Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

RNDr. Peter Banda

Autoreferát dizertačnej práce

Anonymous Leader Election in One- and Two-Dimensional Cellular Automata

(Anonymná voľba šéfa v jedno a dvojrozmerných celulárnych automatoch)

na získanie akademického titulu philosophiae doctor
v odbore doktorandského štúdia:
9.2.1 Informatika

Bratislava 2014

Dizertačná práca bola vypracovaná v externej forme doktorandského štúdia na Katedre aplikovanej informatiky Fakulty matematiky, fyziky a informatiky Univerzity Komenského v Bratislave.

Predkladateľ: RNDr. Peter Banda
Katedra aplikovanej informatiky
Fakulta matematiky, fyziky a informatiky
Univerzita Komenského
Mlynská dolina
842 48 Bratislava

Školiteľ: prof. RNDr. Jiří Pospíchal, DrSc.
Ústav aplikovanej informatiky, FIIT STU, Bratislava

Oponenti: .....
.....
.....
.....
.....
.....

Obhajoba dizertačnej práce sa koná ..... o ..... h pred komisiou pre obhajobu dizertačnej práce v odbore doktorandského štúdia vymenovanou predsedom odborovej komisie 9.2.1 Informatika na .....

Predseda odborovej komisie:
prof. RNDr. Branislav Rovan, PhD.
Fakulta matematiky, fyziky a informatiky
Univerzita Komenského

1 Introduction

In the thesis, we tackled the core distributed problem of leader election introduced by Smith (1971). Given a net of processors, the problem is to design a distributed algorithm that elects a single processor, a leader, starting from an initial configuration where all the processors are in the same state. Leader election is an important prerequisite to many distributed algorithms for such tasks as finding maximal cliques, exploring graphs, and broadcasting. The most common approach is to consider leader election of distinguishable unique processors by applying a search for a processor with particular minimal or maximal ID (Frederickson and Lynch, 1987). The probabilistic Las Vegas algorithms, which assign IDs to processors randomly, may run forever but they terminate within finite time on average (Itai and Rodeh, 1994).

Even though the leader election problem originates in the theory of distributed algorithms, our model and methodology are applied in the unconventional context of multi-agent systems and complexity research. To be able to analyze the system dynamics, we opted for a distributed system with the simplest structure, binary cellular automaton (CA) (Neumann, 1966), which has been extensively used to study various aspects of dynamical systems and artificial life. Despite being one of the structurally simplest distributed systems, a CA can exhibit various types of behavior, including complex dynamics and self-organization (Langton, 1990; Wolfram, 1986). We tackled the most difficult, however most fundamental, variant of leader election, in the minimal, anonymous, and uniform architecture of binary one- and two-dimensional CAs.

Our methodology leveraged evolution of CAs (Crutchfield et al., 2003) by employing genetic algorithms (Mitchell, 1996), an advanced gradient-descent climb, to find (sub)optimal solutions, i.e., CA's transition tables, for leader election. The findings (Banda, 2008, 2009a, 2011) showed that the emergent dynamics of the best CAs is characterized by sophisticated coordination and global computation of cells, a product of spatio-temporal structures or events, namely regular domains, particles and particle interactions, known from the theory of computational mechanics (Crutchfield and Hanson, 1993). The best-performing CAs for leader election showed a remarkably high performance of $0.94 - 0.99$. Our CA model has $O(N)$ time complexity, and each processor (cell) uses just $O(1)$ memory. We also analyzed the dynamics of two-dimensional CAs by stability measures: the Derrida measure (Derrida and Pomeau, 1986; Shmulevich and Kauffman, 2004), and the damage spreading (Rohlf et al., 2007) with a discrete version of Lyapunov stability (Büsing et al., 2010). Our findings agree with the general properties of the complex regime, namely, more complex the CA's dynamics, more successful the leader election. Further, we briefly introduced and analyzed two-dimensional asynchronous leader election.

In the thesis we also asked whether a better-performing CA than we found could exist. We identified general limitations that no one- or two-dimensional CA can overcome (Banda et al., 2014). We showed that a minimal, fully uniform and anonymous (no identifications) architecture of CA cannot produce a correct output from all input configurations. We enumerated such *unsolvable configurations*, both symmetric and loosely-coupled configurations, using linear algebra and group theory and formulated a universal upper bound on

performance for the anonymous one- and two-dimensional leader election problem.

Our findings are directly applicable for design of more effective and robust distributed protocols and networks. We also suggest that by CA-based leader election we could better understand and model biological processes such as morphogenesis of cell differentiation (Nagpal, 2003), where leader election breaks symmetry in a newly formed organism.

2 Background and Definitions

In this section we introduce the basic concepts and methodology that support our cellular automata approach to leader election.

2.1 Cellular Automata

John von Neumman (Neumann, 1966) introduced the concept of a cellular automaton (CA) to explore logical requirements for machine self-replication and information processing in nature. Despite having no central control and limited communication among components, CAs are capable of universal computation and can exhibit various dynamical regimes. As one of the structurally simplest distributed systems, CAs have become a fundamental model for studying complexity in its purist form (Wolfram, 1986; Crutchfield et al., 2003).

By definition, a CA (Codd, 1968) consists of a lattice of N components, called *cells*, with cycled boundaries (toroid topology) and a *state set* Σ. A state of the cell with index i is denoted $s_i \in \Sigma$. A *configuration* is then a sequence of cell states:

$$\mathbf{s} = (s_0, s_1, \ldots, s_{N-1}).$$

Given topology and the number of neighbors n, a *neighborhood* function $\eta : N \to \Sigma^n$ defines the set of cells whose state is accessible (visible) to cell i. Note that usually each cell is its own neighbor. A *transition rule* $\phi : \Sigma^n \to \Sigma$ is applied synchronously to each cell's neighborhood, resulting in an update of that cell's state $s_i^{t+1} = \phi(\eta_i^t)$. The transition rule can be represented either by a transition table or a finite state transducer (Hordijk, 2000). The *global transition rule* $\Phi : \Sigma^N \to \Sigma^N$ is defined as the transition rule with a scope over all configurations

$$\mathbf{s}^{t+1} = \Phi(\mathbf{s}^t).$$

2.2 Leader Election as a Computational Task for Cellular Automata

Our model of a CA has a uniform one- or two-dimensional topology, deterministic uniform transition function, synchronous update, and anonymous cells. We opted for an anonymous instance of leader election because non-anonymous models assume that the system's components are distinguishable, which requires a central entity to assign unique IDs to the

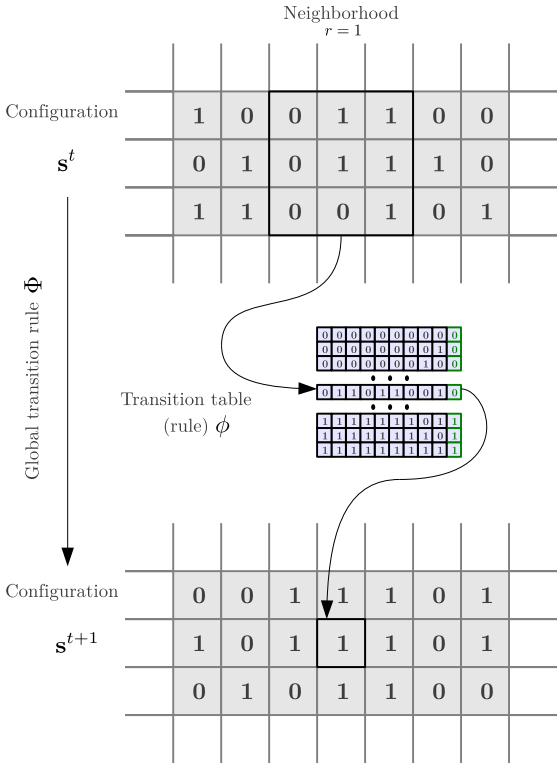


Figure 1: Schematic of the configuration update for a binary two-dimensional cellular automaton, namely Woltz and deOliveira's CA performing density classification.

components. Therefore, each (new) component must register first, which violates the very essence of distributed control.

A CA computational task reflects the relationship between an input (initial configuration) and desired output (final configuration). Any CA implementing leader election must transform an initial configuration to a fixed point configuration with exactly one active cell, where an active cell is a cell in the leader state $1 \in \Sigma$. Note that for a binary CA, all non-active cells must be in the state 0.

Definition 2.1. *The* leader election *task T for the state set Σ (alphabet) and the leader state $1 \in \Sigma$ is*

$$T : \Sigma^N \to \{\mathbf{s} \in \Sigma^N|_{\#_1}\mathbf{s} = 1, \Phi(\mathbf{s}) = \mathbf{s}\},$$

where $\#_1\mathbf{s}$ is the number of 1's in a configuration $\mathbf{s}$.

We say that CA for leader election is *self-stabilizing* whenever the system eventually reaches a correct state with a single active (leader) cell, regardless of the initial configuration. Note that the original definition (Smith, 1971) expects the processors to be initially all in the same state. We showed that the self-stabilizing property cannot hold for all configurations (more in Sections 4.6 and 4.7).

2.3 Evolving Cellular Automata (Related Work)

The idea of applying evolutionary optimization (Mitchell, 1996) to CAs is not new. Several models and techniques have been proposed. Koza (1992) applied genetic programming on CAs to generate simple random numbers. Sipper (1995) evolved non-uniform CAs for the iterated prisoner's dilemma and other problems. Note that nonuniform CAs where each cell has a different transition rule make solving a computational task easier, however, the search space is much larger.

In his pioneering work Packard (1988) employed genetic algorithms to evolve uniform CAs to perform the density classification task. Note that the population of chromosomes, which encode candidate CAs as bit vectors, consists of the transition (look-up) tables' outputs using the standard ordering. In the density classification task a final configuration is expected to consist of all ones if the initial configuration contains more ones, all zeros otherwise. Since Packard, density classification has been widely used as a benchmark task for one- and two-dimensional CAs.

Ganguly et al. (2002) modelled an associative memory (pattern recognition) by so-called generalized multiple attractor cellular automata. They have developed constrained genetic algorithms, with the help of which the evolutionary process can be guided through a special class of additive or linear CA.

The largest contribution to the field of evolutionary cellular automata is, beyond a doubt, due to the former EvCA (Evolutionary Cellular Automata) and CM (Computational Mechanics) groups at Santa Fe Institute led by M. Mitchell and J. Crutchfield. The EvCA and CM groups evolved one-dimensional binary CAs to perform the density classification (Crutchfield et al., 2003) and synchronization (Das et al., 1995) tasks.

More recently, a two-dimensional variant of the density classification has been successfully tackled by several authors (Marques-Pita and Rocha, 2008). Other tasks for two-dimensional CAs include synchronization, spatial density niching, and rectangle image bounding (Cenek, 2011).

2.4 Computational Mechanics

Since CAs are completely discrete, it was quite difficult to analyze their behaviors with instruments known from the theory of conventional dynamical systems. The methodology of computational mechanics (Crutchfield and Hanson, 1993) finally bridged this gap by synthesising the concepts from both computational and dynamical system theories. The global, collective dynamics of CA can be, therefore, understood and described in terms of space-time structures: domains, particles, and particle interactions.

A *regular domain* is homogeneous space-time regions that are temporally invariant and forms a background for a collective CA computation. A *particle* is a spatially localized and temporally periodic structure at the boundary of two domains with limited width, which transfers information on distances. A *particle interaction*, such as, $\alpha + \beta \rightarrow \gamma$, processes information encoded in colliding particles.

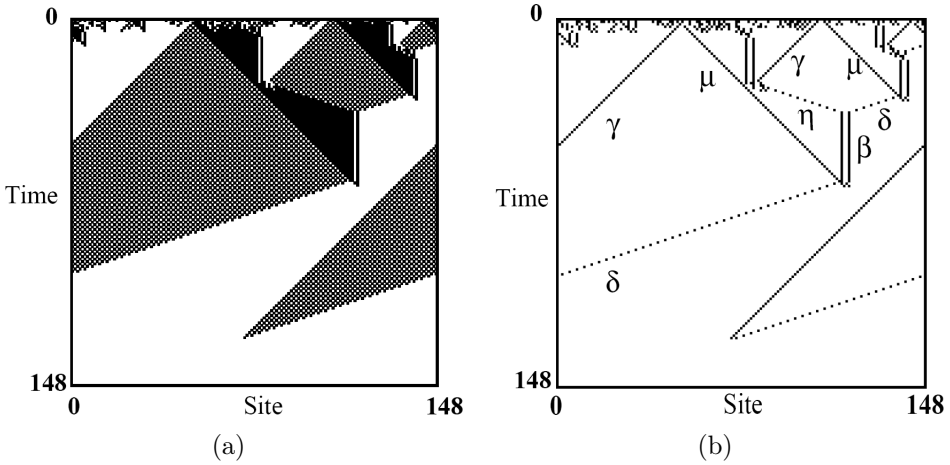


Figure 2: (a) Space-time diagram and (b) filtered version with particle identification (Crutchfield et al., 2003).

2.5 Derrida Measure

The Derrida measure (Derrida and Pomeau, 1986) traces the convergence or divergence of two perturbed configurations after one update. Given perturbation strength $p \leq N$, we first generate an initial configuration $\mathbf{s^1}$ using uniform distribution and create another initial configuration $\mathbf{s^2}$ by randomly flipping p bits in $\mathbf{s^1}$. The Hamming distance d_t is therefore $d_t = d(\mathbf{s^1}, \mathbf{s^2}) = p$. Next, we run the CA on both configurations and again calculate the Hamming distance at time $t+1$ as $d_{t+1} = d(\Phi(\mathbf{s^1}), \Phi(\mathbf{s^2}))$. Now, the Derrida measure (Shmulevich and Kauffman, 2004), an annealed approximation of the system's stability, is obtained by plotting d_t against d_{t+1} for different starting Hamming distances averaged over sufficiently large numbers of samples and normalized by the system size N.

The Derrida measure describes how fast a perturbation spreads to other cells in one time step, where the identity line $d_t = d_{t+1}$ separates the dynamical regimes. The curves below the identity line correspond to ordered systems, which are not sensitive to perturbations, so information does not spread and eventually die out. If the $\frac{d_{t+1}}{d_t}$ ratio is larger than one the CA is chaotic, and so even a small state change diverges and disturbs the system's dynamics. Finally, the curves tangent to the identity line are critical. In this region, a perturbation will not die out nor spread out. The critical regime has been shown optimal for information processing and computing. Furthermore, the most important part of the Derrida plot is for the small Hamming distances. The more the Derrida curve lies above the main diagonal for small values d_t, the more chaotic the system.

2.6 Damage Spreading

Similarly to the Derrida measure, the damage spreading measure (Rohlf et al., 2007) is a form of sensitivity analysis. The idea is to track how fast the smallest possible state

perturbation of a single bit spreads throughout the CA. By definition, the damage spreading $\overline{d}(T)$ is simply the Hamming distance of two trajectories of the same system after time T starting from two configurations that differ only in a single position (bit) averaged over multiple runs. The curves show how fast a perturbation disturbs the system.

The most interesting part of these curves is the initial (steep) portion, which shows the rate of convergence or divergence. That tells us how information transmission and noise propagation are embedded in the system. In fact, $\overline{d}(1)$, the Hamming distance after a single update, could be used for calculation of a variant of Lyapunov stability (exponent) for discrete systems (Büsing et al., 2010). More precisely, the Lyapunov exponent $\lambda = \lim_{t\to\infty} \frac{1}{T}\left(\frac{d(T)}{d(0)}\right)$ could be approximated as

$$\lambda = ln(\frac{\overline{d}(1)}{\overline{d}(0)}).$$

Since $\overline{d}(0)$, the size of initial perturbation, is 1, $\lambda = ln\,\overline{d}(1)$. The critical line for the Lyapunov exponent is $\lambda = 1$. For $\lambda > 1$ the system's trajectories tend to diverge; the system is unstable and chaotic. For $\lambda < 1$ the system converges or contracts, hence it is stable and ordered. At the threshold of these regions, around $\lambda = 1$, complex dynamics combining ordered and chaotic properties occurs. This region promotes efficient information transfer and processing.

3 Dissertation Proposal and Goals

The main goal of the dissertation was to explore the minimal, anonymous, uniform instance of leader election in the architecture of one- and two-dimensional binary CAs. The partial goals were to

- Evolve one- and two-dimensional CAs by genetic algorithms for the leader election problem.
- Analyze the best-performing one-dimensional CAs by the theory of computational mechanics.
- Calculate performance upper bound for one- and two-dimensional leader election.
- Analyze the best-performing two-dimensional CAs by the perturbation stability measures.
- Evolve and evaluate two-dimensional asynchronous CAs for the leader election problem.

4 Dissertation Results and Contributions

In the dissertation we successfully demonstrated that even a cellular automaton, mere distributed system consisting of indistinguishable and uniform cells operating just with a binary state, is capable of complex dynamics enabling global coordination of cells and ultimately leader election. To the best of our knowledge it is the simplest resource-efficient model of leader election using only $O(1)$ (binary state) memory per each cell and $O(N)$ time complexity. The main contributions are as follows.

4.1 Evolution of One-Dimensional Leader Election

Here we solved leader election in one-dimensional CAs by employing genetic algorithms (GA), a stochastic optimization tool, inspired by Darwin's classical theory of natural selection. This work has been published in parts in (Banda, 2008, 2009b, 2010, 2011).

A chromosome, the main constituent of GA, represents a transition function ϕ shared globally by all cells. More specifically, a chromosome is a linear vector of length $|\Sigma|^n$ consisting of the output bits of a transition table, where Σ is the state set and n is the number of cells in a neighborhood. Since we deal exclusively with binary CA, $|\Sigma| = 2$ and a transition table (chromosome) consists of 2^n rows. The size of the chromosome space in which the GA searches is, therefore, 2^{2^n}.

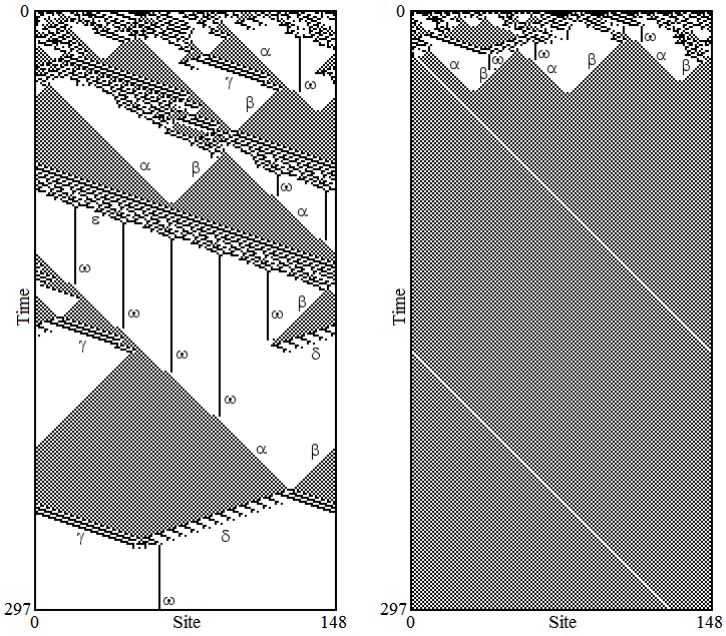


Figure 3: The strategy of mirror particles (a one-dimensional CA): space-time diagrams of (a) successful and (b) unsuccessful leader election.

Performance (fitness) is calculated as the fraction of correctly computed initial configurations to the total number of randomly drawn test initial configurations. Following Definition 2.1 we assigned the tested CA a point only if its final configuration is a fixed point with exactly one active cell. Chromosomes, i.e., candidate look-up tables undergo selection, cross-over and mutation. We used elite selection, one-point and shuffle cross-over, and per-bit mutation.

The evolutionary search uncovered various strategies, which we called the strategy of mandatory function, the density reduction, the divide and eliminate strategy, the first particle-based strategy, and the (improved) strategy of mirror particles.

The first three strategies are localistic and often exploit the statistical properties of generated ICs, such as mean and variance. There is no long-distance information exchange, so a leader is elected only on short subsequences of cells. That results in a sharp decrease of performance with respect to N. The last two strategies leverage a particle collision-based model of the computational mechanics, allowing a coordination of cells at the global scale. The most successful (improved) strategy of mirror particles reached a performance of 0.997 for uniform and 0.945 for density-uniform distribution and $N = 149$.

4.2 Dynamics Analysis of One-Dimensional Leader-Electing CAs

We analyzed the dynamics of one-dimensional leader-electing CAs by computational mechanics. For each strategy we automatically identified domains, particles, and their interactions (Section 2.4). This work has been published in parts in Banda (2008, 2009b, 2010, 2011).

The best-performing strategy of mirror particles is characterized by the occurrence of pair-like particles moving with the same speed but in opposite directions. We call these *mirror particles*. These include particle pairs α and β, γ and δ, ε and ζ. They all lie at the border of domain 0^* and the zig-zag domain $(01)^*$.

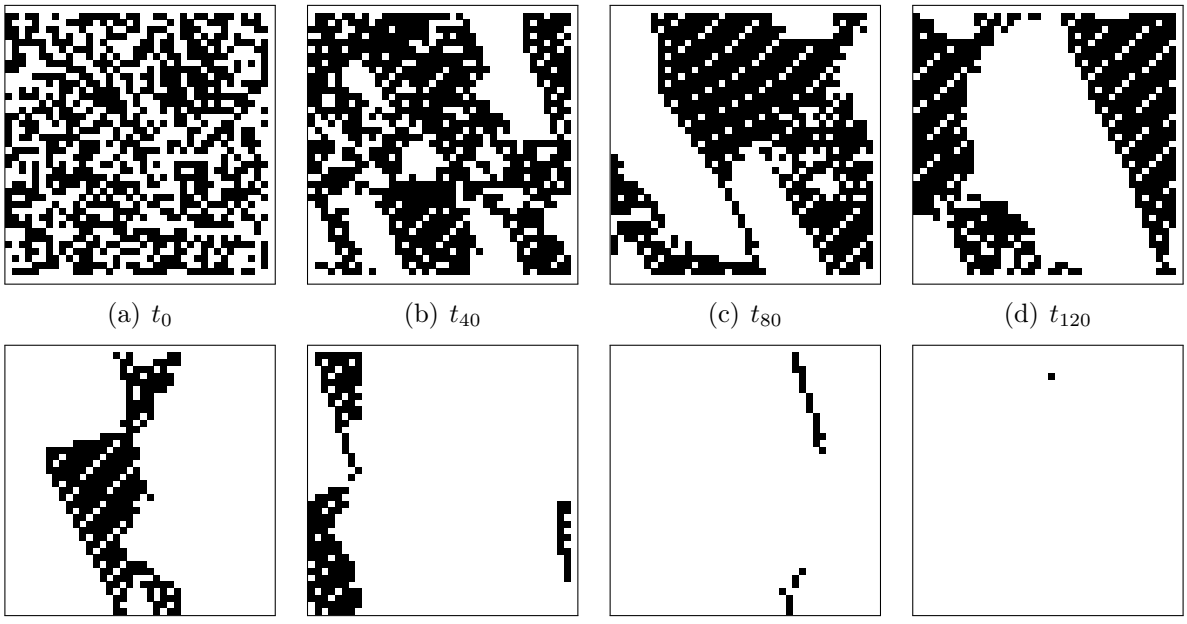
The strategy of mirror particles resolved the problem of two or more active cells in a final configuration by a two-phase sweep consisting of $\alpha + \beta$ and $\gamma + \delta$ interactions. More precisely, the main rule responsible for leader (particle ω) election is the interaction $\alpha + \beta \to \gamma + \delta$ followed by $\gamma + \delta \to \omega$. A collision of α and β indicates that the final stage of leader election could start. Particles γ and δ are emitted to verify if there are any particles left. They turn around the whole configuration with high opposite velocities. In case they do not collide on their routes, they meet in the middle, and finally produce a global leader ω.

The number of particles with positive velocities equals those with negative ones. All particles are fairly fast. Their absolute velocities range from 1 to 3 excluding a non-moving particle ω. Also, the differences of colliding particles' velocities are high. That has a positive impact on the overall performance.

Fitness of this evolutionary strategy is 0.99; performance reaches 0.944 and 0.992 for $N = 149$ depending on the IC distribution type, and stays high for much larger N. For $N = 999$ cells performance is extremely low; only about 0.01, yet for even larger N, such as, 1001 and 1301, it rebounds to 0.97 for density-uniform and 0.99 for uniform distribution.

An explanation of this anomaly is the principal restriction on the number of cells N. The strategy of mirror particles produces satisfactory results for the leader election problem only for $N \equiv 5 \bmod 6$, $N \geq 23$. A goal to produce a single leader ω in a final configuration can be achieved by the interaction $\gamma + \delta \rightarrow \omega$ occurring for $N \equiv x \bmod 6$, $x \in \{2, 5\}$. Further, a pair of particles γ and δ needed for this interaction are produced from α and β only for $N \equiv x \bmod 6$, $x \in \{1, 3, 5\}$. As a result, the only acceptable number of cells allowing leader election (with satisfactory result) is 5 mod 6. CA dynamics for the modulo classes 0, 2 and 4 produce a global zig-zag domain Λ^1, the remainder 5 leads to a stable point of the leader particle ω, and the remainders 1 and 3 result in a cyclic behavior $\alpha + \beta \rightarrow \gamma + \delta \rightarrow \alpha + \beta \rightarrow \ldots$.

The improved strategy of mirror particles reaches 0.997 performance for uniform distribution. Nevertheless, higher performance is offset by an additional N-modulo restriction. Namely, this strategy is usable only for the $N \equiv 5$ modulo 12 number of cells.



(a) t_0 (b) t_{40} (c) t_{80} (d) t_{120}

(e) t_{160} (f) t_{180} (g) t_{200} (h) t_{212}

Figure 4: Example space-time diagrams of the best-performing leader-electing two-dimensional CA targeting $N = 29^2$ on lattice size $N = 40^2$. Figures show a CA computation starting with a uniform initial distribution (time t_0), followed by 7 state snapshots. The CA reaches a final configuration with a single active cell at time t_{212}.

4.3 Evolution of Two-Dimensional Leader Election

We tackled leader election in two-dimensional CA with square lattices, and Moore neighborhoods, i.e., square neighborhoods with $r = 1$ and 9 cells. Because the evolution of

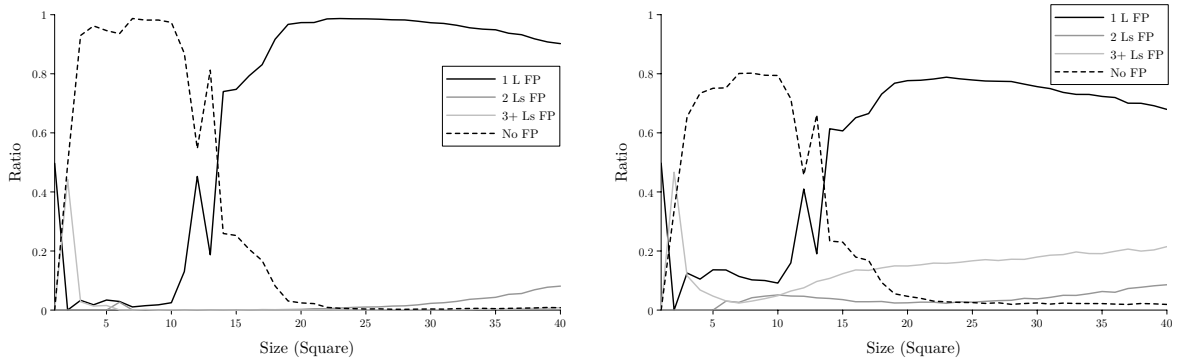
two-dimensional leader election is generally more difficult than the one-dimensional case, we improved several key attributes of the GA, such as cross-over type, population size, and number of test configurations.

For two-dimensional leader election we executed four sets of evolutionary runs: the leader election with $N = 19^2$ (LE), the density minimization task (DM), and two leader election tasks starting from an initial population generated from the last best chromosomes of DM, one with with $N = 19^2$ (LE 19^2) and another with $N = 29^2$ (LE 29^2). Because leader election in two dimensions is very rare, to aid the evolutionary search, we introduced an intermediate task, the density minimization (DM), which continuously reduces the density (the ratio of ones) in a final fixed point configuration.

Since the LE 29^2 task targets largest system's size, performance of the best leader-electing CA stays above 0.9 for $N = 19^2$ up to 40^2. Performance for $N = 19^2$ is 0.9672 for uniform and 0.768 for density-uniform initial distributions. The system scales well and for $N = 39^2$ performance reaches 0.9078 for uniform and 0.6918 for density-uniform distribution (Figure 5).

This is the best performing and most scalable CA, yet it is also the slowest. Time complexity is, however, due to two-dimensional information spreading, sublinear. A larger execution time (compared to the poor-performing CAs) is caused by propagating regions that move around the lattice, contract, and eliminate any remaining active cells.

During leader election the initial random configuration with ones and zeros uniformly distributed splits into different slowly-contracting regions, which keep connected by lines of active cells. These regions propagate left, sweep any remaining active cells, transform



(a) uniform, $t_{MAX} = 1000$ (b) density-uniform, $t_{MAX} = 1000$

Figure 5: Performance of the best leader electing CA targeting 29^2 for the square sizes $N = 1^2, \ldots, 40^2$ calculated as an average over 10^4 runs using uniform and density-uniform initial distributions. The maximal time t_{MAX} allowed for leader election is 300 in (a) and (b), and 1000 in (c) and (d). Figures (a-d) show the ratio of runs that end in a fixed point with a single active cell (1 L FP), two active cells (2 Ls FP), three or more active cells (3+ Ls FP), or no fixed point (No FP). Figure (e) plots the number of ones in a final configuration.

to a single moving line that shrinks from both sides, and finally contracts to a single active cell. Solid performance is due to the occurrence of high-density regular domains $(111110)^*$ and $(11111110)^*$. These domains contract at a slower pace, and so reduce the risk of disconnection. Having more active cells in a configuration helps it to keep propagating and contracting fronts together. Interestingly, because of an initial increase in density, the system is too unstable for small lattices for which the CA cycles and never reaches a fixed point.

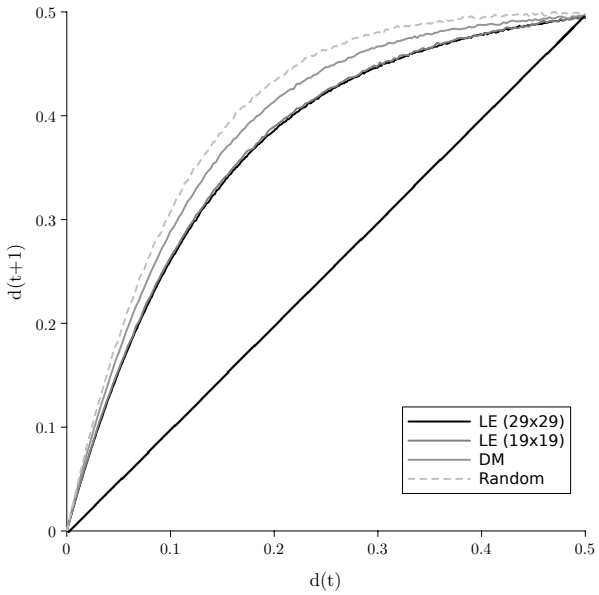


Figure 6: Derrida curves for 100 randomly generated two-dimensional CAs with Moore neighborhood, and the last best CAs from all evolutions for the density minimization task (DM), and the leader election task with $N = 19^2$ (LE 19^2) and $N = 29^2$ (LE 29^2). Averages over all CAs and 100 configurations per each Hamming distance d_t are plotted. Note that the identity line represents the critical dynamical regime, i.e., the closer to the line, the more complex the dynamics. The portion shown in the plot is limited to $d_t \leq 0.5$.

4.4 Dynamics Analysis of Two-Dimensional Leader-Electing CAs

We calculated perturbation stability of two-dimensional leader-electing CAs by the Derrida measure (Section 2.5) and the damage spreading (Section 2.6).

We evaluated the Derrida measure for randomly generated two-dimensional CAs with Moore neighborhoods, and the best CAs for the three evolutionary tasks we performed. As shown in Figure 6 Derrida curve for random CAs is far above the line of criticality, hence even a small perturbation results in trajectories that diverge rapidly, and the dynamics is chaotic. The dynamics of the density-minimizing CAs are more ordered, since their Derrida curve lies bellow the curve for random CAs. The density minimization task is, however,

not as challenging as leader election, whose Derrida curve is closest to the identity line and its dynamics are, therefore, most complex. In a nutshell, the more complicated the task, the more complex the CA dynamics required to solve the problem.

The results for damage spreading and Lyapunov stability are consistent with the Derrida measure. The CAs from LE 29^2, solving the most complicated instance of the problem, requiring global coordination of cells, have low $\lambda = 1.288$ showing their dynamics are the most complex. The CAs of the LE 19^2 task has slightly higher $\lambda = 1.305$, the DM is already quite chaotic and reaches $\lambda = 1.414$, and finally randomly-generated CAs are the most chaotic with $\lambda = 1.499$.

4.5 Evolution of Asynchronous Two-Dimensional Leader Election

We evolved and analyzed leader election in an asynchronous CA, where the global update function Φ is implemented by invocation of each cell's update $\phi(\eta_i)$ independently in a random order (permutation of cells) during each global step.

Our results showed that asynchronous leader election is indeed possible, and for $N = 19^2$ performance of the best CA reaches 0.965 for uniform and 0.735 for density-uniform initial distributions. Yet because of nondeterministic update, the asynchronous CA could not benefit from regular propagated patterns enabling coordinated exchange of information over distances in the synchronous case. Therefore, the strategy of asynchronous leader election is to increase density of the contracting regions in order to reduce the risk of disconnection. This strategy is statistical and localistic, and so it works only partially and the CA scales poorly. Performance for $N = 39^2$ and uniform distribution is 0.487.

4.6 Enumeration of Unsolvable Configurations and Performance Upper Bound for One-Dimensional Leader Election

In the dissertation we showed that configurations that are symmetric or loosely-coupled are principally unsolvable for any CA tackling the leader election problem. This work has been published in parts in (Banda et al., 2014).

A configuration $\mathbf{s} \in \Sigma^N$ is *symmetric* iff $\mathbf{s} = \mathbf{q}^m$, where $\mathbf{q} \in \Sigma^l$ for some integers $m > 1$ and $l > 0$. Once a configuration is symmetric, the CA's transition rule ϕ must maintain the configuration symmetry, and therefore no leader can be elected. A loosely-coupled configuration is a configuration, where the distance between active cells is too large ($\geq 2r + 1$ for radius r). By an implication of the leader election definition a loosely-coupled configuration must be a fixed point, and therefore, if number of active cells > 1, no leader can be elected either. We enumerate such unsolvable configurations using basic linear algebra and combinatorics. More precisely, the number of symmetric configurations $|S^a_{N,k}|$ for lattice size N and $k \leq N$ active cells is

$$|S^a_{N,k}| = \sum_{i=1}^{\omega(d)} (-1)^{i+1} \sum_{\substack{J \subseteq \{1,\ldots,\omega(d)\} \\ |J|=i}} \binom{\frac{N}{\prod_{j\in J} r_j}}{\frac{k}{\prod_{j\in J} r_j}} (|\Sigma|-1)^{(N-k)/\prod_{j\in J} r_j}.$$

where $d = \gcd(k, N)$, and $N = \prod_{i=1}^{\omega(N)} p_i^{\alpha_i}$, $k = \prod_{i=1}^{\omega(k)} q_i^{\beta_i}$, and $d = \prod_{i=1}^{\omega(d)} r_i^{\gamma_i}$ are the prime factorizations of N, k, d, respectively.

The number of loosely-coupled configurations $|L_{N,k}|$ for lattice size N and k active cells is

$$|L_{N,k}| = \left(\binom{k+f-1}{k-1}(2r+1) + \binom{k+f-1}{k} \right) (|\Sigma|-1)^{N-k}.$$

where r is a radius, and $f = N - k(2r+1)$ the number of free cells, which do not belong to the neighborhood of any active cell.

By combining these two formulas we derive an universal upper bound on performance. The proportion of unsolvable configurations decreases dramatically with the system size for uniform distribution (Figure 7(a)).

4.7 Enumeration of Unsolvable Configurations and Performance Upper Bound for Two-Dimensional Leader Election

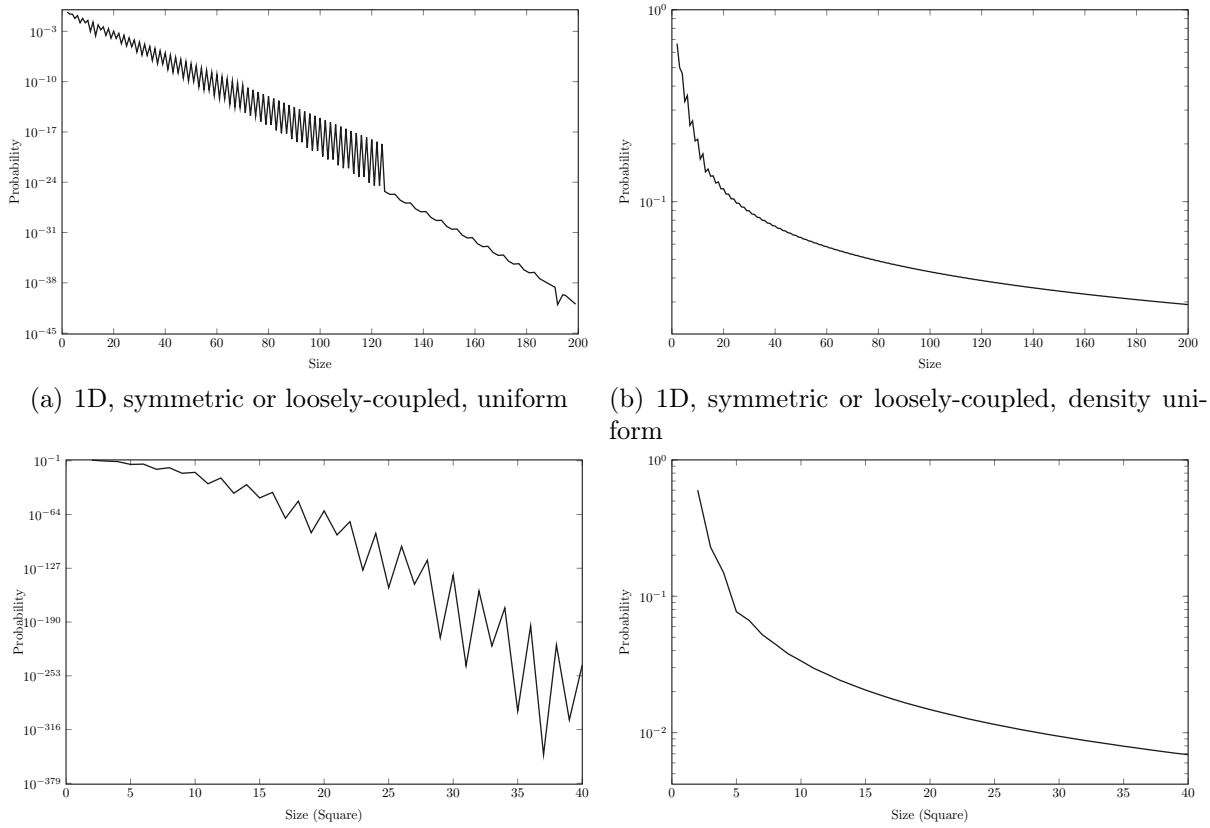
Similarly to the one-dimensional case, symmetric and loosely-coupled two-dimensional configurations are principally unsolvable for any CA tackling the leader election problem. We generalized the concept of configuration symmetry by employing vector projections and group theory. In fact, any projection of states on the toroidal lattice by a non-zero vector v defines a configuration symmetry. The number of symmetric two-dimensional configurations $|S^a_{N,k}|$ for lattice size $N = n^2$ and $k \leq N$ active cells can be calculated as

$$|S^a_{n\times n,k}| = \sum_{\substack{0\leq l_1 \leq (r_1+1) \\ \vdots \\ 0\leq l_{\omega(d)} \leq (r_{\omega(d)}+1)}} (-1)^{1+\sum_{i=1}^{\omega(n)} l_i} \left(\prod_{i=1}^{\omega(d)} \binom{r_i+1}{l_i} \right) \binom{\frac{n^2}{\prod r_i}}{\frac{k}{\prod r_i}} (|\Sigma|-1)^{\frac{n^2-k}{\prod r_i}},$$

where $d = \gcd(k, n)$, $\prod r_i = \prod_{i=1}^{\omega(d)} r_i^{\min(l_i,2)}$, and $n = \prod_{i=1}^{\omega(n)} p_i^{\alpha_i}$, $k = \prod_{i=1}^{\omega(k)} q_i^{\beta_i}$, and $d = \prod_{i=1}^{\omega(d)} r_i^{\gamma_i}$ are the prime factorizations of n, k, d, respectively.

As opposed to the one-dimensional case, enumeration of two-dimensional loosely-coupled configurations is substantially more difficult. In fact, the problem of enumerating loosely-coupled configurations is a generalized problem of non-attacking kings, for which exist no universal closed formula for arbitrary k and n. As a matter of fact, rather than deriving an explicit formula for $|L_{n\times n,k}|$, we numerically obtained approximate values by statistical sampling. More specifically, we randomly drew k positions for active cells, and then

checked whether the configuration is loosely coupled. After we repeated this process $30,000$ times, we multiplied the fraction of loosely-coupled configurations with the total number of placements of k active cells $\binom{n^2}{k}$. Finally, we multiplied the result by $(|\Sigma| - 1)^{n^2-k}$, the number of ways to place the remaining $n^2 - k$ inactive cells on lattice.



(a) 1D, symmetric or loosely-coupled, uniform

(b) 1D, symmetric or loosely-coupled, density uniform

(c) 2D, symmetric or loosely-coupled, uniform

(d) 2D, symmetric or loosely-coupled, density uniform

Figure 7: Probability of selecting unsolvable binary configurations for one- and two-dimensional symmetric or loosely-coupled configurations using uniform and density-uniform distributions.

References

Banda, P. (2008). Cellular automata evolution and leader election in a symmetric ring. In Bielikova, M., editor, *Proceedings in Informatics and Information Technologies, IIT.SRC 2008 - Student Research Conference*, pages 303–310. STU.

Banda, P. (2009a). Computational mechanics, evolution of cellular automata and leader

election problem (in Slovak). In Kelemen, J., Kvasnicka, V., and Rybar, J., editors, *Cognition and Artificial Life '09*, pages 19–28. SLU Opava.

Banda, P. (2009b). Upper bound performance of cellular automata for the leader election problem in arbitrary ring. In Bielikova, M., editor, *Proceedings in Informatics and Information Technologies, IIT.SRC 2009 - Student Research Conference*, pages 158–165. STU.

Banda, P. (2010). Binary cellular automata approach to anonymous, self-stabilizing leader election on rings. Rigorous thesis, Faculty of Mathematics, Physics and Informatics, Comenius University.

Banda, P. (2011). Cellular automata evolution of leader election. In Kampis, G., Karsai, I., and Szathmáry, E., editors, *Advances in Artificial Life. Darwin Meets von Neumann*, volume 5778 of *Lecture Notes in Computer Science*, pages 310–317. Springer Berlin / Heidelberg.

Banda, P., Caugmann, J., and Pospichal, J. (2014). Configuration symmetry and performance upper bound of one-dimensional cellular automata for the leader election problem. *International Journal of Unconventional Computing (Accepted)*.

Büsing, L., Schrauwen, B., and Legenstein, R. (2010). Connectivity, dynamics, and memory in reservoir computing with binary and analog neurons. *Neural computation*, 22(5):1272–1311.

Cenek, M. (2011). *Information Processing in Two-Dimensional Cellular Automata*. PhD thesis, Portland State University.

Codd, E. F. (1968). *Cellular automata*. Academic Press.

Crutchfield, J. P. and Hanson, J. E. (1993). Turbulent pattern bases for cellular automata. *Physica D*, 69(3/4):279–301.

Crutchfield, J. P., Mitchell, M., and Das, R. (2003). Evolutionary design of collective computation in cellular automata. In *Evolutionary dynamics: Exploring the interplay of selection, accident, neutrality, and function*, pages 361–411. Oxford.

Das, R., Crutchfield, J. P., Mitchell, M., and Hanson, J. E. (1995). Evolving globally synchronized cellular automata. In Eshelman, L., editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 336–343, San Francisco, CA. Morgan Kaufmann.

Derrida, B. and Pomeau, Y. (1986). Random networks of automata: a simple annealed approximation. *EPL (Europhysics Letters)*, 1(2):45.

Frederickson, G. N. and Lynch, N. A. (1987). Electing a leader in a synchronous ring. *JACM*, 34(1):98–115.

Ganguly, N., Maji, P., Dhar, S., Sikdar, K. B., and Chaudhuri, P. P. (2002). Evolving cellular automata as pattern classifier. In *ACRI '01: Proceedings of the 5th International Conference on Cellular Automata for Research and Industry*, pages 56–68, London, UK. Springer-Verlag.

Hordijk, W. (2000). *Dynamics, emergent computation, and evolution in cellular automata.* PhD thesis, University of New Mexico, Albuquerque, NM.

Itai, A. and Rodeh, M. (1994). Symmetry breaking in distributed network. In *Proceeding of the 22nd Annual IEEE Symp. of Foundations of Computer Science (FOCS)*, pages 245–260. IEEE Press.

Koza, J. R. (1992). *Genetic programming: on the programming of computers by means of natural selection.* The MIT Press, Cambridge, MA.

Langton, C. G. (1990). Computation at the edge of chaos: Phase transitions and emergent computation. *Physica D: Nonlinear Phenomena*, 42(1):12–37.

Marques-Pita, M. and Rocha, L. M. (2008). Conceptual structure in cellular automata-the density classification task. In *ALIFE*, pages 390–397.

Mitchell, M. (1996). *An introduction to genetic algorithms.* MIT Press.

Nagpal, R. (2003). A catalog of biologically-inspired primitives for engineering self-organization. In *Engineering Self-Organising Systems*, volume 2977 of *Lecture Notes in Computer Science*, pages 53–62. Springer.

Neumann, J. V. (1966). *Theory of self-reproducing automata - edited and completed by Burks.* Illinois Press.

Packard, N. H. (1988). *Adaptation toward the edge of chaos.* University of Illinois at Urbana-Champaign, Center for Complex Systems Research.

Rohlf, T., Gulbahce, N., and Teuscher, C. (2007). Damage spreading and criticality in finite random dynamical networks. *Physical review letters*, 99(24):248701.

Shmulevich, I. and Kauffman, S. A. (2004). Activities and sensitivities in boolean network models. *Physical Review Letters*, 93(4):048701.

Sipper, M. (1995). Studying artificial life using a simple, general cellular model. *Artificial Life Journal*, (2):1–35.

Smith, A. R. I. (1971). Two-dimensional formal languages and pattern recognition by cellular automata. In *Proceeding of the 22nd Annual IEEE Symp. of Foundations of Computer Science (FOCS)*, pages 144–152. IEEE Press.

Wolfram, S. (1986). *Theory and Application of Cellular Automata.* Singapore: World Scientific.

The Author's Relevant Publications

Banda, P. (2006). Cellular automata evolution, computational mechanics and global coordination (in Slovak). Master's thesis, Faculty of Mathematics, Physics and Informatics, Comenius University.

Banda, P. (2008). Cellular automata evolution and leader election in a symmetric ring. In Bielikova, M., editor, *Proceedings in Informatics and Information Technologies, IIT.SRC 2008 - Student Research Conference*, pages 303–310. STU.

Banda, P. (2009a). Computational mechanics, evolution of cellular automata and leader election problem (in Slovak). In Kelemen, J., Kvasnicka, V., and Rybar, J., editors, *Cognition and Artificial Life '09*, pages 19–28. SLU Opava.

Banda, P. (2009b). Upper bound performance of cellular automata for the leader election problem in arbitrary ring. In Bielikova, M., editor, *Proceedings in Informatics and Information Technologies, IIT.SRC 2009 - Student Research Conference*, pages 158–165. STU.

Banda, P. (2010a). Binary cellular automata approach to anonymous, self-stabilizing leader election on rings. Rigorous thesis, Faculty of Mathematics, Physics and Informatics, Comenius University.

Banda, P. (2010b). Symmetry breaking and limits of self-stabilization in cellular automata for the leader election problem (in Slovak). In Kelemen, J. and Kvasnicka, V., editors, *Cognition and Artificial Life '10*, pages 37–42. SLU Opava.

Banda, P. (2011). Cellular automata evolution of leader election. In Kampis, G., Karsai, I., and Szathmáry, E., editors, *Advances in Artificial Life. Darwin Meets von Neumann*, volume 5778 of *Lecture Notes in Computer Science*, pages 310–317. Springer Berlin / Heidelberg.

Banda, P., Caugmann, J., and Pospichal, J. (2014). Configuration symmetry and performance upper bound of one-dimensional cellular automata for the leader election problem. *International Journal of Unconventional Computing (Accepted).*