DOI: http://dx.doi.org/10.7551/978-0-262-31709-2-ch050

Cellular Automata Coevolution of Update Functions and Topologies: A Tradeoff between Accuracy and Speed

Christian Darabos[1†], Craig O. Mackenzie[1], Marco Tomassini[2], Mario Giacobini[3], and Jason H. Moore[1]

[1]The Geisel School of Medicine at Dartmouth College, Hanover, NH 03755, USA
[2]Information Systems Department, Faculty of Business and Economics, University of Lausanne, Switzerland
[3]Computational Epidemiology Group at Department of Veterinary Sciences and
Complex Systems Unit at Molecular Biotechnology Center, University of Torino, Italy
[†]Christian.Darabos@dartmouth.edu

Abstract

Biological organisms have the ability to develop novel phenotypes in response to environmental changes. When several traits are evolved simultaneously or as a result of one another, we talk of coevolution. Cellular Automata (CAs) have been successfully used to artificially evolve problem specific update functions. The resulting CAs are, however, much slower and more sensitive to perturbations than those with an evolved underlying topology and fixed uniform update rule. Unfortunately, these are not nearly as accurate, and suffer from scaling up the total number of cells. We propose a hybrid paradigm that simultaneously coevolves the supporting network and the update functions of CAs. The resulting systems combine the higher fitness and performance of the update evolution and the robustness properties and speed of the topology evolution CAs. Moreover, these systems seem to perform better as the size of the CA scales up, where as single-feature evolution systems are negatively impacted. Coevolution in CAs is an interesting tradeoff between the two single trait evolutions.

Introduction

In biology, coevolution refers to the concurrent or sequential mutation in organisms driven by changes in a related biological object (Yip et al., 2008). Coevolution can occur at many different levels of biology: from populations and species, to adaptation of a predator to its (adapted) prey, to the evolution of a parasite/symbiont and its host, down to related mutations in amino-acids and proteins within a single organism. All members taking part in coevolution exert mutual selective pressure on each other, influencing the evolutionary process of the other. When taking place within a single biological entity, coevolution is beneficial to the entire organism. Multiple traits coevolve in order to produce individuals with a higher degree of "fitness" with respect to their environment. Cellular automata (CAs) have been used for years as a proxies for the simulation of rudimentary organisms and biological processes. In a prominent study, Mitchell *et al.* have successfully used genetic algorithms (GAs) to artificially evolve a single feature, the update function shared by all cells, of small radius one-dimensional CAs (Mitchell et al., 1993) to perform a prototypical task. However, tasks must not be only prototypical. CAs using GA evolved functions have proven able to undertake complex tasks, applied, for instance to identifying combinations of genetic markers associated with clinical endpoints (Moore and Hahn, 2002b,a). More recently, we have conducted a study evolving a different property of CAs, the underlying network topology of CAs, with comparable success (Tomassini et al., 2005). The resulting evolved topologies are general graphs, which exhibit attributes of social network. A pioneering work by Sipper and Ruppin studied the coevolution of cellular machines (non-uniform CAs), now commonly know as (random) Boolean networks, which are non-uniform variants of the CA in which each cell has its own update function, instead of a single function shared by all cells (Sipper and Ruppin, 1997).

In this work, we propose a new framework for CA evolution consisting of the simultaneous evolution of the single update function shared by all cells (uniform CAs) and the supporting network topology of the CAs. We hypothesize that evolutionary algorithms (EAs) will generate individuals with a high capacity to solve the task at hand, and develop network topologies supporting speed, robustness, and resilience to transient failures better than that of strictly regular CAs (Tomassini et al., 2007). We compare the fitness-based performance of entire populations of the two single-feature artificial evolutions against a population of CAs simultaneously evolving both the update function and the layout of the cellular connections. Additionally, we analyze the scalability of both the existing frameworks and of the new one, as the performance of CAs with a relatively small fixed number of neighbors generally decreases with a larger number of cells. Finally, we conduct a statistical profiling of the artificially evolved network topologies in order to study the emergent properties of CAs with a higher performance.

Background

CAs and the Density Classification Task

CAs are dynamical, usually deterministic, discrete, abstract models used to simulate and study distributed computation. A standard CA consists of a finite number N of identical cells. Each cell can take one of a finite number of states s, here, the two Boolean states $s \in \{0, 1\}$. Each cell has a local knowledge of the state of a fixed number of n neighboring cells, including itself. The state of each cell is updated synchronously in discrete time steps, according to a local, identical update function or rule (these terms will be used interchangeably throughout this work) shared by all cells. Cells are usually arranged on a d-dimensional grid, where typically $d \in \{1, 2, 3\}$. In this study, we focus on one-dimensional, or linear CA, in which cells are arranged on a regular ring structure, connecting to a radius of r cells on each side. Thus the neighborhood size is $n = 2r + 1$. At any given *discrete time step* t, the set of all states s_i^t of all cells is called the *configuration* of the CA such that $c^t = (s_0^t, s_1^t, \ldots, s_{N-1}^t)$, thus CAs with N nodes have exactly 2^N possible configurations. Starting from an initial configuration (IC or c^0) at time $t = 0$, the CA will travel across transient configurations before reaching a previously visited state of the system. After at most 2^N time steps, the CA will start cycling deterministically through a subset of configurations.

The Density Classification Task The density classification task is a prototypical distributed computational task for CAs and is defined as follows. Let ρ_0 be the fraction of 1s in the IC (i.e. time step 0). The CA's task is to determine whether ρ_0 is greater than or less than $1/2$. If $\rho_0 > 1/2$, then the goal is to have the CA converge to a fixed-point configuration of all 1s; otherwise to a fixed-point configuration of all 0s, after a number of time steps with the order of N, where the CA has a odd size N to eliminate the case $\rho_0 = 0.5$. This computation is trivial for a computer having a central control and will provide the answer in $O(N)$ time. However, it is nontrivial for one-dimensional CA, with a small radius, since such a CA can only transfer information at finite speed relying on local information exclusively, while density is a global property of the configuration of states (Mitchell et al., 1993).

Graph Properties

A CA can be seen as a mathematical object known as a graph, where each cell resides on a vertex, and edges between vertices represent two neighboring cells. Therefore, formal definitions of graph theory do apply to CAs. For ease of reference, we summarize concepts used in subsequent sections particular to this work (see (Newman, 2010) for complete reference). In this work, a graph G, or network, consists of a set of v vertices V, and a set of e undirected, unweighted edges E. The degree k of a vertex is the number of edges connected to it. Thus the average degree $\bar{k}$ of G is the average of the degree over V. A path between vertices u and v is defined as the sequence of unique edges traversed when going from u to v. Its length is the number of edges in the sequence. The average path length (APL) of G is the average length of the *shortest* path between all pairs of vertices. The clustering coefficient C_j of a vertex j is defined as the ratio between the E_j edges that actually exist between the k_j neighbors of j and the number of possible edges between these nodes: $C_j = 2E_j/k_j(k_j - 1)$. The clustering coefficient (CC) of a graph is defined as the average C_j across all vertices. The degree distribution $P(k)$ of a graph G is a function that gives the probability that a randomly selected vertex has k edges incident to it.

Artificial Evolution of CAs

It has been shown that the density task cannot be solved perfectly by a uniform, two-state CA with finite radius $r < (N-1)/2$ (Land and Belew, 1995). Despite the lack of a perfect solution, it is desirable to find one or more solutions that achieve the highest degree of performance possible.

Evolving the Update Function In general, it is extremely difficult to infer the local CA function that will give rise to the desired global computation due to possible nonlinearities and large-scale collective effects. On the other hand, exhaustive evaluation of all 2^{2^n} possible functions is limited to small radii $r \in \{1, 2\}$ by the computational cost. As first proposed by Mitchell *et al.* (Mitchell et al., 1994, 1993) for uniform CAs and by Sipper for nonuniform ones (Sipper, 1997; Sipper and Ruppin, 1997), EAs have proven to be a very effective heuristic to search in the colossal solution space of all update functions. Additionally, EAs have been applied to the discovery of efficient update functions for complex CA systems, such as CAs with multidimensional structure (Breukelaar and Bäck, 2005), as opposed to the linear, monodimensional nature of Mitchell's CAs, and those of interest in the present work.

Evolving the Topology In order to modify the underlying topology of the network supporting the cells, we will consider an extension of the concept of CAs. Therefore, cells can be connected in any way, provided that multiple edges are disallowed. Sipper and Ruppin have already examined the influence of different connectivity patterns on the density task. They studied the coevolution of network architectures and CA rules, resulting in non-uniform, high-performance networks (Sipper and Ruppin, 1997). More recently, Watts also moved away from regular structures, and hand constructed general uniform CAs for the density task (Watts, 1999). Because of the heterogeneous degree distribution of his networks, he had to reduce the update function to its simplest, and most flexible form, the majority rule (MR), to accommodate cells with varying neighborhood sizes. At each time step, each cell will assume the state of the majority of its neighbors in the graph. Watts built many networks with performance values exceeding that of CAs with evolved rules on regular lattices with similar average degree. Network structures yielding a good performance tend to have "long" links, creating shortcuts between distant cells that somewhat compensate for the lack of information transmission of the regular lattice case.

In both these studies, the authors correctly recognized that reducing the average cell to cell distance, i.e. the APL, has a positive effect on the performance of the CA. Inspired by their work, we have explored the effect of artificially evolving the underlying topology of CAs with uniform MR starting with a population of regular lattice and one of random topology CAs (Tomassini et al., 2004, 2005, 2007). Evolved networks resulting from either initial populations tend to converge in the "small-world" region of the spectrum between regular structures and random ones (Watts and Strogatz, 1998). Indeed, these evolved topologies exhibited long reaching shortcuts across the network, thus significantly shortening the APL. Their CC is higher than that expected of equivalent random networks. Finally, their degree distribution was slightly skewed to the right, showing the emergence of a few higher-than-average connected hub cells, which is another property of social networks.

Methods

In order to compare and contrast the performance of existing paradigms to that of our coevolution of topology and rules, we have implemented the two single-feature evolutions according to the original framework specifications. However, in order to make them comparable to each other and to our work, some parameters needed to be harmonized. Unless otherwise specified, parameters are identical across all simulation sets.

Evolutionary Algorithm for CAs

In this work, we use an EA with the aim of evolving CAs for the density task. We assume the reader is familiar with the concept of artificial evolution, evolutionary computation, evolutionary algorithms, and genetic algorithms (Holland, 1975; Bäck, 1996; Mitchell, 1996). For all 100 experimental replicates, we generate an initial unstructured population of size $P = 100$ individuals (i.e. individuals are not spatially limited in their interactions). The definition of individuals varies according to the framework used: update function, network topology, or both simultaneously (see below). We explore the scalability of the systems by studying CAs of size $N \in \{99, 199, 299\}$. Regardless of the size N and of the framework used, the initial population at generation $g = 0$ is made of P uniform regular (i.e. ring) CAs with a radius $r = 3$, thus the neighborhood size of each cell is $n = 7$, including itself. The radius changes as the EA progresses when evolving the topology and with coevolution. Evolution is ended when the entire population has reached an optimal fitness, or after a maximum of g_m generations. We find experimentally that $g_m = 100$ is enough for the populations to reach a *fitness plateau* where improvement becomes marginal or null.

The fitness function of the EA used to evaluate the "quality" of a CA individual in the population consists of experimentally evaluating the ability of the CA to solve the density task over a sample of 100 ICs with uniformly distributed densities $\rho \in [0, 1]$. CAs are allowed a maximum number of $2N$ time steps to converge. If two consecutive configurations c^t and c^{t+1} are identical, the CA is stopped, as it has reached a *single configuration* attractor, and all consecutive configurations will remain identical. The fitness is defined as the fraction of instances (i.e. ICs) for which the CA produces the correct fixed point, given the known density of the IC. At each generation a different set of ICs is generated for each individual.

The next generation is obtained by repeating P standard *binary tournament* selection over the entire population. We describe the mutation methods below, as they depend on the evolution strategy. However, the mutation rates are selected quantitatively to yield the best results. The concept of *recombination* of network individuals is ill defined, and cumbersome to implement. However, we have used standard single-point recombination in frameworks that evolve the update function with a probability $p_c = 0.25$. At the end of each evolutionary process, we select the *elite population* (EP) by selecting the CAs that fall in the 95^{th} percentile of *performance* (i.e. all individuals with a performance that is within the absolute best). For performance evaluation, the entire population is exposed to 1,000 instances of the most difficult problems possible, that is, on ICs with $\rho \approx 0.5$. Note the difference between *fitness* (100 uniform ICs) and *performace* (1'000 difficult ICs). In order to obtain statistically sound results, we replicate all experiments 100 times, and record the average fitness, the best fitness at each generation, and network statistics, such as the degree distribution, the APL and the average CC at the end of the evolutionary process.

Evolution of the Update Function (UFE) When we evolve the update function of uniform CAs with fixed topology, each individual of the initial population is assigned a different random update function in the form of a Boolean lookup table. This table is shared by all cells in the CA and its size is 2^n, where $n = 7$ is the size of the neighborhood. At each time step, cells synchronously update their binary state according to the state of their neighborhood. The ordering of the neighborhood is predetermined, thus making the system fully deterministic. At every generation, selected parent individuals will produce mutated offspring. The mutation will impact the offspring's lookup table and is susceptible to change the binary value of each position in the table with a probability $p_m = 5 \times 10^{-3}$. The offspring will replace its weaker parent, if it has a better fitness than either. This approach is similar, although not entirely identical to that of Mitchell *et al.* (Mitchell et al., 1994, 1993).

Evolution of the Topology (TE) When evolving the topology, each regular uniform CA of the initial population is assigned the same update function, that is, the MR. At each time step, cells will update their state to reflect the majority of its neighborhood. As mentioned previously, the MR has been proven incapable of solving the density task when applied to a regular CA. In order to increase its fitness, we allow each CA to modify the structure of its supporting network. At every generation, selected individuals produce a mutated offspring that has a high probability of replacing its parent in the next generation if it shows a higher degree of fitness. Each cell will see its neighborhood mutated with a probability $p_m = 5 \times 10^{-3}$. If a cell is mutated, it will lose one of its neighbor or gain a random one with equal probability (Tomassini et al., 2005). This operator prevents drastic changes in the average degree of the CAs, which would giving an advantage to those with a higher connectivity (i.e. more edges). We disallow mutations that would produce self-loops or duplicate edges. In the case of a tie in an even sized neighborhood, the state will be drawn at random.

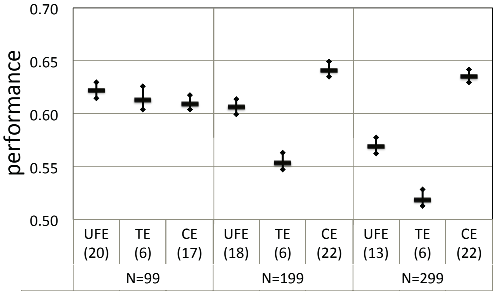
Coevolution (CE) This is the new framework we propose, allowing simultaneous evolution of the update function and the underlying topology of the network. Similar to the evolution of the update function only, coevolution starts with a population of uniform regular CAs, each with a randomly generated lookup table. As in the precedent frameworks, selected parents at each generation will produce mutated offspring, which may replace its parents in the subsequent generation. Mutations will now affect first the topology of the CA, by mutating edges with a probability $p_t = 0.001$, and then the lookup table, with probability $p_r = 0.003$. In this case, the size of the lookup table might need to be adapted to the growing sizes of the neighborhoods. Indeed, every time the size n_{max} of the largest neighborhood increases by 1, where $n'_{max} = n_{max} + 1$, the table size doubles, growing from $2^{n_{max}}$ to $2^{n_{max}+1} = 2^{n'_{max}}$. The new half of the lookup table is completed with randomly selected values with equal probability. This ensures that, when the new neighbor is *off*, the target cell behaves just as it did before the new neighbor was added. When it is *on*, the target cell has a whole new behavior.

Experimental Results

In order to compare results with previously proposed evolutionary CAs frameworks, we conducted parallel simulations for all possible combinations of frameworks (rule-only, topology-only, and coevolution) and sizes $N \in \{99, 199, 299\}$. At each generation of the artificial evolution, we record the average fitness of each population, and the fitness of the best individual. After the last generation has been produced, we segregate the EP as described above. We compute standard graph statistics on the CA networks of EP to shed some light on the mathematical properties that set the evolutionary frameworks apart from one another.

Performance and Fitness

The ultimate goal of evolving CA for computation, regardless of the framework, is to obtain individuals that excel at solving both the average case and the "worst case scenario" of the task at hand. In fact, a single individual is enough as long as its performance is satisfying the task's criteria of quality and speed. Figure 1 shows the results of the performance evaluation of the EP on difficult problems. Each column represents a combination of frameworks and CA sizes. Performance, just like fitness, is on a scale from $[0, 1]$ representing the fraction of correctly classified IC in at most $2N$ time steps. We show the consistency of the EP results by showing the absolute best performance (upper mark), the average performance (horizontal line), and the lowest performance (lower mark). The number in parentheses is the size of the EP for each case, and gives an ideas of how rich the solution space is in "good individuals".



	UFE (20)	TE (6)	CE (17)	UFE (18)	TE (6)	CE (22)	UFE (13)	TE (6)	CE (22)
	N=99			N=199			N=299		
min.	0.615	0.604	0.604	0.599	0.547	0.635	0.562	0.513	0.629
max.	0.630	0.626	0.617	0.614	0.563	0.650	0.577	0.529	0.642
▬ avg.	0.622	0.613	0.609	0.607	0.553	0.641	0.570	0.519	0.635

Figure 1: Performance of Best Evolved CAs. Performance of all three frameworks: update function/rule evolution (UFE), topology evolution (TE), and coevolution (CE) of individuals $N \in 99, 199, 299$ in the elite population (EP). Performance is measured over 1,000 ICs with $\rho \approx 0.5$. Average number of individuals in each run is in parentheses. In each column, the horizontal bar represents the average performance, the bottom mark represents the lowest performance, and the top mark shows the best performance. Results are averaged over 100 independent simulations.

In the case of smaller CAs, results across all frameworks are virtually undistinguishable, yet consistently above the 0.6 mark. In any case considered, the deviation between the maximum and the minimum fitness is minute (~ 0.02). The clear distinction comes with the scaling of the systems to larger CAs. The performance of TE CAs drops significantly as the system grows. Similarly, the UFE suffers a decrease of performance as the system grows, though not as much as TE CAs. On the other hand, coevolving CE systems see their performance remain stable or marginally improve as

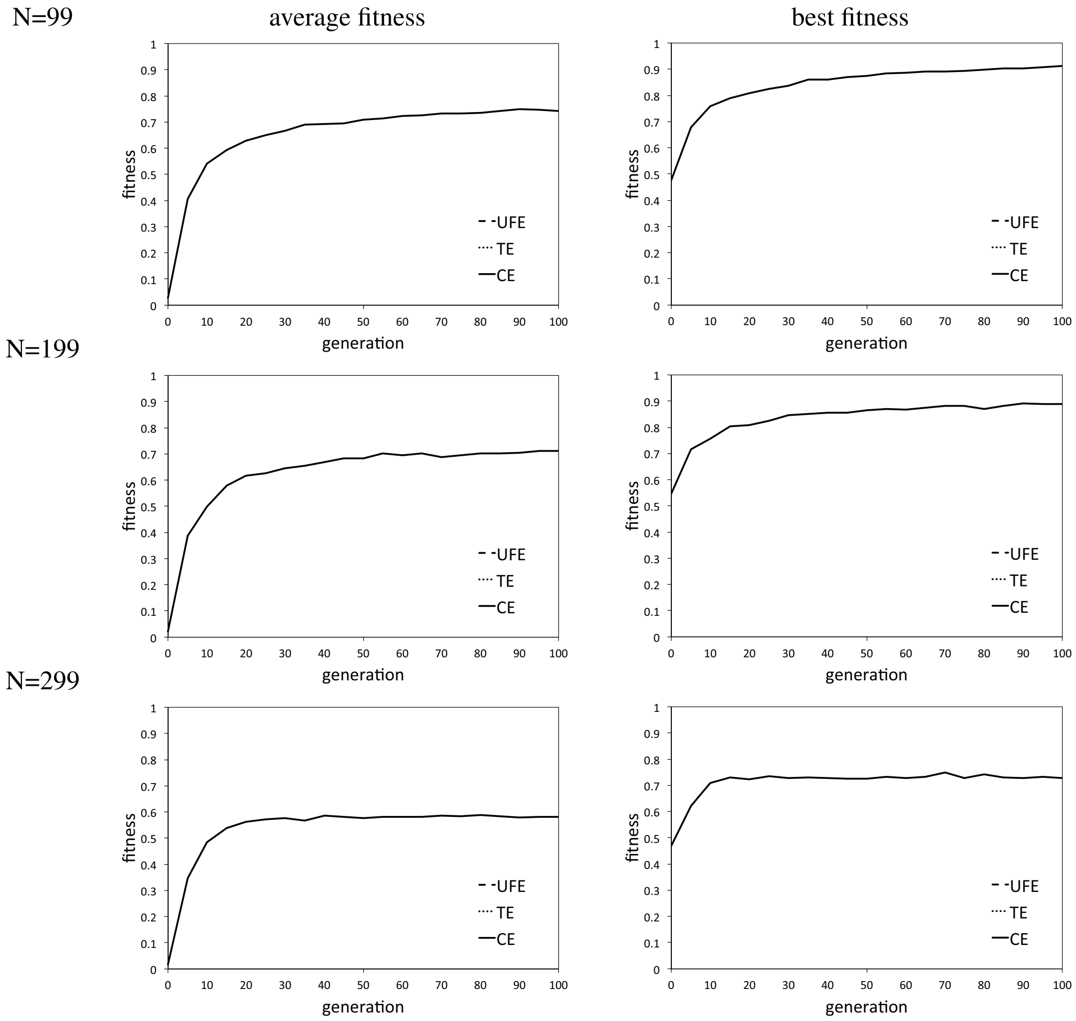


Figure 2: Fitness progression of Evolutionary CAs. Average fitness over the course of 100 generations for populations of evolutionary CAs (left column) and fitness of the best individual (right column). Systems of size $N = 99$ (top row), $N = 199$ (middle row), and $N = 299$ (bottom row). Each panel shows results of all three frameworks: rule evolution (UFE, dashed line), topology evolution (TE, dotted line), and coevolution (CE, continuous line). Results are averaged over 100 independent simulations.

the CAs get larger. Interestingly, UFE and CE seem to consistently evolve a larger number of good individuals (Figure 1, numbers in parenthesis) than TE, therefore the solution space of TE is likely to be the poorest in EP CAs. This difference can be explained by the complexity and overall size of the actual solution space. In the case of UFE, there is at most 2^{2^n} different update possible that constitutes the entire space. In the case of TE, however, the solution space is much larger, made of all possible topological network structures of N vertices. Surprisingly, the EP size of CE is also large, despite of the much larger solution space, made of all possible topologies coupled with all possible update function.

To understand the results presented above, we analyze the time progression of our evolutionary CAs over the 100 generations. We trace the development of UFE, TE, and CE populations' average fitnesses in the left-hand column of Figure 2, whereas the right-hand side panels show the curves for the fitness of the best individual in the CA population, with the highest fitness value. Additionally, we show the scaling of these systems, which will also help us appreciate the shape and richness (or lack thereof) of the different solution spaces. When comparing the different combinations of evolutionary framework and sizes, we notice that the general trends are similar across all panels of a column in Figure 2. CAs under UFE and CE have the steepest learning curve. CE yield the second steepest curve, reaching a lower, CA size dependent, fitness plateau almost at the same time as UFE. The difference of average fitness between CE and UFE increases notably with the scaling in size N. However, we notice that this difference is less pronounced in the best fitness curves in the right panels. This might explain the results in performance presented in Figure 1. The trace of TE has a different shape, with a slow start, turning into a steep slope, only reaching it's fitness plateau consistently significantly higher than that of UFE or CE. Surprisingly, only the average population of TE seems impacted by the scaling. Additionally, in the average fitness panels, we note that the UFE and CE curves start at a fitness close to 0, where as TE starts with a lead of about 0.5, we note that the final fitness of TE is notably above that of UFE and CE, which seems to contradict the performance results above. In fact, we see the importance of the probability ρ of the ICs. Indeed, topology-only evolutionary CAs perform on average better than the other two on easier problems.

Best fitness results in the right-hand side panels of Figure 2 are consistent with those of the average fitness. Again, TE CAs reach a near optimum fitness, despite a slower start. The best individuals under UFE and CE preform closely, although UFE become clearly superior to CE with the increasing size of the system. Which again proves that the solution landscape of these EAs is not necessarily favorable to those rich in good individuals, as even randomly initialized individuals are capable of high fitness. Moreover, the fact that topology-only evolution yield a best fitness near 0.5 at the first generation also agrees with Mitchell's finding that uniform topologies with the majority rule cannot solve the density task better than a random system. The scaling of the CA sizes impacts the fitness of the coevolution framework the most, both in average and maximum fitness. We even witness a slight drop in best fitness for $N = 199$, a drop more prominent in $N = 299$, where after a slight, or no increase at all, the best fitness plateaus at a value lower than that of the maximum reached earlier in the evolution. We can only explain the fact that CE is detrimental for larger systems' fitness by the competing evolution of the two traits simultaneously. Nevertheless, the artificial evolution is beneficial in terms of producing CAs capable of consistently solving the task with any density ρ, especially those closest to $\rho \approx 0.5$ and is clearest in the larger CAs. The results of UFE and TE are much more sensitive to the difficulty of the problem, where the fitness drops in both cases below the performance of CE in larger systems under difficult problems.

In summary, we note that the fitness in CE's may be less associated with performance than in the other models. This may indicate that high fitness does not (linearly) translate into high performance. It would appear that the TE's are the most flexible at solving the majority problem. While they may not be the best on the really hard problems, they show great adaptability to the a wide range of initial conditions.

Properties of Evolved Topologies

EA's only goal is to optimize the performance of the CAs. However, looking beyond the fitness and performance, we are interested in studying the properties emerging from the evolved topologies, and how they differ when obtained solely by topology evolution, and when the network's evolution is combined with adaptations of the update function. Therefore, we analyze the graph and statistical properties of the supporting network structure of CAs before the evolutionary process starts (i.e. a regular ring structure) and after the last generation of the EA. Figure 3 offers a visual representation of sample CA structures before (A) and after the evolution (B) and (C). Figure 3(A) shows the regular topology of UFE CAs (at all times) and of all CAs before the specific evolutionary process takes place. All vertices have a degree $k = 4$, and are thus all the same size. In the example graphs post-evolution (B) and (C), we see the emergence of cells of higher and lower degrees. The average degree of the networks is, however, maintained at all times $\bar{k} \approx 4$ by the edge mutation within the graph (see Methods' section). The vertices' sizes are proportional to their degree, and ordered according to that criteria. At first glance, TE gives rise to a greater diversity in the vertices' degrees, whereas CE CAs seem more homogeneous in their degree distribution. The degree distributions of both frameworks, topology evolution and coevolution, and sizes of CAs are depicted in Figure 4. The values for the degree distributions are averaged over the elite population of the 100 replicates. From the degree distributions in Figure 4, we can see the EA has shifted the peak of the CE from all nodes having a degree $k = 7$ to a majority of $k = 8$. Although there is some spread in the degrees, the function is narrowly centered around its peak, with little deviation, and no extreme values. TE has, on the other hand, facilitated a larger heterogeneity in the degree distribution, with a significantly wider bell shaped curve, and no clear peak at a single value of the degree k. TE CA networks has therefore more extreme values, where the minimum degree of a TE network is smaller than that of a CE network of the same size: $k_{min}^{TE} < k_{min}^{CE}$. The opposite is true of maximum degrees: $k_{max}^{TE} > k_{max}^{CE}$. The number of cells does not appear to have a marked influence on the shape of the degree distribution of emerging networks. The size of the CA only affects the magnitude of the function, not the shape. In Table 1, we report two essential statisti-

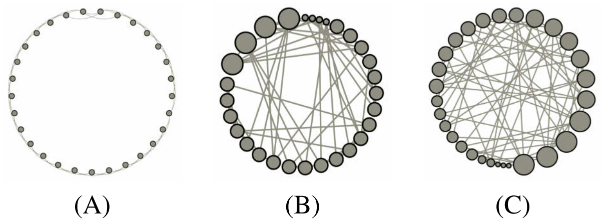
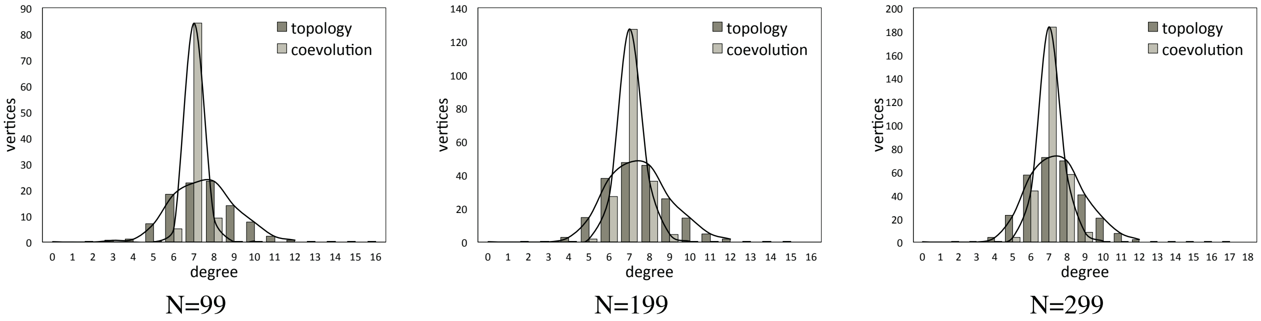
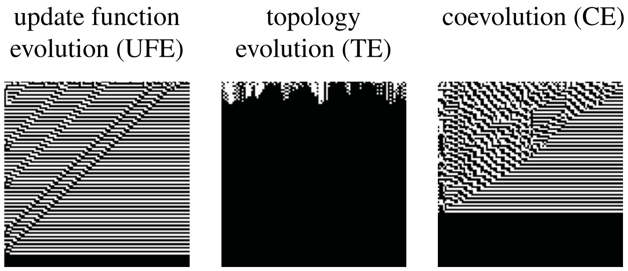


Figure 3: CA Topologies. Instances of a regular CA's initial network topology (A), and the CA topologies resulting from topology-only evolution (B) and coevolution (C). In the bottom row, the size of the vertices is proportional to the size of the neighborhood of the cell (i.e. degree of the cell). CA size $N = 29$ for ease of reading.



N=99 N=199 N=299

Figure 4: Degree Distributions of Evolved Topologies. CA topologies of sizes $N \in 99, 199, 299$. The degree distributions (bar plots) show the number of vertices (Y-axis) having a given degree (X-axis). Each panel contains the results for CAs with evolution of the topology-only (dark grey) and coevolution (light grey). Results are averaged over 100 independent simulations. Continuous lines are only meant as a guide for the eye to the trend of the degree distribution probability functions.



ATS	UFE	TE	CE
N=99	90.37	12.41	76.15
N=199	141.08	18.16	127.07
N=299	440.20	23.32	132.21

Figure 5: Density Classification Task performed by Evolved CAs. Examples of single time step (vertical axis) evolutions of evolved CAs solving the density classification task starting from an arbitrary IC with $\rho \approx 0.5$ (top row of the horizontal axis in each panel) for all three CA evolution frameworks. The table below the panels show the average number of time steps (ATS) necessary to successfully solve the density task over 1,000 ICs.

cal properties to the study of the CA topologies, the average path length and the average clustering coefficient after the evolution. Formal definitions of these two metrics can be found in the Background section.

The degree distributions in Figure 4 and the values of APL and CC in Table 1 suggest that the network structures emerging from artificial evolution share properties with technological, social, and other "real-world" networks. These architectures generally show greater resilience to perturbations than regular structures (Newman, 2010; Watts, 1999). Moreover, the even degree distributions, with a few hub-like cells, the short APL and higher CC of graphs after coevolution are all properties placing them even closer to "real-life" on the spectrum of all graphs.

Speed of Convergence

If the quality of the results is key to the success of a CA, the speed at which the CA will converge to a solution is a non-negligible factor. In our previous studies (Tomassini et al., 2005, 2007), we show that the diffusion of information is immensely facilitated by the emergence of shortcuts across the networks, thus the shortening of the APL. Due to limited space, we present in Figure 5 only three examples representative of the ability of evolved-topology CAs to converge to a solution faster than regular CA, regardless of the excellence of their evolved rules.

In Figure 5, we show the progression of evolved CAs through the configuration space converging towards the correct IC density value. Mitchell *et al.* have analyzed the emerging patterns visible in this type of figure (Mitchell et al., 1993). In our results, TE CAs are consistently orders of magnitude faster than CE (over 500%), which are, in turn, significantly faster than UFE ($10 - 300\%$) to reach a steady configuration.

Conclusions & Future Work

Cellular Automata are, despite their apparent simplicity, powerful models for distributed computations, provided that an adequate update function can be found. In the present work, we demonstrate once again the ability of EA to develop highly performant rules to solve a prototypical CA task, and remaining unaffected by the scaling of the CA size. The solution space is, in this case, rich in good solutions, making the EA capable of finding large numbers of functions that perform well. From a distributed computation perspective, the regular topology of standard CAs can be a weakness, as these structures are more susceptible to fail under transient perturbation. One alternative is to evolve the topology, and leave the majority rule as a constant update function. This TE paradigm evolves topologies showing properties of resilient systems, and the resulting CAs are a great deal faster than UFE. Unfortunately, their

Table 1: Evolved CAs Network Statistics. Average path length (APL) and average clustering coefficient (CC) for evolved CA networks in the EP for each evolutionary framework, and CA sizes $N \in 99, 199, 299$.

	N=99			N=199			N=299		
	rules	topology	coevolution	rules	topology	coevolution	rules	topology	coevolution
APL	8.59	2.53	4.410	16.91	2.97	5.11	25.25	3.23	4.68
CC	0.71	0.21	0.67	0.71	0.21	0.67	0.71	0.22	0.59

performance is inversely proportional to the density, and to the size of the CA. In this work, we propose a framework that simultaneously evolves the update function and topology underlying. We have developed a novel update function implementation that integrates with the changing topology. The CAs resulting from CE demonstrate performance levels comparable to, and scale better than UFE, and are considerably faster. Moreover, the performance of CE systems remains constant (even slightly increases) as we scale up the size of the CAs. On the contrary, this scaling affects adversely the performance of both UFE and TE. Finally, they exhibit properties possibly making them even more robust network systems than TE CAs. We are planning on implementing a complementary study for structured populations, as they have been shown to increase the EA performance. In addition, we will conduct a thorough analysis of the resilience and fault tolerance of all evolved CAs (Tomassini et al., 2005).

Acknowledgments

The authors gratefully acknowledge Britney E. Graham at Dartmouth College for her help in editing this article. CD and JHM gratefully acknowledge the financial support of the NIH (grants AI59694, LM010098 and LM009012). MG gratefully acknowledges the financial support of the local research founding program of the University of Torino.

References

Bäck, T. (1996). *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, New York.

Breukelaar, R. and Bäck, T. (2005). Using a genetic algorithm to evolve behavior in multi dimensional cellular automata: emergence of behavior. In *GECCO*, pages 107–114.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, Michigan.

Land, M. and Belew, R. K. (1995). No perfect two-state cellular automata for density classification exists. *Physical Review Letters*, 74(25):5148–5150.

Mitchell, M. (1996). *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA.

Mitchell, M., Crutchfield, J. P., and Hraber, P. T. (1994). Evolving cellular automata to perform computations: mechanisms and impediments. *Physica D*, 75:361–391.

Mitchell, M., Hraber, P. T., and Crutchfield, J. P. (1993). Revisiting the edge of chaos: evolving cellular automata to perform computations. *Complex Systems*, 7:89–130.

Moore, J. H. and Hahn, L. W. (2002a). Cellular automata and genetic algorithms for parallel problem solving in human genetics. In *Proceedings of the 7th International Conference on Parallel Problem Solving from Nature*, PPSN VII, pages 821–830, London, UK, UK. Springer-Verlag.

Moore, J. H. and Hahn, L. W. (2002b). A cellular automata approach to detecting interactions among single-nucleotide polymorphisms in complex multifactorial diseases. In *Pacific Symposium on Biocomputing 2002*, pages 53–64. http://www.odysci.com/article/1010112988060408.

Newman, M. (2010). *Networks: An Introduction*. Oxford University Press, Inc., New York, NY, USA.

Sipper, M. (1997). *Evolution of Parallel Cellular Machines: The Cellular Programming Approach*, volume 1194 of *Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, New York.

Sipper, M. and Ruppin, E. (1997). Co-evolving architectures for cellular machines. *Physica D*, 99:428–441.

Tomassini, M., Giacobini, M., and Darabos, C. (2004). Evolution of small-world networks of automata for computation. In et al., X. Y., editor, *Parallel Problem Solving from Nature - PPSN VIII*, volume 3242 of *Lecture Notes in Computer Science*, pages 672–681. Springer Verlag, Berlin.

Tomassini, M., Giacobini, M., and Darabos, C. (2005). Evolution and dynamics of small-world cellular automata. *Complex Systems*, 15:261–284.

Tomassini, M., Giacobini, M., and Darabos, C. (2007). Performance and robustness of cellular automata computation on irregular networks. *Advances in Complex Systems*, 10:85–110.

Watts, D. J. (1999). *Small Worlds: The Dynamics of Networks Between Order and Randomness*. Princeton University Press, Princeton, NJ.

Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of ”small-world” networks. *Nature*, 393:440–442.

Yip, K. Y., Patel, P., Kim, P. M., Engelman, D. M., McDermott, D., and Gerstein, M. (2008). An integrated system for studying residue coevolution in proteins. *Bioinformatics*, 24(2):290–292.