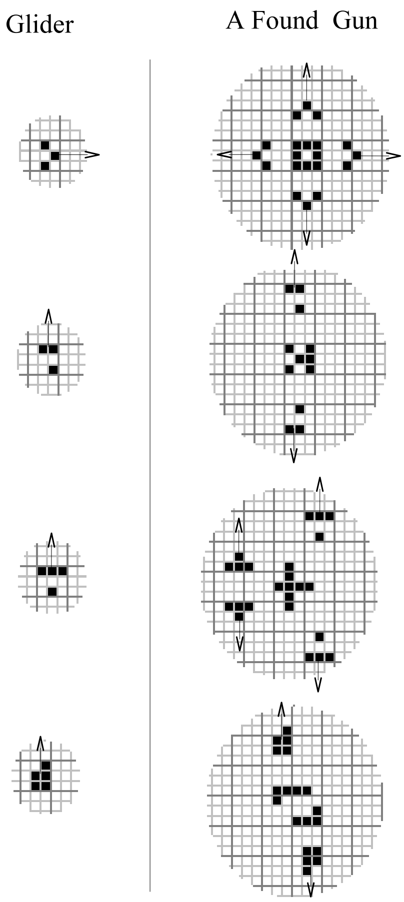

Fig. 12. Four gliders and their guns. These four automata can be noticed: s2acekiv3avyrk4akrqzc5cv6viec7c/b2e3ijecrk4yivwkqjec5kv6viea7e8, s1c2iv3aecqk4ywrtze5kqeyjai6vkea7c/b2a3vecrk4yiwkre5krqcev6ke7c, s1e2civ3iacqrk4ivkrtc5kreyja6via7ec8/b2e3iajer4ivjz5krvai6e7e8 and s2aev3ivjycq4aivwkc5krceva6e7ec/b2e3iark4aijc5cjai6ikca8.

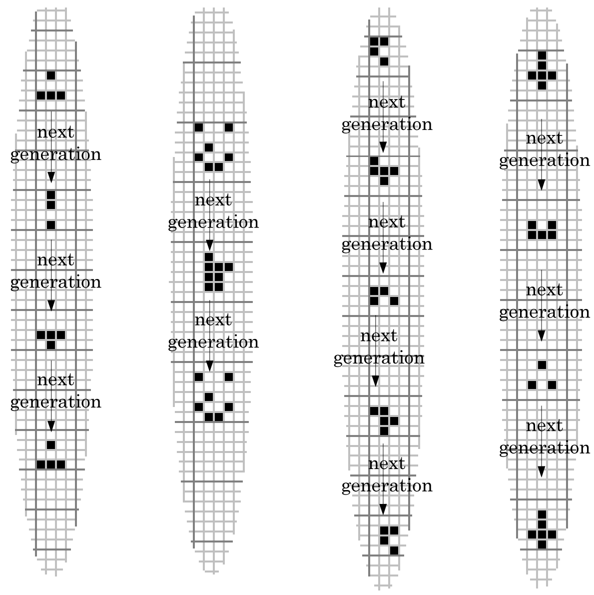

Fig. 13. Gliders for which no guns were found.

exhibit this glider.

Figure 13 shows 4 gliders for which our algorithm did not succeed in finding guns. For these gliders, our algorithm finds rules in which the giving glider emerges from the evolution of cells but not from an emergent gun as the one of the Game of Life [14], [15].

V. Synthesis and perspectives

This paper deals with the emergence of computation in complex systems with local interactions. An evolutionary algorithm searching for glider guns has been presented, building on previous work in [3], [2].

In particular, different evolution strategies and different types of selection, crossover, and mutation operators are explored for this algorithm. With the best parameters, the algorithm finds new guns that spontaneously emerge. For 50 different gliders we found 26 guns. For most of these gliders, no glider gun was known before.

The algorithm succeeded in finding 971 glider guns [26] for the glider called glider 3. Only one gun was known before for this glider, found by Sapin et al. The discovery of the emergence and existence of so many different glider guns for the same glider has only been possible through the use of the GA. Thus the results reported here represent a significant contribution to the area of complex systems that considers computational theory.

Next Work could be to compared the result of the used fitness function to the results of other fitness function and the result of other search methods.

All automata that exhibit a discovered gun may be potential candidates for being universal automata. Further goals are now to develop an automatic system for the demonstration of universal automata. Then, another domain that seems worth exploring is how this approach could be extended to automata with more than 2 states.

Future work could also evaluate all found rules and calculate for each rule some rule-based parameters, e.g., Langton's lamda [19]. All rules supporting glider guns may have similar values for these parameters that could lead to answer at the question 'Where are the edges of computational universality ?' and may therefore lead to a better understanding of the emergence of computation in complex systems with local interactions.

VI. Acknowledgements

The work was supported by the Engineering and Physical Sciences Research Council (EPSRC) of the United Kingdom, Grant EP/E005241/1.

References

[1] A. ADAMATZKY. Universal dymical computation in multi-dimensional excitable lattices. *International Journal of Theoretical Physics*, 37:3069–3108, 1998.

[2] E. SAPIN,. O. BAILLEUX, and J. CHABRIER. Research of a cellular automaton simulating logic gates by evolutionary algorithms. *EuroGP03. Lecture Notes in Computer Science*, 2610:414–423, 2003.

[3] E. SAPIN,. O. BAILLEUX, and J. CHABRIER. Research of complex forms in the cellular automata by evolutionary algorithms. *EA03. Lecture Notes in Computer Science*, 2936:373–400, 2004.

[4] E. R. BANKS. *Information and transmission in cellular automata*. PhD thesis, MIT, 1971.

[5] C. BAYS. Candidates for the game of life in three dimensions. *Complex Systems*, 1:373–400, 1987.

[6] D. ANDRE,. J. R. KOZA,. F. H. BENNETT III and M. A. KEANE. Genetic programming III: Darwinian invention and problem solving. *San Francisco, CA: Morgan Kaufmann.*, 1999.

[7] E. SAPIN,. O. BAILLEUX,. J. CHABRIER, and P. COLLET. Demonstration of the universality of a new cellular automaton. *IJUC*. To be published.

[8] E. SAPIN,. O. BAILLEUX,. J. CHABRIER, and P. COLLET. A new universel automata discovered by evolutionary algorithms. *Gecco2004.Lecture Notes in Computer Science*, 3102:175–187, 2004.

[9] E. BERLEKAMP,. J. H. CONWAY, and R.GUY. Winning ways for your mathematical plays. *Academic Press, New York*, 1982.

[10] M. MITCHELL,. J. P. CRUTCHFIELD, and P. T. HRABER. Evolving cellular automata to perform computations: Mechanisms and impediments. *Physica D*, 75:361–391, 1994.

[11] W. HORDIJK,. J. P. CRUTCHFIELD and M. MITCHELL. Mechanisms of emergent computation in cellular automata. *Parallel Problem Solving from Nature-V, A. E. Eiben, T. Bck, M. Schoenauer, and H.-P. Schwefel (eds.), Springer-Verlag*, 866:344–353, 1998.

[12] D. EPPSTEIN. http://www.ics.uci.edu/~eppstein/ca/, Apr. 2007.

[13] A. ADAMATZKY. G. J. MARTINEZ and H. V. MCINTOSH. Phenomenology of glider collisions in cellular automaton rule 54 and associated logical gates chaos. *Fractals and Solitons*, 28:100–111, 2006.

[14] M. GARDNER. The fantastic combinations of JOHN CONWAY's new solitaire game "life". *Scientific American*, 223:120–123, 1970.

[15] M. GARDNER. On cellular automata, self-reproduction, the garden of eden, and the game of life. *Scientific American*, 224:112–118, 1971.

[16] J. REGGIA. J.D. LOHN. Automatic discovery of self-replicating structures in cellular automata. *IEEE Transactions on Evolutionary Computation*, 1:165–178, 1997.

[17] K. MORITA,. Y. TOJIMA,. I. KATSUNOBO, and T. OGIRO. Universal computing in reversible and number-conserving two-dimensional cellular spaces. *In A. Adamatzky (ed.), Collision-Based Computing, Springer Verlag.*, pages 161–199, 2002.

[18] J. R. KOZA. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press, 1992.

[19] C. L. LANGTON. Computation at the edge of chaos. *Physica D*, 42, 1990.

[20] K. LINDGREN and M. NORDAHL. Universal computation in simple one dimensional cellular automata. *Complex Systems*, 4:299–318, 1990.

[21] P. T. HRABER,. M. MITCHELL and J. P. CRUTCHFIELD. Revisiting the edge of chaos : Evolving cellular automate to perform computations. *Complex systems*, 7:89–130, 1993.

[22] N. MARGOLUS. Physics-like models of computation. *Physica D*, 10:81–95, 1984.

[23] R. DAS,. J. P. CRUTCHFIELD,. M. MITCHELL and J. E. HANSON. Evolving globally synchronized cellular automata. *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 336–343, 1995.

[24] P. RENDELL. Turing universality of the game of life. *In A. Adamatzky (ed.), Collision-Based Computing, Springer Verlag.*, pages 513 – 539, 2002.

[25] E. SAPIN. Approche volutionniste de la recherche d'automates cellulaires universels. *TSI*. To be published.

[26] E. SAPIN. http://uncomp.uwe.ac.uk/sapin/ieee/gun, Apr. 2007.

[27] M. SIPPER. Evolution of parallel cellular machines. *D. Stauffer, editor, Annual Reviews of Computational Physics*, V. World Scientific:243–285, 1997.

[28] J. VENTRELLA. A particle swarm selects for evolution of gliders in non-uniform 2d cellular automata. *In Artificial Life X: Proceedings of the 10th International Conference on the Simulation and Synthesis of Living Systems*, pages 386–392, 2006.

[29] J. VON NEUMANN. *Theory of Self-Reproducing Automata*. University of Illinois Press, Urbana, Ill., 1966.

[30] S. WOLFRAM. Universality and complexity in cellular automata. *Physica D*, 10:1–35, 1984.

[31] S. WOLFRAM. Twenty problems in the theory of cellular automata. *Physica Scripta*, pages 170–183, 1985.

[32] S. WOLFRAM. *A New Kind of Science*. Wolfram Media, Inc., Illinois, USA, 2002.

[33] S. WOLFRAM and N. PACKARD. Two-dimensional cellular automata. *Journal of Statistical Physics*, 38:901–946, 1985.

[34] A. WUENSCHE. Discrete dinamics lab (ddlab),www.ddlab.org, 2005.