Dynamic control and information processing in the Belousov–Zhabotinsky reaction using a coevolutionary algorithm

Rita Toth, Christopher Stone, Andrew Adamatzky, Ben de Lacy Costello, and Larry Bull

Citation: J. Chem. Phys. **129**, 184708 (2008); doi: 10.1063/1.2932252
View online: http://dx.doi.org/10.1063/1.2932252
View Table of Contents: http://jcp.aip.org/resource/1/JCPSA6/v129/i18
Published by the American Institute of Physics.

Dynamic control and information processing in the Belousov–Zhabotinsky reaction using a coevolutionary algorithm

Rita Toth,[a] Christopher Stone,[b] Andrew Adamatzky,[c] Ben de Lacy Costello,[d] and Larry Bull[e]
Unconventional Computing Group, University of the West of England, Bristol BS16 1QY, United Kingdom

(Received 7 November 2007; accepted 1 May 2008; published online 14 November 2008)

We propose that the behavior of nonlinear media can be controlled dynamically through coevolutionary systems. In this study, a light-sensitive subexcitable Belousov–Zhabotinsky reaction is controlled using a heterogeneous cellular automaton. A checkerboard image comprising of varying light intensity cells is projected onto the surface of a catalyst-loaded gel resulting in rich spatiotemporal chemical wave behavior. The coevolved cellular automaton is shown to be able to either increase or decrease chemical activity through dynamic control of the light intensity within each cell in both simulated and real chemical systems. The approach is then extended to construct a number of simple logical functions. © *2008 American Institute of Physics*.
[DOI: 10.1063/1.2932252]

INTRODUCTION

The theory of evolution was introduced by Charles Darwin, namely, that all living species evolve over the course of generations through a natural selection process.[1] The complex mixture of biochemical reactions that build up a living system is the result of evolution over billions of years.[2] These reactions are interacting with each other nonlinearly producing and maintaining a semiclosed system: An organism, which consists of cells.[3,4] Living cells communicate with each other and with their environment by passing, receiving, and processing information and these processes are dynamically controlled. How the dynamical behavior of (bio)chemical systems is controlled and how they process information is an important question.

There is also growing interest in research into the development of "nonlinear computers."[5] The aim is to harness the as yet only partially understood intricate dynamics of nonlinear media to perform complex "computations" more effectively than with traditional architectures and to further the understanding of how such systems function. Previous theoretical and experimental studies have shown that reaction-diffusion chemical systems are capable of information processing.[6–10] In our previous work (see overview in Ref. 10), we demonstrated that nonlinear chemical systems are capable of implementing various kinds of computational procedures. Experimental prototypes of reaction-diffusion processors have been used to solve a wide range of specialized computational problems, including image processing,[11,12] path planning,[13,14] robot navigation,[15] computational geometry,[16] "chemical diode,"[17] counting,[18] and implementing memory.[19,20] In addition to these applications, logic gates were constructed in excitable chemical systems[9,21–24] and in bistable systems.[25] Various logic gates as well as simple computational devices based on pattern recognition were studied in coupled continuously fed stirred tank reactors (CSTRs).[26–31] Okamoto *et al.*[32] have considered certain enzymatic reactions as basic biochemical switching devices.

In this work, we produce networks of nonlinear media—reaction-diffusion systems—to achieve a user-defined computation in a way that allows direct control of the media. We use the spatially distributed light-sensitive Belousov–Zhabotinsky (BZ) reaction which supports traveling reaction-diffusion waves and patterns. Exploiting the photoinhibitory property of the reaction, the chemical activity (amount of excitation on the gel) can be controlled by the applied light intensity, namely, can be decreased by illuminating the gel with high light intensity and vice versa. In this way, a BZ network is created via light and controlled using cooperative coevolutionary computing to design heterogeneous automata networks. We adapt the system described by Wang *et al.*[33] and explore its computational potential based on the movement and control of wave fragments. In our experiments, a heterogeneous automata network controls the light intensity in the cells of a checkerboard image projected onto the surface of the light-sensitive catalyst-loaded gel. Initially a certain number of wave fragments are created on the gel and the automata network is shown to be able to either increase or decrease the number of wave fragments through dynamic control of the light intensity within each cell in both simulated and real chemical systems. Building upon these results, we increase the complexity of the task and we design a simple scheme to create a number of two-input Boolean logic gates: AND, NAND, and XOR both in simulation and real chemical experiments.

EVOLUTIONARY ALGORITHMS

An Evolutionary Algorithm[34] (EA) is a search technique inspired by biological evolution that mimics the process of

[a]Electronic mail: Rita.Toth@hotmail.com.
[b]Electronic mail: christopher3.stone@uwe.ac.uk.
[c]Electronic mail: andrew.adamatzky@uwe.ac.uk.
[d]Electronic mail: Ben.Delacycostello@uwe.ac.uk.
[e]Electronic mail: larry.bull@uwe.ac.uk.

natural selection. EAs employ a population of individuals, each coding as a genome a possible solution to a target problem. By testing each of these individuals against the target problem, the fitness of that individual with respect to the problem is obtained. Promising individuals (i.e., those with high fitness) are then selected from the population and evolved by means of recombination and/or mutation. Recombination involves combining elements from promising individuals to create new, hopefully better, individuals. We expect that a high degree of spatial heterogeneity will typically be required in the present system, rendering the use of recombination somewhat superfluous. We therefore do not use recombination in the present work. Instead, mutation is the sole source of variation used in our EA and it occurs by changing randomly chosen genes to other random values with a fixed low probability, allowing alternative possible solutions to be explored. In this way, a new generation of offspring individuals is created from the original parent population, which replaces the previous population. This cycle is repeated for a specified number of generations or until an individual with the desired fitness is found.

EAs are being used increasingly in the design and analysis of complex systems (e.g., Ref. 35). Example applications include data mining, time series analysis, scheduling, process control, robotics, and electronic circuit design. EAs are also used in organic chemistry for laser pulse shape design to split complex molecules into simpler molecules[36] and in the pharmaceutical industry for product-based design of combinatorial libraries.[37] Such techniques can be used for the design of computational resources in a way that offers substantial promise for application in nonlinear media computing since the algorithms are almost independent of the medium in which the computation occurs. This is important in order to achieve effective nonlinear media computing since an EA does not need to directly manipulate the material to facilitate learning and the task itself can be defined in an unsupervised manner. In contrast, most traditional learning algorithms use techniques that require detailed knowledge of and control over the computing substrate involved. Indeed, Harding and Miller[38] have recently described the use of an EA to design a computational system using liquid crystals and Tour *et al.*[39] constructed a number of logic gates from molecular switches using an EA.

HETEROGENEOUS AUTOMATA NETWORK

We use an automata network to control the chemical system. The network has a cellular-automaton topology, i.e., finite automata are arranged in a (2D) lattice with aperiodic boundary conditions (an edge cell has five neighbors, a corner cell has three neighbors, and all other cells have eight neighbors each). Use of an automata network with such a 2D topology is a natural choice given the spatiotemporal dynamics of the BZ reaction. Each automaton updates its state depending on its own state and the states of its neighbors. States are updated in parallel and in discrete time. In cellular automata (CA), all cells have the same state transition function (rule), whereas in this automata network each cell/automaton can have its own state transition function. In this work, the transition function of every cell is evolved by an EA. Rules are initially random and change over the course of an experiment as the EA searches the space of possible solutions.

Such automata networks are sometimes called heterogeneous CA, but the notion is misleading and contradictory to the exact sense of CA. However, we will be using this terminology in our paper to keep consistency with previous works on evolving CA.

Sipper[40] has presented a nonuniform, or heterogeneous, approach to evolving CA. Each cell of a one-dimensional or 2D CA is also viewed as an EA population member, mating only with its lattice neighbors and receiving an individual fitness. However, the reliance upon each cell having access to its own fitness means it is not applicable in the majority of chemical computing scenarios we envisage. Instead, fitness is based on emergent global phenomena in our approach. Thus, following Kauffman,[41] we use a simple coevolutionary approach wherein each cell of a 2D CA controller is developed via a simple EA. Coevolution is a particular form of evolution whereby the fitness of an individual in a population undergoing evolution depends on the fitness of other members of that population. Due to the use of a single global fitness measure, cells do not evolve in isolation and fitness is influenced by the state of all cells in the grid.

For a given experiment, a random set of CA rules is created for a 2D array of size 10×10, i.e., 100 cells. The rule for each cell is represented as a gene in the EA's genome, which at any one time takes one of the discrete light intensity values used in the experiment. As previously mentioned, the grid edges are not connected (i.e., the grid is planar and does not form a toroid) and the neighborhood size of each cell is of radius 1; cells consider neighborhoods of varying size depending on their spatial position, varying from three in the corners, to five for the other edge cells, and eight everywhere else. In the model, each of the 100 cells consists of 400 (20×20) simulation points for the reaction. The reaction is thus simulated numerically by a lattice of size 200×200 points, which is divided into the 10×10 grid in the same way as for the chemical experiment.

CHEMICAL MODEL

Features of the chemical system are simulated using a two-variable Oregonator model modified to account for photochemistry:[42,43]

$$\frac{\partial u}{\partial t} = \frac{1}{\varepsilon}\left(u - u^2 - (f\nu + \Phi)\frac{u-q}{u+q}\right) + D_u \nabla^2 u$$

$$\frac{\partial \nu}{\partial t} = u - \nu.$$

The variables u and v represent the instantaneous local concentrations of the bromous acid autocatalyst and the oxidized form of the catalyst, $HBrO_2$ and tris (bipyridyl) Ru (III), respectively, scaled to dimensionless quantities. The rate of the photoinduced bromide production is designated by Φ, which also denotes the excitability of the system. Low simulated light intensities facilitate excitation while high intensi-

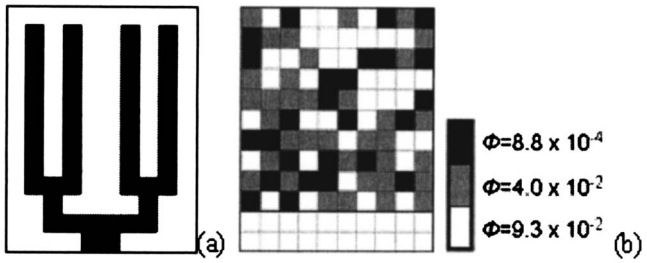

FIG. 1. (a) Initiation pattern; (b) a typical example of a coevolved light pattern.

ties result in the production of bromide ions that inhibits the process, experimentally verified by Kádár *et al.*[42] The system was integrated using the Euler method with a five-node Laplacian operator, time step $\Delta t=0.001$ and grid point spacing $\Delta x=0.62$. The diffusion coefficient, D_u, of species u was unity, while that of species v was set to zero as the catalyst is immobilized in the gel. The kinetic parameters were set to $\varepsilon=0.11$, $f=1.1$, and $q=0.0002$. The medium is oscillatory in the dark which made it possible to initiate waves in a cell by setting its simulated light intensity to zero. At different Φ values, the medium is excitable, subexcitable, or nonexcitable.

CONTROL PROCESS

Waves were initiated by setting Φ to zero for a small area under and just outside the bottom center of the grid. These waves were channeled into the grid and broken up into 12 fragments by choosing an appropriate light pattern, as shown in Fig. 1(a). The black area represents the excitable medium while the white area is nonexcitable. After initiation, three light levels were used: one is sufficiently high to inhibit the reaction, one is at the subexcitable threshold such that excitation just manages to propagate, and the other low enough to fully enable it. The modeled chemical system was run for 600 iterations of the simulator. This value was chosen to produce network dynamics similar to those obtained in experiment over 10 s of real time.

A color image was produced by mapping the level of oxidized catalyst at each simulation point into an RGB value. These color images were designed to mimic those obtained from a digital camera during experiments and thus allow direct comparison between simulation and experimental results.

Image processing of the color image was necessary to determine chemical activity. This was done by differencing successive images on a pixel by pixel basis to create a black and white thresholded image. Each pixel in the black and white image was set to white (corresponding to excitation) if the intensity of the red or blue channels in successive color images differed by more than 5 out of 256 pixels (1.95%). Pixels at locations not meeting this criterion were set to black. An outline of the grid was superimposed on the black and white images to aid visual analysis of the results. Though this may appear to be a somewhat circuitous methodology, for reasons of consistency, we wanted to use the same techniques for extraction and processing of results for both modeling and experiment.

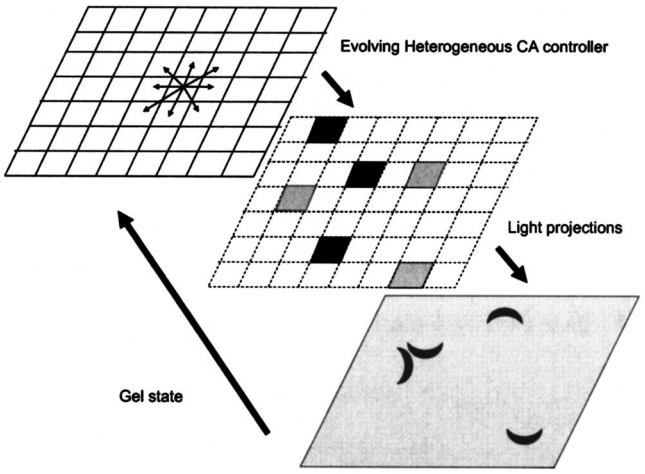

FIG. 2. Relationship between the CA controller, applied grid pattern, and (simulated) chemical system comprising one process control cycle.

The black and white images were then processed to produce a 100 bit description of the grid for the CA. In this description, each bit corresponds to a cell and it is set to true if the average level of activity within the given cell is greater than a predetermined threshold of 10%. Here, activity is computed for each cell as the fraction of white pixels in that cell. This binary description represents a high-level depiction of activity in the BZ network and is used as input to the CA. 1 cycle of the CA is performed whereby each cell of the CA considers its own state and that of its neighbors (obtained from the binary state description) to determine the light level to be used for that grid cell in the next time step. Each grid cell may be illuminated with one of three possible light levels. The CA returns a 100-digit ternary action string, each digit of which indicates whether high ($\Phi=0.093\,023$), subexcitable threshold ($\Phi=0.04$), or low ($\Phi=0.000\,876$) intensity light should be projected onto the given cell. The progression of the (simulated) chemical system, image analysis of its state, and operation of the CA to determine the set of new light levels comprises one control cycle of the process. Figure 2 provides a simplified view of one control cycle and a typical light pattern generated by the CA controller is shown in Fig. 1(b).

Another 600 iterations are then simulated with those light levels projected, etc., until 25 control cycles have passed. After 25 control cycles, the fitness of the emergent behavior is calculated. As previously mentioned, the EA used in this work employs a single global fitness measure. The nature of the tasks undertaken means that it is not possible to decompose solutions obtained by the EA and apportion fitness to their constituent parts. Instead, a global fitness is determined according to how well the task has been performed and this fitness is assigned to the genome for each CA cell.

After fitness has been assigned, some proportion of the CA genes are randomly chosen and mutated. Mutation is the variation operator used by the EA to modify a CA cell's transition rule to allow the exploration of alternative light levels for the grid state randomly selected. During mutation, the transition value for one of the possible states considered

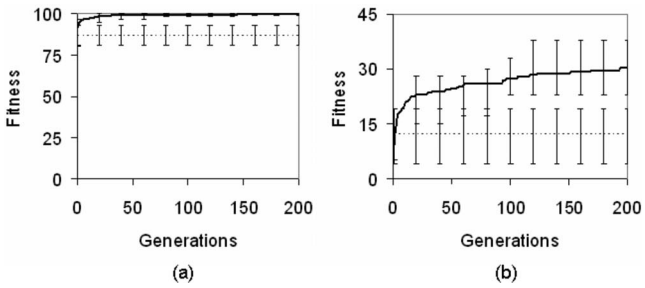

FIG. 3. The typical fitness over time for both the (a) inhibition and (b) excitation tasks with the model. Solid lines, mutation rate 1000; dashed lines, random controller.

by a CA cell is changed to one of the other possible light intensities. For a CA cell with eight neighbors, there are 2^9 possible grid state to light level transitions, each of which is a potential mutation site. After mutation, a generation of the EA is complete and the simulation is reset and repeated as described.

The EA keeps track of which CA states are visited since mutation. On the next fitness evaluation (at the end of a further 25 control cycles), mutations in states that were not visited are discarded on the grounds that they have not contributed to the global fitness value and are thus untested. We also performed control experiments with a modified version of the EA to determine the performance of a random CA controller. This algorithm ignored the fitness of mutants and retained all mutations except those from unvisited states.

NUMERICAL RESULTS

We begin by considering two tasks. In the first task, the CA controller must decrease the amount of excitation on the grid. In the second task, the goal is to increase the overall amount of excitation, i.e., the CA controller must create new fragments.

Figure 3(a) shows the fitness of our coevolutionary approach averaged over ten runs for the inhibition task wherein fitness is calculated as the number of cells in the grid that have an activity level less than the 10% threshold. Note that for this task fitness increases as grid activity decreases. As can be seen, the amount of excitation decreases during learning. Figure 4 shows the snapshots of the spatiotemporal behavior of a typical solution produced.

Figure 3(b) shows the fitness of our coevolutionary approach averaged over ten runs for the excitation task wherein fitness is calculated as the number of cells in the grid that have an activity level greater than or equal to the 10% threshold. As can be seen, the amount of excitation increases during learning. Figure 4 shows the snapshots of the spatiotemporal behavior of a typical solution produced. If the task were to have been tackled by a person with prior knowledge of the BZ reaction, the number of fragments would be increased simply via the predominant projection of the lowest light level. However, it appears from observation that the number of fragments and thus the total excitation level is also increased via the application of appropriate high and low light intensity cells placed by the CA so as to maneuver and split existing fragments. However, the exact mechanism of this process is yet to be fully understood.

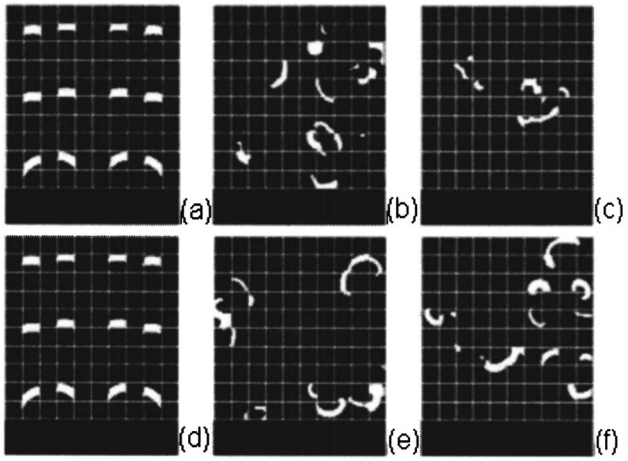

FIG. 4. Example solutions to the modeled [(a)–(c)] inhibition and [(d)–(f)] excitation task after [(a) and (d)] initiation and coevolved at generations of [(b) and (e)] 30 (b, e) and [(c) and (f)] 200. State on last of the 25 control cycles shown.

EXPERIMENTAL IMPLEMENTATION

Given the success of our approach using the simulated chemical reaction, we have attempted to implement the same two tasks using a real chemical system constructed using the following methodology.

Sodium bromate, sodium bromide, malonic acid, sulfuric acid, tris(bipyridyl) ruthenium (II) chloride, and 27% sodium silicate solution stabilized in 4.9*M* sodium hydroxide were purchased from Aldrich (UK) and used as received unless stated otherwise.

To create the gels a stock solution of sodium silicate was prepared by mixing 222 ml of the purchased sodium silicate solution with 57 ml of 2*M* sulfuric acid and 187 ml of de-ionized water.[33] $Ru(bpy)_3SO_4$ was recrystallized from the chloride salt with sulfuric acid.[44] Gels were prepared by mixing 2.5 ml of the acidified silicate solution with 0.6 ml of 0.025*M* $Ru(bpy)_3SO_4$ and 0.65 ml of 1.0*M* sulfuric acid solution. Using capillary action, portions of this solution were quickly transferred into a custom-designed 25 cm long, 0.3 mm deep Perspex mould covered with microscope slides. The solutions were left for 3 h to permit complete gellation. After gellation the adherence to the Perspex mould is negligible leaving a thin gel layer on the glass slide. After 3 h, the slides were carefully removed from the mould and the gels on the slides were washed in de-ionized water at least five times to remove by-products. The gels were 26×26 mm^2, with a wet thickness of approximately 300 μm. The gels were stored under water and rinsed just before use.

The catalyst-free reaction mixture was freshly prepared in a 30 ml CSTR, which involved the *in situ* synthesis of stoichiometric bromomalonic acid from malonic acid and bromine generated from the partial reduction of sodium bromate. This CSTR in turn continuously fed a thermostated open reactor with fresh catalyst-free BZ solution in order to maintain a nonequilibrium state. The final compositions of

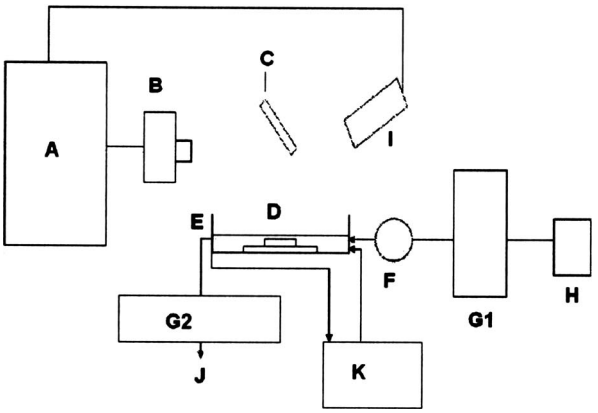

FIG. 5. A block diagram of the experimental setup: A, computer; B, projector, C, mirror; D, microscope slide with the catalyst-loaded gel; E, thermostated Petri dish; F, CSTR; G1 and G2, pumps; H, stock solutions; I, camera; J, effluent flow; K, thermostated water bath.

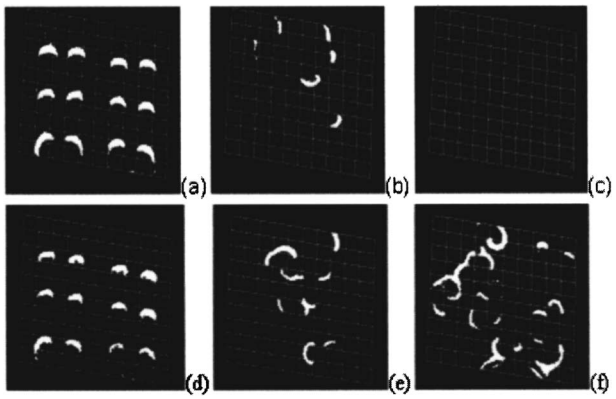

FIG. 6. An example solution to the [(a)–(c)] inhibition and [(d)–(f)] excitation task in chemical experiment after [(a) and (d)] initiation and coevolved at generations of [(b) and (e)] 30 and [(c) and (f)] 200. State on the last of the 25 control cycles is shown.

the catalyst-free reaction solution in the reactor were 0.42*M* sodium bromate, 0.19*M* malonic acid, 0.64*M* sulfuric acid, and 0.11*M* bromide. The residence time was 30 min.

An InFocus model LP820 projector was used to illuminate the computer-controlled image. Images were captured using a Lumenera Infinity2 USB 2.0 scientific digital camera. The open reactor was surrounded by a water jacket thermostated at 22 °C. Peristaltic pumps were used to pump the reaction solution into the reactor and remove the effluent. A schematic representation of the experimental setup is shown in Fig. 5. The spatially distributed excitable field on the surface of the gel was achieved by the projection of a 10×10 cell checkerboard grid pattern generated using a computer. The checkerboard image comprised of cells with three possible intensity levels of 0.035, 1.6, and 3.5 mW cm^{-2}, representing excitable, the subexcitable threshold, and nonexcitable domains, respectively.

The checkerboard grid pattern was projected onto the catalyst-loaded gel through a 455 nm narrow bandpass interference filter, 100/100 mm focal length lens pair, and mirror assembly. The size of the projected grid was approximately 20 mm^2. Every 10 seconds, the checkerboard pattern was replaced with a uniform gray level of 3.5 mW cm^{-2} for 10 ms during which time an image of the BZ fragments on the gel was captured. The purpose of removing the grid pattern during this period was to allow activity on the gel to be more visible to the camera and assist in subsequent image processing of chemical activity.

Captured images were cropped to the grid location and processed to identify activity in the same manner as for the model (see above).

EXPERIMENTAL RESULTS

Figure 6 shows the examples of the spatiotemporal dynamics exhibited by the real chemical system. In both cases, the behavior observed was qualitatively similar to that observed during numerical simulation. Moreover, our coevolutionary approach is able to control the chemical system to achieve the desired goal to a similar degree of accuracy, i.e., fitness, as was seen in the simulations, as shown in Figs. 7(a) and 7(b) [compare with Figs. 3(a) and 3(b) over the same period]. We were able to run a maximum of 40 generations (which required about 6 h) in real chemical experiments because after that time the excitability of the system changed due to the "desensitization" effect of high intensity light.[45]

As before, if the excitation task were to have been tackled by a person with prior knowledge of the BZ reaction, the number of fragments would be increased simply via the predominant projection of the lowest light level. However, it again appears from experimental observation that the number of fragments and thus the total excitation level is increased

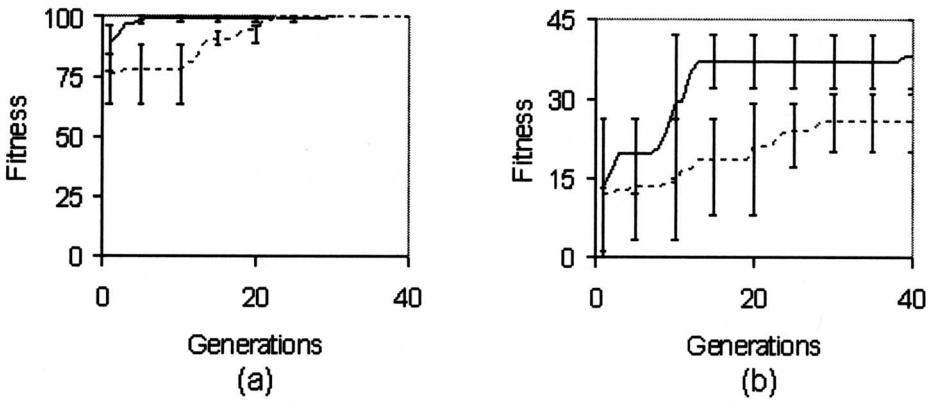

FIG. 7. The typical fitness over time for both the (a) inhibition and (b) excitation tasks on the real chemical system averaged over three runs. Solid lines, mutation rate 1000; dashed lines, random controller.

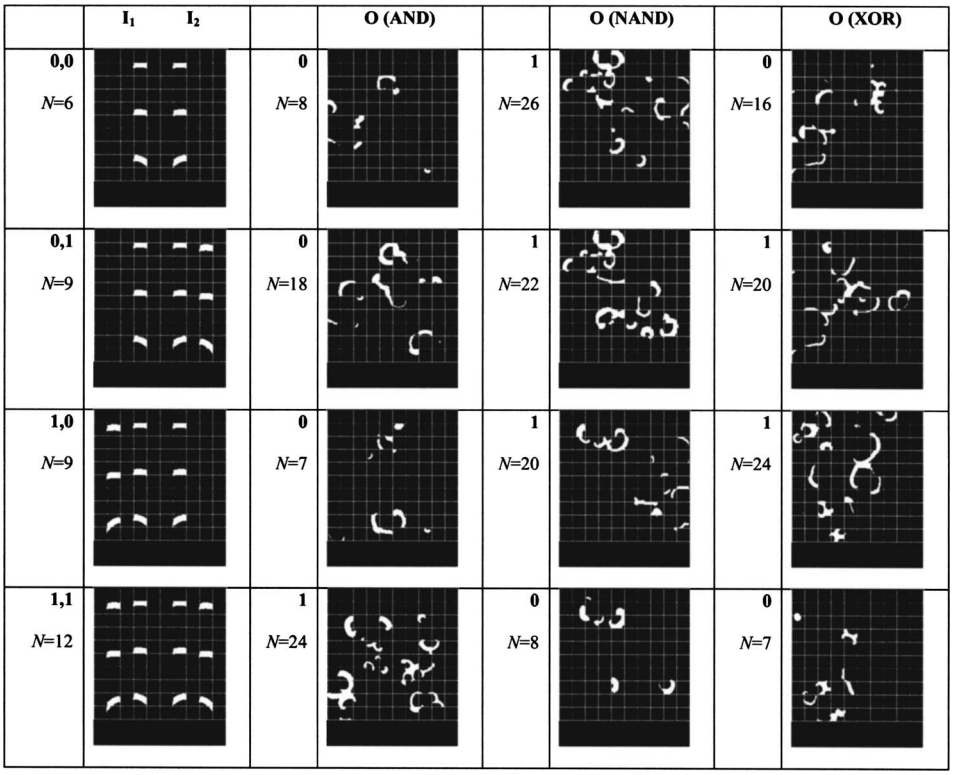

FIG. 8. Typical examples of solutions of AND, NAND, and XOR logic gates in simulation, required active cells: 20, *N*: actual number of active cells. Input states I_1, I_2 for the logic gates are shown on the left and consist of two binary digits, spatially encoded using left and right "initiation trees" [Fig. 1(a)]. Input values of 0 are encoded using a single branch of the relevant tree resulting in three fragments, while binary 1 is encoded using both branches of the tree resulting in six fragments. The EA found a solution in 56 (AND), 364 (NAND), and 1656 (XOR) input presentations.

also via the projection of appropriate high and low light intensity cells placed by the CA so as to maneuver and split existing fragments.

SIMPLE LOGIC GATES

To begin examining the potential for the coevolution of such structures in the continuous, nonlinear 2D media described, we have designed a simple scheme to simulate a number of two-input Boolean logic gates. As before, excitation is fed in at the bottom of the grid into the same branching pattern. To encode logicals "1" and "0" either both branches or just one branch of the two "trees" shown in Fig. 1(a) are allowed to fill with excitation, i.e., the grid is divided into two for the inputs (Fig. 8). The number of active cells in the grid, that is, those with activity at or above the 10% threshold, is used to distinguish between logicals 0 and 1 as the output of the system. That is, since the above results indicate that it is possible to either increase or decrease excitation through the coevolution of CA cell rules, we aim to build upon this by increasing the complexity of the task. For example, in the case of XOR, the controller must learn to keep the number of active cells below the specified level for the 00 and 11 cases but increase the number for the 01 and 10 cases.

NUMERICAL RESULTS

Figure 8 shows the typical examples of each of the three logic gates learned using the simulated chemical system. Here all parameters were as before except that the mutation rate was set either at 4000 or 6000. The required number of active cells was set at 20. That is, 20 or more grid locations must exhibit sufficient excitation to be above the 10% activity threshold for the output of the system to be considered as a logical 1, otherwise an output of logical 0 is assumed. Each of the four possible input combinations is presented in turn, 00–11, and for each input presentation the system is allowed to develop for 25 control cycles in a similar way to the inhibition and excitation tasks. Fitness of the logic gate is evaluated after the complete sequence of four-input presentations. Each correct output scores 1, resulting in a maximum possible fitness of 4 for a correctly functioning gate. Figures 9(a) and 9(b) show the fitness averaged over ten runs for AND and NAND tasks with a mutation rate of 4000, and similar results for XOR are shown in Fig. 9(c) for mutation rate 6000.

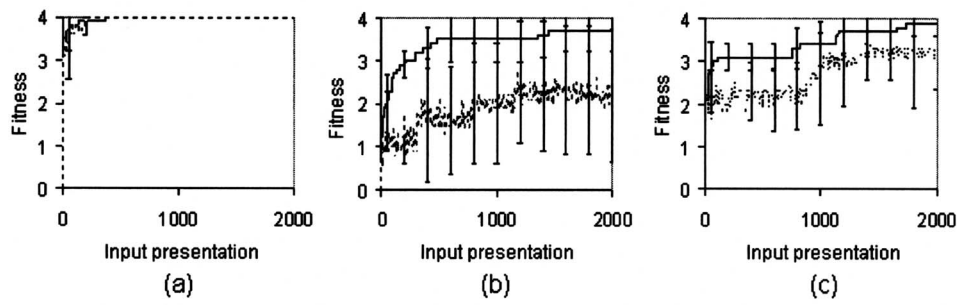

FIG. 9. The average fitness over time for ten runs for the (a) AND gate, (b) NAND gate, and (c) XOR gate on the simulated chemical system. *AND, NAND* with mutation rate of 4000, and XOR with 6000 (solid lines). Dashed lines: random controller (ten runs).

TABLE I. The success rate of ten runs and the minimum, maximum, and average number of input presentations required to find a solution for each gate. An experiment was deemed successful if it found a solution within 2000 input presentations.

Gate	Controller	Mutation rate	Success rate	Min.	Max.	Av.	Std. dev.
AND	Coevolutionary	4000	10/10	8	144	61	45.69
	Random	4000	10/10	4	200	64	71.73
	Coevolutionary	6000	10/10	16	84	49	21.89
	Random	6000	10/10	8	176	66	58.64
NAND	Coevolutionary	4000	7/10	288	>2000	1065	767.21
	Random	4000	4/10	300	>2000	1454	744.49
	Coevolutionary	6000	9/10	24	>2000	847	829.22
	Random	6000	6/10	84	>2000	1247	727.81
XOR	Coevolutionary	4000	9/10	348	>2000	808	510.08
	Random	4000	10/10	20	1080	455	333.68
	Coevolutionary	6000	9/10	32	>2000	1118	574.97
	Random	6000	6/10	212	>2000	1336	635.84

Favorable comparisons to an equivalent random controller are also shown in each case, as before.

Table I shows a more detailed comparison, namely, the results of ten runs for each gate with two different mutation rates (4000 and 6000). All these experiments were also run with random CA controllers. Due to the high computational requirements needed to perform the simulations, a limited number of input presentations were allowed for each experiment and an experiment was considered successful if the controller found a solution within 2000 input presentations. The success rate shows the number of successful runs out of 10. The AND task was so simple that a solution was easily found even with a random controller for both mutation rates. This task was simple because the first three inputs provided activity levels similar to the correct outputs, and only the activity levels provided by the 11 input needed to be changed to generate appropriate output activity, namely, the controller had to increase excitation to get higher than the required number of active cells (that is, those with an activity level greater than or equal to 10%). In contrast, the NAND gate was the most difficult task, because the controller had to achieve the opposite activity levels to those provided by the input states. For 00, 01, and 10 inputs the initial number of fragments were less than the required value so the controller had to increase the excitation to get the correct logical 1 output ($N \geqslant 20$), while for the 11 input the controller had to decrease the excitation to achieve the logical 0 output ($N < 20$). This is seen in the success rates shown in Table I. For this task, the use of a coevolutionary CA controller and an increased mutation rate resulted in a higher success rate and lower average number of input presentations required to find a solution for the NAND gate compared to a random controller and/or lower mutation rate. The XOR task was also hard because the activity levels provided by three of the initial inputs (00, 01, and 10) were the opposite of the desired output activity levels and only the 11 input provided an appropriate direct basis for correct output activity.

We have to mention here that, in the cases where the success rate was less than 10, the averages in Table I, are the lowest possible averages, since 2000 was taken as the number of input presentations required, even though no solution was found in these cases. For this reason, we can only use these data as an indication of the difficulty of the task. Comparing these results with results of the same tasks but with a random controller, we can conclude that the success rates for the coevolutionary CA controller were usually greater than or equal to the success rates of the runs with a random controller. These results indicate the ability of the coevolutionary approach for universal computation since all functions can be constructed by NAND gates.

EXPERIMENTAL RESULTS

Figures 10 and 11 show how similar performance is possible on the real chemical system for each of the three logic functions. In order to produce working XOR and NAND gates from these experiments, it was necessary to use a value of 15 for the required number of active cells due to the relative difficulty of these tasks. All other parameters were the same as those used for numerical simulation.

Because of the limited lifetime of the medium, experiments were once again limited to 40 input presentations. Initial runs with the CA controller produced no correctly functioning logic gates within this time frame (dashed lines on Fig. 11). This is not surprising considering that the average number of input presentations needed to find a solution in numerical simulation was considerably higher than 40 because of the difficulty of the task. We therefore carried out experiments where the genomes of the CAs were seeded with ones evolved during successful simulation runs. CAs seeded in this way (solid lines in Fig. 11) found solutions very quickly—16 or 20 input presentations—compared to numerical simulation. This speedup indicates the similarity of solutions obtained from numerical simulations compared to those needed for the chemical system. With a seeded solution, a few generations of evolution are still needed to adapt to the differences between the two systems, but this is one or two orders of magnitude lower than evolving a solution from a random genome. These results show that the EA is capable of adapting to small changes in its environment and finding a solution very quickly when presented with domain-specific knowledge obtained from modeling.

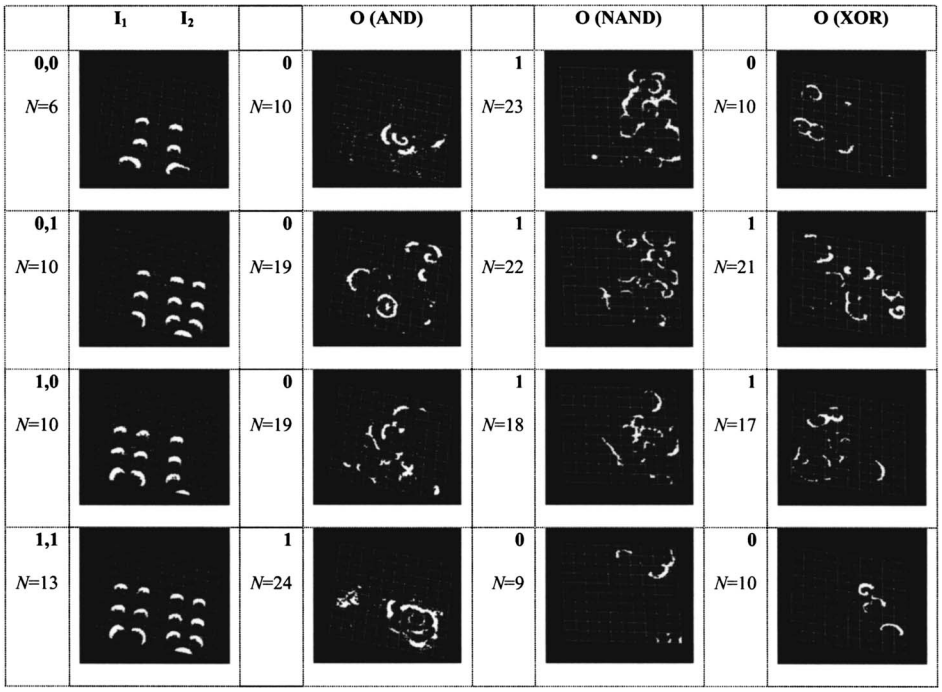

FIG. 10. Typical examples of solutions of AND, NAND, and XOR logic gates in chemical experiment, required number of active cells: 15 (20 for AND), *N*: actual number of active cells. Input states I_1, I_2 for the logic gates are shown on the left and consist of two binary digits, spatially encoded using left and right initiation trees [Fig. 3(a)]. Input values of 0 are encoded using a single branch of the relevant tree resulting in three or four fragments, while binary 1 is encoded using both branches of the tree resulting in six or seven fragments. The EA—seeded with a CA evolved during the simulated runs—found a solution in 16 input presentations in each case.

DISCUSSION AND CONCLUSIONS

Excitable and oscillating chemical systems have previously been used to solve a number of simple computational tasks. However, the experimental design of such systems has typically been nontrivial. In this paper, we have presented initial results from a methodology by which to achieve the complex task of designing such systems—through the use of coevolution. We have shown using both simulated and real systems that it is possible in this way to control dynamically the behavior of a light-sensitive BZ reaction, showing fundamental control by increasing or decreasing the amount of excitation and the implementation of a number of logic gates. We also demonstrated that the real chemical experiments can be seeded with learnt solutions evolved during modeling.

Our results may seem to be very complicated solutions to simple problems, especially in the case of changing the activity on the surface of the gel. A researcher with prior knowledge of the BZ reaction may suggest that we could have increased the activity simply by increasing the number of low light intensity cells or vice versa. Instead, considerable time was spent by the EA to solve the problem by positioning appropriate high and low light intensity cells so as to maneuver and split existing fragments. In addition, we do not fully understand how the EA has reached the various solutions to the set problems. This lack of transparency may be confusing and prevent researchers from designing systems that exploit such complex systems. However, we need to consider that natural and artificial evolutionary processes are "blind watchmakers"[46]—they have no comprehension of the problem they solve. While human researchers use their prior knowledge to make their decisions with a specific eye toward improvement and only implement the solutions that they consider good and/or useful, evolutionary processes have no preconceptions and choose solutions from all possible ones. They do not rule out steps because of prior knowledge and preconceptions, so they may find novel solutions that a human designer would never consider. Thus, there is the possibility that the use of this type of EA in chemistry may aid the discovery of novel phenomena that would in turn help in the understanding of underlying dynamical control and information processing in natural systems and therefore ultimately help the design of novel information processing technologies based around natural systems.

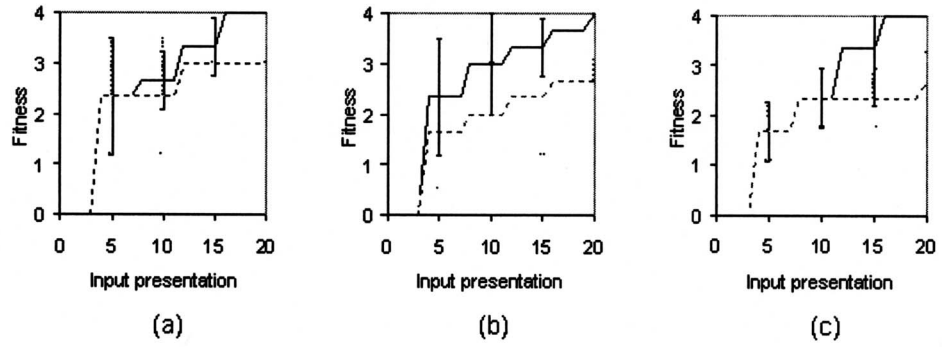

FIG. 11. Showing the typical fitness over time for (a) the AND gate (mutation rate 4000), (b) NAND gate (mutation rate of 4000), and (c) XOR gate (mutation rate of 6000) on the real chemical system. Dashed line: random initial controllers (three runs each).

In previous work, we introduced the possibility of constructing gates using a dynamical architectureless approach based on collision based computing.[6–10] We demonstrated both in computational[7,8] and experimental studies[9] using a subexcitable BZ system that under carefully controlled conditions compact wave fragments develop in the medium. These fragments then travel for reasonably long distances when undisturbed and their collisions can be interpreted as the implementation of logical operations.

The present paper advances the design of experimental and simulated prototypes of reaction-diffusion processors used to implement logical computation and which mimic a conventional hardware type approach with wires and gates in a fixed morphology.[17–24] We have developed a hybrid (automaton network plus BZ reaction) system, where the "hard-wired" architecture (the topology of excitation channels or wires) is dynamically changing over time to achieve the desired task.

To present the true and false values of Boolean variables in the BZ medium, we used a threshold function, namely, if the amount of excitation exceeds a certain amount then the system represents logical truth, otherwise falsity. This was done purely for simplicity in these initial experiments. In principle, the experimental approach can be extended to n-valued logical systems by employing classical conventions: $x \wedge y = \min(x,y)$ and $x \vee y = \max(x,y)$. These operations, originally invented in n-valued postlogic, have been adopted in a variety of logical systems, including fuzzy and continuous-valued logics. The choice of negation operator in such chemical n-valued logical circuits may depend on many factors; however, we do not foresee any problems in implementing cyclical negation. Some preliminary results on implementing a wide range of logics in geometrically constrained BZ systems have been already obtained.[24]

We are also planning to explore the use of EAs to design collision-based architectureless computers, where instead of the dynamically changing checkerboard image only one light level—the subexcitable light level—will be used. The use of larger gels, populations to evolve cell rules, and alternative evolutionary approaches to chemical computing are also being considered.

ACKNOWLEDGMENTS

This work was supported under EPSRC Grant No. GR/T11029.

[1] C. Darwin, *On the Origin of Species by Means of Natural Selection, or the Preservation of Favored Races in the Struggle for Life* (John Murray, London, 1859); modern reprint *Charles Darwin, Julian Huxley, The Origin of Species*, Signet Classics (Penguin, New York, 2003).
[2] S. A. Kauffman, *The Origins of Order: Self-Organization and Selection in Evolution* (Oxford University Press, Oxford, 1993).
[3] N. C. Spitzer and T. J. Sejnowski, Science **277**, 1060 (1997).
[4] D. Bray, Nature (London) **376**, 307 (1995).
[5] A. Adamatzky, *Computing in Nonlinear Media and Automata Collectives* (Institute of Physics, Bristol, 2001).
[6] *Collision Based Computing*, edited by A. Adamatzky (Springer, New York, 2003).
[7] A. Adamatzky, Chaos, Solitons Fractals **21**, 1259 (2004).
[8] A. Adamatzky and B. De Lacy Costello, Chaos, Solitons Fractals **34**, 307 (2007).
[9] B. De Lacy Costello and A. Adamatzky, Chaos, Solitons Fractals **25**, 535 (2005).
[10] A. Adamatzky, B. De Lacy Costello, and T. Asai, *Reaction Diffusion Computers* (Elsevier, New York, 2005).
[11] N. G. Rambidi, in *Molecular Computing*, edited by T. Seinko, A. Adamatzky, N. Rambidi, and M. Conran (MIT, Cambridge, MA, 2003).
[12] A. Adamatzky, B. de Lacy Costello, and N. Ratcliffe, Phys. Lett. A **297**, 344 (2002).
[13] O. Steinbock, A. Toth, and K. Showalter, Science **267**, 868 (1995).
[14] A. Adamatzky and B. de Lacy Costello, Chaos, Solitons Fractals **16**, 727 (2003).
[15] A. Adamatzky, P. Arena, A. Basile, R. Carmona-Galán, B. de Lacy Costello, L. Fortuna, M. Frasca, and A. Rodríguez-Vázquez, IEEE Trans. Circuits Syst., I: Regul. Pap. **51**, 926 (2004).
[16] A. Adamatzky and B. De Lacy Costello, Phys. Lett. A **309**, 397 (2003).
[17] K. Agladze, R. R. Aliev, T. Yamaguchi, and K. Yoshikawa, J. Phys. Chem. **100**, 13895 (1996).
[18] J. Gorecki, K. Yoshikawa, and Y. Igarashi, J. Phys. Chem. A **107**, 1664 (2003).
[19] I. N. Motoike and K. Yoshikawa, Chaos, Solitons Fractals **17**, 455 (2003).
[20] I. N. Motoike, K. Yoshikawa, Y. Iguchi, and S. Nakata, Phys. Rev. E **63**, 036220 (2001).
[21] A. Toth and K. Showalter, J. Chem. Phys. **103**, 2058 (1995).
[22] J. Sielewiesiuka and J. Gorecki, Phys. Chem. Chem. Phys. **4**, 1326 (2002).
[23] O. Steinbock, P. Kettunen, and K. Showalter, J. Phys. Chem. **100**, 18970 (1996).
[24] I. N. Moitoke and A. Adamatzky, Chaos, Solitons Fractals **24**(1), 107 (2005).
[25] O. E. Rössler, in *Physics and Mathematics of the Nervous System*, edited by M. Conrad, W. Güttinger, and M. Dal Cin (Springer, Berlin, 1974).
[26] A. Hjelmfelt, E. D. Weinberger, and J. Ross, Proc. Natl. Acad. Sci. U.S.A. **88**, 10983 (1991).
[27] A. Hjelmfelt, E. D. Weinberger, and J. Ross, Proc. Natl. Acad. Sci. U.S.A. **89**, 383 (1992).
[28] A. Hjelmfelt, F. W. Schneider, and J. Ross, Science **260**, 335 (1993).
[29] A. Hjelmfelt and J. Ross, J. Phys. Chem. **97**, 7988 (1993).
[30] D. Lebender and F. W. Schneider, J. Phys. Chem. **98**, 7533 (1994).
[31] K.-P. Zeyer, G. Dechert, W. Hohmann, R. Blittersdorf, and F. W. Schneider, Z. Naturforsch., A: Phys. Sci. **49**, 953 (1994).
[32] M. Okamoto, T. Sakai, and K. Hayashi, Biol. Cybern. **58**, 295 (1988).
[33] J. Wang, S. Kádár, P. Jung, and K. Showalter, Phys. Rev. Lett. **82**, 855 (1999).
[34] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing* (Springer, New York, 2003).
[35] K. DeJong, *Evolutionary Computation* (MIT, Cambridge, MA, 2005).
[36] A. Assion, T. Baumert, M. Bergt, T. Brixner, B. Kiefer, V. Seyfried, M. Strehle, and G. Gerber, Science **282**, 919 (1998).
[37] G. Valerie, J. Comput.-Aided Mol. Des. **16**, 371 (2002).
[38] S. Harding and J. Miller, Proceedings of the IEEE Congress on Evolutionary Computation, IEEE, 2004 (unpublished), p. 1800.
[39] J. M. Tour, W. L. Van Zandt, C. P. Husband, S. M. Husband, L. S. Wilson, P. D. Franzon, and D. P. Nackashi, IEEE Trans. Nanotechnol. **1**, 100 (2002).
[40] M. Sipper, *Evolution of Parallel Cellular Machines* (Springer, New York, 1997).
[41] S. A. Kauffman, *On the Origins of Order* (Oxford, New York, 1993).
[42] S. Kádár, T. Amemiya, and K. Showalter, J. Phys. Chem. **101**, 8200 (1997).
[43] H.-J. Krug, L. Pohlmann, and L. Kuhnert, J. Phys. Chem. **94**, 4862 (1990).
[44] Y. Gao and H.-D. Försterling, J. Phys. Chem. **99**, 8638 (1995).
[45] I. Cassidy and S. C. Müller, Phys. Rev. E **74**, 026206 (2006).
[46] R. Dawkins, *The Blind Watchmaker* (W. W. Norton & Company, Inc., New York, 1996).