

Cellular Automata: Statistical Models With Various Applications

Catherine Beauchemin, Department of Physics, University of Alberta

April 22, 2002

Abstract

This project offers an overview of the use of Cellular Automata (CA) as a statistical technique to study complex systems. First, elementary CA, such as the ones studied by Wolfram, will be presented in detail. Then, the role of CA as a tool to study statistical mechanics systems will be demonstrated through the simulation of the Ising spin model. Finally the expandable nature of CA will be explored through the simulation of a more complex CA: pheromone trail laying and following by ants.

Contents

1	Introduction: The Vast World of CA	1
2	Wolfram's Elementary CA	2
3	Describing a CA in Statistical Mechanic Terms	4
4	Describing a Statistical System in terms of a CA	6
5	Extending CA to Multi-Agent Systems	9
5.1	An Algorithm for Trail Network Formation and Trail Following by Ants	10
5.2	Results from the Simulation	13
6	Conclusion: Cellular Automata as an Extension of Statistical Mechanics	16
	References	17

1 Introduction: The Vast World of CA

Self-organizing behavior is a very hot topic these days. In fact, it has become such a trend that scientists from all areas are now trying to apply it to an always wider variety of problems. Such buzzwords as “Emergence”, “Distributed Intelligence”, “Stigmergy” or “Swarm Intelligence” are gaining increasing popularity.

Self-organization occurs in systems composed of many extremely simple agents which interact through very simple rules to collectively exhibit extremely complex behavior. The intelligence is neither contained in the individuals, nor in a centralized information structure. Rather, the intelligence or complexity of those systems arises from the multiple interactions between the individuals. Just like droplets accumulating slowly with simple rules can give rise to the still poorly-understood snowflake, termites can build such an intricate nest that it keeps the colony at a constant humidity level — a real marvel of architecture.

The complexity of those systems comes from the extremely large number of agents that compose them. Hence, a statistical approach to CA seems to be the ideal way to mimic and hopefully gain insights into the mechanisms that rule these impressive systems.

In Section 2, the concepts of Cellular Automaton (CA) will be introduced. A good way to do this is to present examples of elementary CA, such as those studied in great detail by Stephen Wolfram. Section 3 suggests a statistical mechanics approach to describing the elementary CA. In Section 4 the approach is reversed and a statistical mechanic system, the 1-D Ising spin model, will be solved using CA simulations. Results from the simulations will be analyzed and discussed. Finally, Section 5 will demonstrate how the use of CA can be extended to more complex systems. The particular case of trail laying and following by ants will be explored through CA simulations. Again, results from the simulations will be analyzed and discussed along with a brief statistical analysis of the system.

2 Wolfram’s Elementary CA

Cellular automata were originally introduced by John von Neumann and S. Ulam under the name of “cellular spaces” as possible idealization of biological systems (von Neumann, 1963, 1966). They sought to show that biological processes such as the reproduction and evolution of organized forms could be modeled by simple cells following local rules for changing a cell parameter with time [5].

A simple CA usually consists of a regular uniform lattice (or “array”), that can either be finite or infinite (periodic boundaries) in extent. The lattice contains a discrete variable (or “cell”) at each site, that can assume m possible discrete values.

The state of a CA is completely specified by the values of the variables at each site. The CA evolves in discrete time steps, with the value of the variable at a given site being affected by the values of variables at sites in its “neighborhood” on the previous time step. The “neighborhood” of a site can be defined in numerous ways and can be as extended as one wishes. At a new time step, the variables at all sites are updated simultaneously, based on the values of the variables and their defined “neighborhood” at the preceding time step, and according to a definite set of local rules [5, 8].

Stephen Wolfram has done most of the early work on CA. In the ’80s, he published an extensive series of articles on CA. He concentrated most of his work on elementary CA, namely 1-D CA with only two possible values at each site: 0 or 1. Typically, the neighborhood he chose was defined to be the cell itself plus its two immediate neighbors: the cells to its left and right. This type of CA as been thoroughly studied by Wolfram and will be the main subject of this section.

From the definition of the neighborhood, it follows that the local rules for the evolution of the CA will have to be defined for each combination of states of the 3 sites involved in defining the subsequent state of a single site. For the sake of clarity, let’s label lc , cc and rc — for left, center and right cell, respectively — the 3 cells involved in the determination of the state of the cc in the next time step. The set of possible configurations for the 3 cells is then defined in the following manner:

111	110	101	100	011	010	001	000
0	1	0	1	1	0	1	0

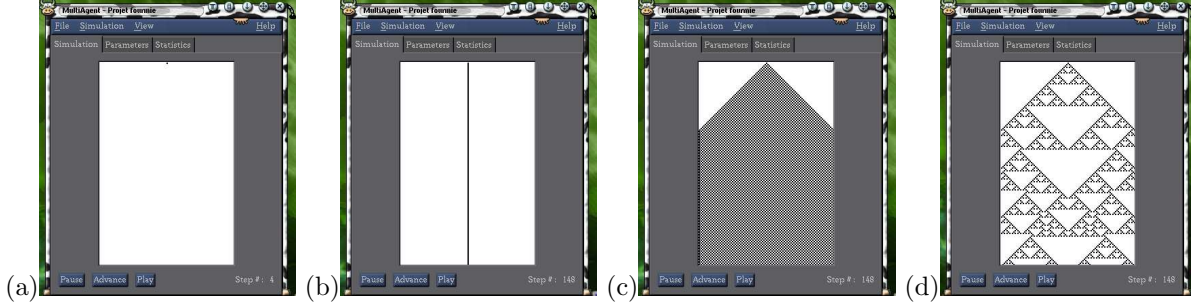


Figure 1: Evolution of CA using rules of (a) type 1 (Rule 232), (b) type 2 (Rule 36), (c) type 3 (Rule 178), and (d) type 4 (Rule 90).

where the bottom line indicates the state of the cc at the next time step given the (lc, cc, rc) configuration on the top line. This means that any rule can be defined by an eight digit binary number where each digit would represent the next state of the cc given its current state and that of its immediate neighbors. The rule detailed on the second line is called “Rule 90”, from the binary number 01011010 defining it. Recall that a binary number is read in the following manner:

$$01011010 = 0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 90 \quad (1)$$

In all, for a CA based on a 3-cell local rule, there are 8 configurations for which a result must be specified which makes for a total of $2^8 = 256$ rules. However, those rules are usually reduced to a total number of 32 “legal” rules. The legal rules come from the fact that the CA were initially used for modeling biological processes such as the reproduction of organized forms. This meant, for example, that a starting configuration of all-zero sites, corresponding to the absence of any cells, should remain an all-zero configuration as it evolves in time — the triplet 000 should always give 0. Secondly, the rules should be symmetric so that the left and right cells are interchangeable. This forces the rules for configurations 100 and 001 to be equal and for 110 and 011 to be equal [9]. Thus, the “legal” rules can be written as:

$$\alpha_1\alpha_2\alpha_3\alpha_4\alpha_2\alpha_5\alpha_4\alpha_1 \quad (2)$$

where each α_i can be 0 or 1; thus the number of legal rules is $2^5 = 32$.

If one decides to start the CA with only one non-zero seed, interesting patterns develop. The 32 legal rules can then be divided into 4 types according to their behaviour when seeded with a single cell. They are:

- type 1: disappear with time (0, 32, 72, 104, 128, 160, 200 and 232)
- type 2: merely copy the non-zero cell forever (4, 36, 76, 108, 132, 164, 204 and 236)
- type 3: yield a completely uniform type of rows (50, 122, 178, 222, 250 and 254) or uniform pairs of rows (54 and 94)
- type 4: develop nontrivial patterns (18, 22, 90, 126, 146, 150, 182 and 218)

An example for each of the 4 types is presented in Figure 1. Each of the 4 types contains 8 rules: this is no coincidence. What determines the personality of the outcome of a certain rule is the combination of its local rules. It is then to be expected that by looking closer at the underlying local rules that are common to the rules of a same type, one can learn how the local rules affect the resulting structures. For example,

- type 1: $\alpha_4 = 0$ and $\alpha_5 = 0$,
- type 2: $\alpha_4 = 0$ and $\alpha_5 = 1$, and
- type 3,4: $\alpha_4 = 1$.

Recall that α_4 specifies the result for the configurations 100 and 001 while α_5 specifies the result for the configuration 010.

The three first types are said to be “simple” CA. The last type however, contains more elaborate CA that develop nontrivial patterns. At infinite time, those nontrivial patterns become self-similar fractals. In fact, all Rules of type 4, with the exception of Rule 150, generate a Sierpinski gasket with $D = 1.5850$. Rule 150, the exception, is the golden ratio, $D = 1.6942$ [5].

3 Describing a CA in Statistical Mechanic Terms

According to the On-Line Harcourt Science and Technology Dictionary [4], Statistical Mechanics is “a branch of physics that is concerned with the macroscopic behavior of a system based on the theory and treatment of microscopic interactions between the constituent parts of the system, in which the number of constituent parts is usually very large.” It is already obvious from this definition that CA have a lot in common with statistical mechanics. Parallels will now be established between a statistical system and a CA. It will be shown that it is possible to describe a statistical system using a CA just like it is possible to describe a CA in terms of statistical mechanics.

Going back to a 1-D Wolfram elementary CA, let’s imagine a single row of N lattice points, where each lattice point represents a particle. Let us say that the values they can take is 0 or 1 indicating, for example, the spin orientation. A microstate of this system is a row of N particles with a given sequence of spins represented by the values 0 and 1. In total there are 2^N possible microstates all with an equal probability of occurring¹. From this definition, it is possible to define a macrostate as the set of all N -celled microstates containing x cells of value 1. If one of the microstates is submitted to a specific rule, for example Rule 90, a new microstate will result. Below is an example of the resulting microstate (2nd line) when Rule 90 is applied to an original microstate (1st line) — note that periodic boundary conditions are used.

```
Original :  0  1  1  0  1  0  0  0
Resulting : 0  1  1  0  0  1  0  0
```

Thus, given a starting microstate, time evolution will result in a trajectory through subsequent microstates. The trajectory is dictated by the rule and may or may not lead back to the starting microstate. Assuming each microstate exists in a starting ensemble of configurations with equal probability of occurrence², time evolution will likely merge many configurations into a few and soon trajectories starting from almost all initial configurations are concentrated onto short cycles, called attractors, containing only a few configurations [5]. Hence the trajectories traced out by the time evolution of several microstates may coalesce, but may never split. For instance, the extremely trivial Rule 0 will turn any starting microstate into the null microstate (all lattice points are 0), so that in one time step, all configurations reduce to just one: this is called a stable fixed point.

Almost all of the rules seen in Section 2 are irreversible, which means that although it is possible to identify a unique resulting microstate for each original microstate, it is not possible to identify a unique original

¹Provided that 0’s and 1’s have an equal probability of occurring.

²Again, provided that 0’s and 1’s have an equal probability of occurring.

microstate for each resulting microstate. In other words, a particular microstate has unique descendents, but does not necessarily have unique ancestors. Rule 240 is the lone exception since it is the identity rule: each microstate is mapped only to itself. As such, Rule 240 is a reversible rule i.e. a rule in which each microstate has a unique descendent and a unique ancestor. For a reversible system, trajectories representing time evolution of different states never intersect or meet. Thus, for reversible systems, the total number of possible configurations will remain constant with time. In the case of an irreversible rule, the total number of possible microstates will be reduced as time evolves. For some systems, all trajectories might reduce to a fixed point solution (a single microstate like in the case of Rule 0), while in other cases, it might reduce to one or a few short cycles or attractors.

Irreversible behavior in CA may be analyzed by considering the behavior of their “entropy”. Entropy is defined, as usual, as the logarithm (in this case base two) of the average number of possible states of a system, that is

$$S(\tau) = \sum_i p_i(\tau) \log_2 \left(\frac{1}{p_i(\tau)} \right) \quad (3)$$

where $p_i(\tau)$ is the probability of obtaining microstate i after τ time steps. $\tau = 0$ corresponds to the starting equiprobable ensemble with all $p_i(0) = 1/2^N$. To understand what this entropy effectively means for an N cell elementary CA, let us look at a few examples. Consider a 2-cell elementary CA. The starting equiprobable ensemble will then be composed of

Microstate 1 at time 0 : 0 0 with $p_1(0) = 1/4$
 Microstate 2 at time 0 : 0 1 with $p_2(0) = 1/4$
 Microstate 3 at time 0 : 1 0 with $p_3(0) = 1/4$
 Microstate 4 at time 0 : 1 1 with $p_4(0) = 1/4$

thus, $S(0) = 4 * (1/4) \log_2(4) = \log_2(4) = 2$. This demonstrates that the entropy, $S = 2$, corresponds to the number of cells necessary to represent the 4 possible microstates.

If the CA is left to evolve for one time step, i.e. $\tau = 1$, according to Rule 18 (00010110) for example, the microstate ensemble would reduce to

(0) 0 0 (0)	(1) 0 1 (0)	(0) 1 0 (1)	(1) 1 1 (1)
↓	↓	↓	↓
0 0	0 1	1 0	0 0

where the values in parenthesis are the states required by the boundaries in order to impose periodic boundary conditions. Note that at time step 1 the system cannot be found in the configuration 11 — the system of 2 cells can now only be in one of three microstates. The distribution of microstates at $\tau = 1$ is

Microstate 1 at time 1 : 0 0 with $p_1(1) = 2/4$
 Microstate 2 at time 1 : 0 1 with $p_2(1) = 1/4$
 Microstate 3 at time 1 : 1 0 with $p_3(1) = 1/4$
 Microstate 4 at time 1 : 1 1 with $p_4(1) = 0/4$

which gives $S(1) = (2/4) \log_2(4/2) + 2 * (1/4) \log_2(4/1) = 0.5(1) + 0.5(2) = 1.5$. This means that the system would effectively only need 1.5 cells to be fully described instead of the $\log_2(3) \approx 1.58$ cells that one would expect to need for the representation of a system with 3 microstates. The smaller entropy is due to the microstates not being equally probable — an expression of the system’s tendency towards organization.

From this example, it is quite clear that those elementary CA are self-organizing, tending to reduce in complexity as they evolve. Entropy is typically maximized when a system is completely disorganized, i.e. when all the microstates act independantly. Thus, the entropy of a CA takes on its maximal value of N (or $\log_2(\# \text{ of states})$) for an equiprobable ensemble. For an irreversible system, one can expect the entropy of the system to go down and eventually stabilize to a given value as τ is increased. Figure 2 below shows

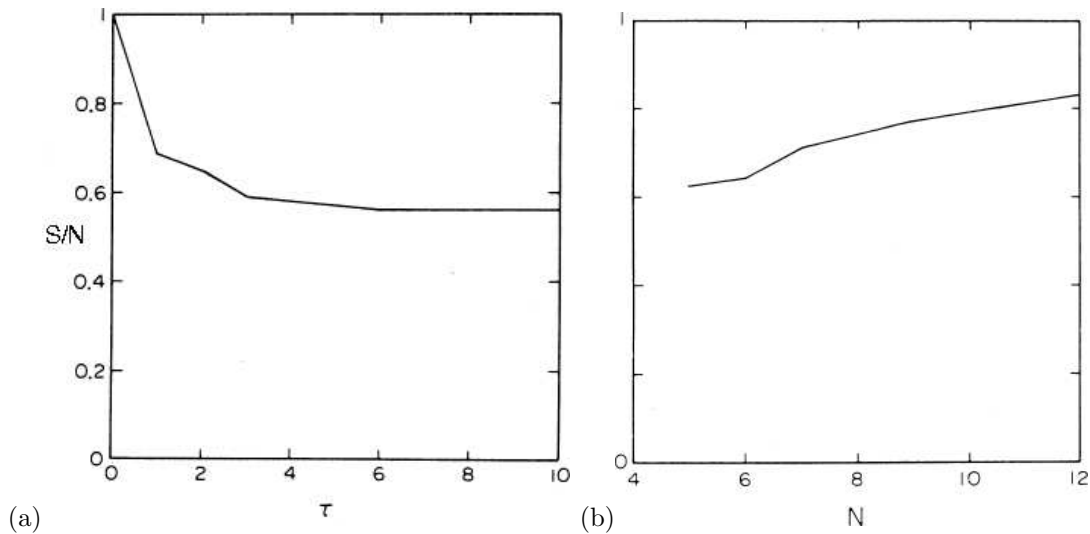


Figure 2: (a) Decrease of the entropy per site $S(\tau)/N$ as τ increases for an ensemble of elementary CA ($N = 10$) evolving according to rule 126 from a starting equiprobable ensemble. (b) The fraction of the 2^N possible configurations of a length N elementary CA (with periodic boundary conditions) *not* reached by evolution (no matter how many time steps τ are taken) from an arbitrary initial configuration under rule 126 as a function of N [8].

how the entropy of Rule 126 (irreversible) varies as τ is increased. It also shows how the fraction of possible configurations that will not be reached increases in time as a function of N . Self-organizing behavior is indicated by a decrease of entropy. The trajectories corresponding to the evolution of microstates are found to become concentrated in limited regions and do not asymptotically fill the available volume densely and uniformly. This “thinning” phenomenon makes self-organization possible, by allowing some configurations to occur with larger probabilities than others even in the large-time equilibrium limit [8]. As can be seen from Figure 2, the fraction of the 2^N configurations of a length of N cellular automata (with periodic boundary conditions) not reached by evolution from an arbitrary starting configuration will increase toward unity as N increases.

Coming back to statistical mechanics, the second law of thermodynamics implies that isolated microscopically reversible physical systems tend with time to states of maximal entropy and maximal “disorder”. However, “dissipative” systems involving microscopic irreversibility, or those open to interactions with their environment, may evolve from “disordered” to more “ordered” states. The states attained often exhibit a complicated structure. Examples are outlines of snowflakes, patterns of flow in turbulent fluids, and biological systems [8]. This is quite consistent with what has been exposed above regarding the evolutions of reversible and irreversible elementary CA.

4 Describing a Statistical System in terms of a CA

In Section 3, it was seen that it is possible to describe a CA in terms of one of the most important statistical mechanic quantities, entropy. In this section, it will be shown that it is possible to describe a statistical system in terms of a CA and solve for it. Let us consider an isolated system consisting of N localized atoms. Each atom has a spin which can be up or down, represented respectively by $+1$ or -1 . This is called the

Ising model. The energy of the system is given by

$$E = -J \sum_i S_i S_{i+1}, \quad (4)$$

the magnetization is given by

$$M = \sum_i S_i, \quad (5)$$

the specific heat is given by

$$C = \frac{\delta \langle E \rangle}{\delta T} = \frac{\langle E^2 \rangle - \langle E \rangle^2}{kT^2}, \quad (6)$$

the susceptibility is given by

$$\chi = \frac{\delta \langle M \rangle}{\delta H} = \frac{\langle M^2 \rangle - \langle M \rangle^2}{kT}, \quad (7)$$

and the probability of flipping a spin by

$$P = \begin{cases} e^{-\Delta E/(kT)} & , \Delta E > 0 \\ 1 & , \Delta E \leq 0 \end{cases}. \quad (8)$$

Usually the Boltzmann constant, k_B , is taken to be 1 so that the temperature, T , can be expressed in units of energy.

Just like for the elementary CA above, the atoms can be arranged on a row of N cells. The simulation can be started with all the spins aligned either up or down — called “cold start” — or with a random arrangement of spins — called at “hot start”. To make the simulation evolve in time, a cell is picked at random and its spin is flipped with probability given by Equation 8. The following example will show what possible values ΔE and ΔM can take. Just like for the elementary CA above, the value of the centre cell at the next time step will depend on its own current value and the current value of its two immediate neighbors which can either be parallel or antiparallel to one another. Here are the 4 possible situations:

$$\begin{array}{l} \uparrow\uparrow\uparrow \implies \uparrow\downarrow\uparrow \rightarrow \Delta E = +4J; \quad \Delta M = -2 \\ \uparrow\downarrow\uparrow \implies \uparrow\uparrow\uparrow \rightarrow \Delta E = -4J; \quad \Delta M = +2 \\ \uparrow\uparrow\downarrow \implies \uparrow\downarrow\downarrow \rightarrow \Delta E = 0; \quad \Delta M = -2 \\ \uparrow\downarrow\downarrow \implies \uparrow\uparrow\downarrow \rightarrow \Delta E = 0; \quad \Delta M = +2 \end{array}$$

Note that the cases $\downarrow\downarrow\downarrow$, $\downarrow\uparrow\downarrow$, $\downarrow\downarrow\uparrow$ and $\downarrow\uparrow\uparrow$ have been omitted since the problem is symmetric, i.e. the quantity important to this model is the relative orientation not the absolute orientation. Thus, the probability of flipping a spin is dictated by the neighborhood of the atom (left, center and right atoms), but also by the external parameters J and T . Note that according to the definition of the probability of flipping a spin, the system will tend towards an equilibrium in which all spins will be parallel if $J > 0$ or anti-parallel if $J < 0$. Thus, a system with $J > 0$ is called a ferromagnet and one with $J < 0$ is called an antiferromagnet. The speed at which the system will tend towards equilibrium depends on J and T . The greater J/T is, the faster the system will tend to equilibrium because the probability of flipping a spin is greater for bigger values of J/T . Finally, when the system has attained equilibrium, it will fluctuate around it. The amplitude of the fluctuations will again depend on J and T ; the larger J/T , the greater the amplitude of the fluctuations.

At $T = 0$, the probability of flipping a spin is $P = 0$ for $\Delta E > 0$ or 1 for $\Delta E \leq 0$. Thus, at $T = 0$, for a uniform initial configuration (all spins up or all spins down), the system will remain indefinitely in the same state with no fluctuations. For a random initial configuration, at $T = 0$, one can expect all the spins to very slowly flip to eventually attain equilibrium. As T is increased, keeping $J \neq 0$ constant, the system will attain equilibrium faster and fluctuations about it will get larger.

Using Equations 4 to 8 presented above, I implemented a CA of the Ising Spin Model. The results are presented in Figure 3. As can be seen from Figure 3, for high temperature the results are in acceptable

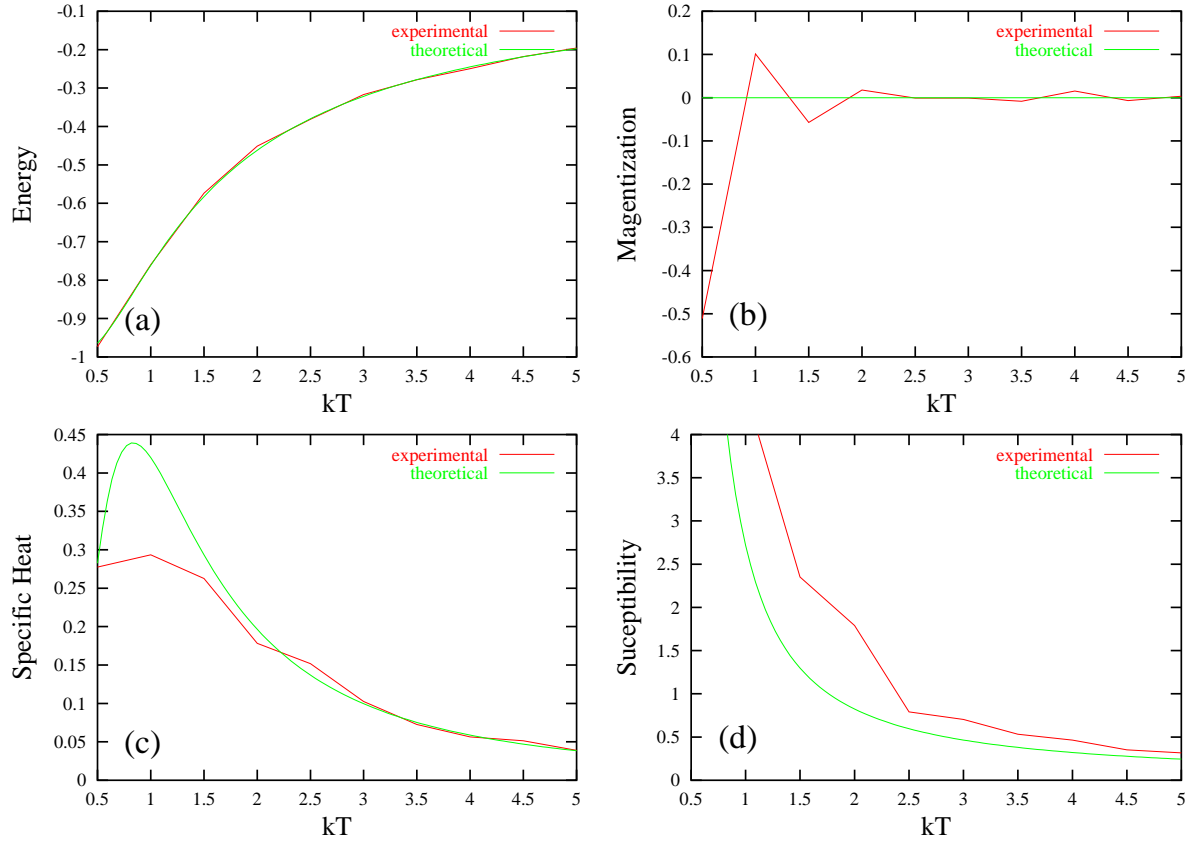


Figure 3: Results from simulations of the Ising Spin Model. It presents (a) Energy, (b) Magnetization, (c) Specific Heat and (d) Suceptibility as a function of kT (where $k = 1$ and $B = 1 \times 10^{-9}$).

agreement with what the theory predicts. The theoretical values for energy, magnetization, specific heat and susceptibility presented in Figure 3 were computed using the following equations [6].

$$\frac{E}{N} = -\tanh\left(\frac{J}{kT}\right) \quad (9)$$

$$\frac{M}{N} = \frac{e^{J/kT} \sinh(B/kT)}{\sqrt{e^{2J/kT} \sinh^2(B/kT) + e^{-2J/kT}}} \quad (10)$$

$$\frac{C}{N} = \frac{1}{N} \frac{dE}{dT} = \frac{(J/kT)^2}{\cosh^2(J/kT)} \quad (11)$$

$$\frac{\chi}{N} = \frac{1}{N} \frac{dM}{dB} = \frac{1}{kT} \frac{e^{-J/kT} \cosh(B/kT)}{[e^{2J/kT} \sinh^2(B/kT) + e^{-2J/kT}]^{3/2}} \quad (12)$$

Unfortunately, a simple model such as this has its limits. Its approach to thermal equilibrium is qualitatively rather than quantitatively correct. Further, since there is very little or no magnetic field at all, there is no preferred direction in space, which means that the average magnetization may vanish, and this may lead to instabilities in which the spins spontaneously reverse. In addition, the phase transition at the Curie temperature, a characteristic of magnetic materials, simply does not occur in the 1-D version of the model [6]. Finally, at low temperature, good statistics are harder to compute since it takes a long time for the system to equilibrate since most spin flips are rejected.

The Ising model does, however, remain a very good example of the use of CA to solve a statistical mechanics problem. In the next section, we will see how CA of greater complexity can be used to model multi-agent systems in an attempt to extract, from the simple local rules, the global underlying mechanisms of such systems.

5 Extending CA to Multi-Agent Systems

It has been seen that statistical mechanic systems can be studied using CA. It has also been seen that CA can be described in terms of statistical mechanical quantities such as entropy. That is just a small part of what CA are about.

Statistical mechanics is a powerful tool for studying systems of many identical elements. Depending on the constraints on the system, equations of increasing complexity are used to efficiently describe the system. For example, the microcanonical ensemble is used to study a system where the total number of particles, N , the volume, V , and the energy, E , of the system are constant. The canonical ensemble lets go of the constraint on E but still keeps N and V fixed. Finally, the grand canonical ensemble relaxes the constraints on N and E , allowing them to fluctuate, while keeping V constant [5].

Even if statistical mechanics is a very powerful tool, it has its limitations. While it is good at describing systems where the number of agents is very large (such that it is possible to take the continuum approximation), it gives much poorer results when N is large but not enough to take the continuum approximation. Furthermore, statistical mechanics is very appropriate for defining a system at its equilibrium state but soon becomes too complex when the system is out of equilibrium. Finally, statistical mechanics is very limited at describing systems of strongly interacting agents. When interactions are not weak and therefore cannot be neglected, it might still be possible to explain a system in terms of a statistical mechanism, but the formalism will be very particular and extremely limited in its applications. It goes without saying that this would involve numerous approximations which might lead to poorly satisfying results.

The term Multi-Agent Systems (MAS) is used here to describe any system constituted of a great number

of simple identical agents (molecules, insects, cells) interacting either directly or through their environment. We will be interested in such systems that, from the simple interactions of their simple constituting agents, will demonstrate complex behavior. The intelligence of such systems is neither found among their agents nor is it centralized in an intelligent decision-making unit. Intelligence will simply be the result of the multitude of interactions among the multitude of agents. Such intelligence has been appropriately named “Swarm Intelligence” and is a very hot topic at the moment.

Let us take the example of a very popular MAS: pheromone trail laying and following by ants. Pheromones are at the very centre of the ant communication system. Those chemical messages are used to signal danger, identify a successful trail and even stimulate the aggressiveness of army ants during a raid. The scenario of pheromone trail laying starts with an exploring ant discovering a food source. After having ingested as much food as possible, the ant returns to the nest laying a pheromone trail. The trail will in turn stimulate foraging ants at the nest. Some of those will manage to follow the pheromone trail all the way to the source, ingest food and go back to the nest marking again the trail they followed to the source. Thus, those new recruits become recruiters and the pheromone trail, getting marked stronger and stronger, will recruit more and more ants.

Of course, this type of foraging recruitment is rarely infallible and in a system where this recruitment technique dominates, it is imperative to take a closer look at the recruited ants that will never make it to the source. In the literature, they are called lost ants [3] or exploring ants [2] — they are foraging ants that do not exploit any food source.

Lost ants explore the foraging territory for a certain amount of time before returning to the nest — hence the name “exploring ants”. This exploration process, an essentially random walk, gives rise to a normal distribution of trajectories centred around the pheromone trail. Ant species with a low level of noise (high precision in track following) are observed to have a narrow distribution (low variance), whereas species with a high level of noise tend to show a larger distribution [3]. Although the lost ants do not exploit any of the known sources, they can discover new food sources. It is then very interesting to investigate what would be the best balance between low accuracy allowing discoveries of new food source, and high accuracy favouring immediate exploitation (see [3] and additional references suggested therein).

5.1 An Algorithm for Trail Network Formation and Trail Following by Ants

My ant simulation is much more simple than this. It is restrained to the mechanisms of the early stages of trail laying and trail following by ants. The project has been taken directly from Watmough and Edelstein-Keshet’s 1995 paper on “Modelling the Formation of Trail Networks by Foraging Ants”. The paper studied a very particular case: the initial formation of a trail by ants emerging from a central nest location. The goal of the study was to understand how the behaviour of individual ants determines the patterns of the resulting trail network [7].

This simple model consists of ants moving around in a defined area, that we will call the arena. As they move, they deposit pheromone and perform an almost random walk biased by the local trail concentration. Here are the basic rules of the algorithm:

- the ants move at a fixed speed of one square per step;
- as each ant moves it deposits pheromone at a constant rate of $\tau = 8$ “units” per time step per ant;
- the pheromone evaporates at a steady rate of “1 unit” per time step;
- saturation of the ant’s antennae occurs at a pheromone level of $C_S = 100$ “units”;

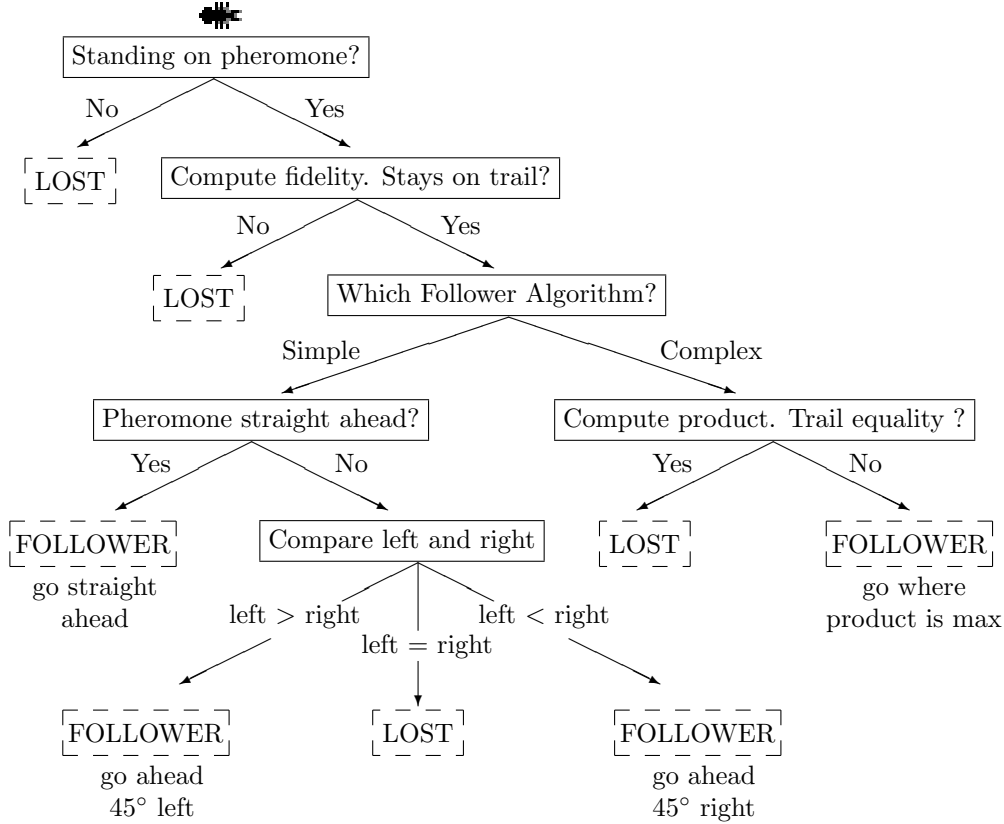


Figure 4: Decision tree for the movement of an ant in `ma_ants`.

- a lost ant (an ant not following a trail) will move randomly but with a turning kernel specifying the probability of making a turn given the angle of the turn;
- a following ant will move where the product of the turning kernel and the trail concentration gives the highest value;
- absorbing boundaries are used which means that ants stepping out of the arena will simply be removed from the simulation;
- ants are added to the simulation at the nest (centre of the arena) at a rate of one per time step, up to a maximum of `MAXANTS` ants.

Figure 4 presents the algorithm for moving an ant as a decision tree. Let us look at how the algorithm unfolds following this decision tree. The “turning kernel” is used to determine the motion of a lost ant. A turning kernel associates a probability, B_n , with a turn of $45n^\circ$ (direction irrelevant). In my simulation, as well as in the article, the “standard” turning kernel, B , is defined as $\{B_1 = 0.360, B_2 = 0.047, B_3 = 0.008, B_4 = 0.004\}$.

A random walk biased by the turning kernel described above will define the manner in which a lost ant will move. An ant is automatically considered lost and will make its next move as such if there is no pheromone at its current location. So the first step is to check if the ant is standing on pheromone. If it is not, its movement will be determined by the lost ant function.

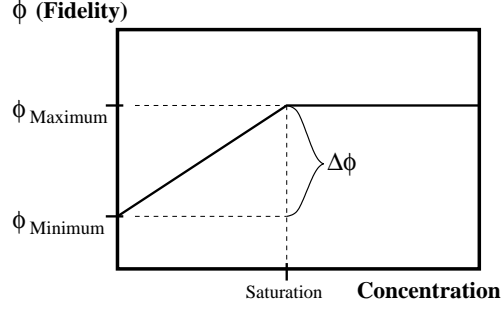


Figure 5: Graph of fidelity as a function of pheromone concentration. The fidelity, ϕ , increases in direct proportion to the trail concentration up to the saturation point.

If an ant is not a lost ant, i.e. it is currently standing on pheromone, it is then a follower. A follower follows a trail with a certain probability of losing it. The probability of staying on a trail is termed the fidelity, ϕ , and is usually dependant on the pheromone concentration of the trail at the ant's current position. Fidelity is assumed to be a linear function of the pheromone concentration up to the saturation level, where fidelity will plateau to a constant. The fidelity as a function of pheromone concentration is depicted in Figure 5. Thus, the probability of staying on the trail is given by:

$$\phi(C) = \Delta\phi \min \left\{ \frac{C}{C_S}, 1 \right\} + \phi_{\text{Minimum}} \quad (13)$$

where C is the pheromone concentration at the current position and C_S is the saturation point. To determine whether an ant stays on the trail, a random number between 0 and 255 inclusively is picked and the ant stays on the trail if the random number is smaller than the computed $\phi(C)$. If the random number is greater than $\phi(C)$, the ant is considered lost and will move according to the lost ant algorithm described above.

Once it has been established that the ant will be a follower, it will move according to a follower ant algorithm — what the authors call the “fork algorithm”. Among other things, the “fork algorithm” defines the behaviour of a follower ant when the trail it is following crosses other trails. In the article, two different fork algorithms are introduced; and like the article, two different algorithms were implemented for the rules determining the movement of a follower ant.

In the first algorithm, the follower considers each of the 8 moves available to it. Then the product of the pheromone concentration, which is at most the saturation level, and the trail swapping kernel (see explanation below) associated with that position is computed for each of the 8 possible moves. If equal probability is found between at least 2 of the positions, the ant is again considered lost and will move according to the lost ant algorithm. Otherwise, the move for which the product gave the highest value will be accepted and performed.

It is important to point out that the turning kernel used for the lost ant algorithm is not the same as that used by the follower ant algorithm — called the “trail swapping” kernel. The trail swapping kernel is defined as $T = \{72, 64, 24, 4, 0, 4, 24, 64\}$ for angles of $0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, -135^\circ, -90^\circ, -45^\circ$ respectively. And the product mentioned above is computed as:

$$\text{product} = T(\theta) \min \left\{ \frac{C(\theta)}{C_S}, 1 \right\} \quad (14)$$

where $T(\theta)$ and $C(\theta)$ are respectively the trail swapping kernel and pheromone concentration as a function of the angle of turn, θ , and C_S is the saturation value for the trail concentration.

The second algorithm is much simpler. The follower checks if there is pheromone ahead and moves there if there is. If not, the follower checks if there is pheromone at an angle of 45° to its left and right. If both the left and right pheromone concentrations are equal, the ant is considered lost and will move according to the lost ant algorithm. Otherwise, the ant will move to the position with the highest pheromone concentration.

Finally, if an ant steps out of the arena, it will be removed from the simulation (absorbing boundary conditions). This is an approximation of an infinite domain. The run times for the simulation are kept an order of magnitude below the mean time that an ant would arrive at the boundary by random motion alone. Thus the majority of the ants at the boundaries are on trails leading out of the region. And since we are concerned with the initial formation of the foraging trails, we can therefore assume that the return traffic is negligible [7].

5.2 Results from the Simulation

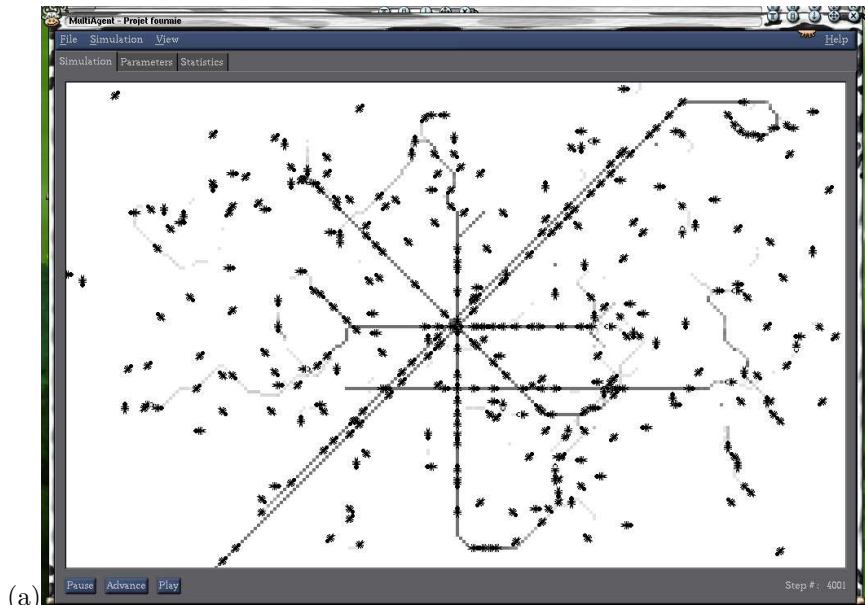
In the article, the results are presented in the form of screen shots of the program after a given number of time steps. This is done to illustrate the qualitative changes to the system resulting from changes to the simulation's parameters. Figure 6 and 7 present screen shots of my simulation along with the corresponding simulation presented in the article.

Before introducing the screen shots, the symbols used to designate the various parameters should be presented:

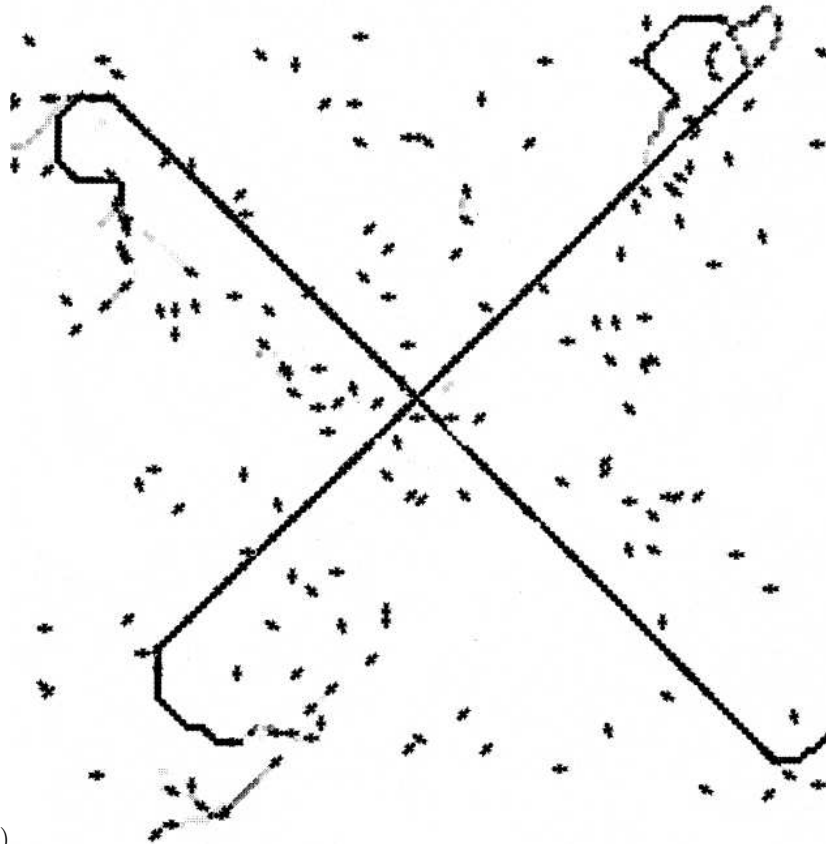
F	=	Total number of follower ants (a function of time)
L	=	Total number of lost ants (a function of time)
C	=	Pheromone concentration (a function of space and time)
C_S	=	Pheromone concentration at which the ant's antennae saturate
$\phi(C)$	=	Fidelity or probability of the ant remaining on the trail at each time step
ϕ_{\min}	=	The minimum value of the fidelity (the intercept of the $\phi(C)$ vs C linear graph, see Figure 5)
$\Delta\phi$	=	The difference between the value at which fidelity plateaus (when $C = C_S$) and ϕ_{\min}
τ	=	Rate at which ants deposit pheromone (how many "units" per time step)
B_n	=	The probability of an exploratory ant turning an angle of $45n^\circ$. Usually expressed as part of the turning kernel used in the lost ant algorithm and defined as $B = B_1, B_2, B_3, B_4$

In the captions of Figures 6 and 7, a value for F , the total number of followers, and L , the total number of lost ants, are presented. In the article, the authors introduce the ratio F/L as a quantitative measure of trail strength. It is said that since all trails are headed by lost ants, the ratio is an indication of the average number of followers per trail. The authors also point out that F/L represents the average trail strength and does not distinguish between the case where there are several trails of moderate strength and the case where there are a few strong trails and many weaker trails. The authors suggest that this distinction be made using a visual count of the number of trails.

In the current context, it might be more interesting to come back to statistical mechanics and try to find a way to determine the entropy of this system. Entropy is a very interesting quantity since it allows one to quantify the level of organization of a system and therefore is a good measure of the level of self-organizing that takes place in the simulated system. As usual, the entropy will be defined as the log of the number of states available to the system at every step. To assess that, let us look at the possible states that a single ant can take. If the ant is a lost ant, it can move to any one of the 8 positions available to it. If, however, the ant is a follower, it can only move to the one position that was determined by which ever "fork algorithm" was being used. Since the total number of states available to any given ant is 8, then \log_8 should be used.

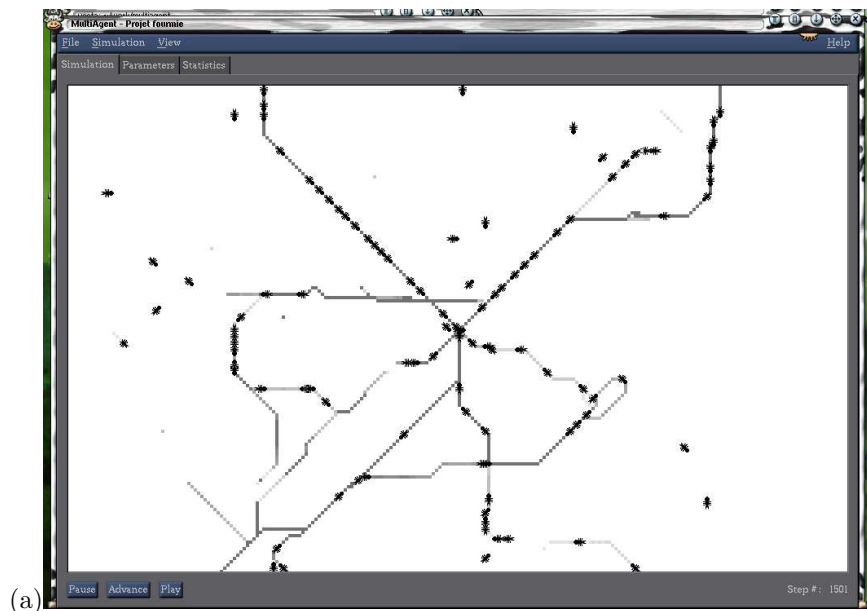


(a)

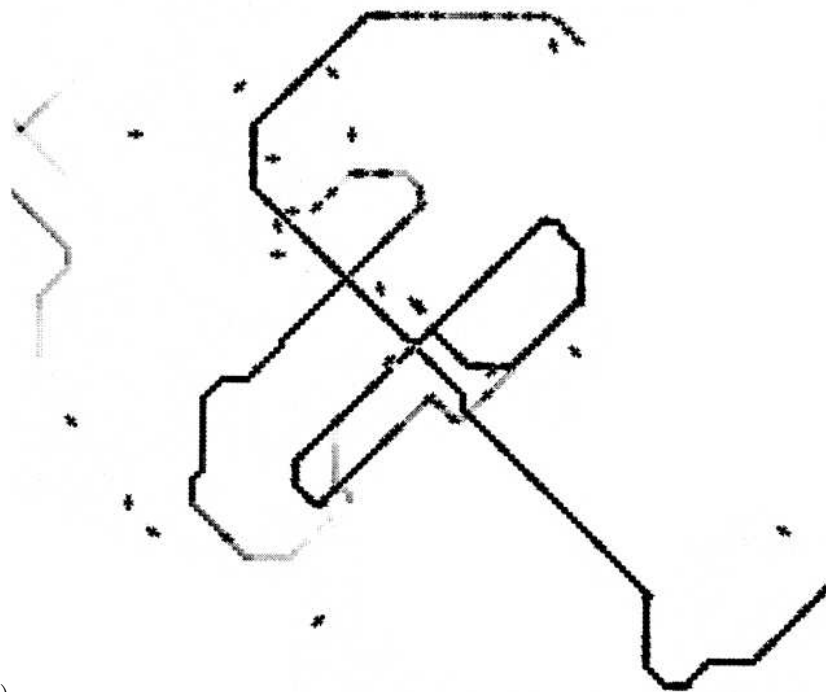


(b)

Figure 6: Testing the influence of the trail deposition rate on the trail network. (a) My simulation with $F(3955) = 203$, $L(3955) = 156$, so $F/L \approx 1.30$. (b) Their simulation [7, Figure 4(b)] with $F(?) = 148$, $L(?) = 108$, so $F/L \approx 1.37$. Both simulations were created using the following parameters: $\phi_{\min} = 247$, $\Delta\phi = 0$, $C_S = 100$, $\tau = 8$, $B = \{0.360, 0.047, 0.008, 0.004\}$ and the simple fork algorithm for 4000 steps.



(a)



(b)

Figure 7: Testing the influence of using a narrower turning kernel on the trail network. (a) My simulation with $F(1470) = 113$, $L(1470) = 22$, so $F/L \approx 5.1$. (b) Their simulation [7, Figure 5(a)] with $F(?) = 141$, $L(?) = 16$, so $F/L \approx 8.8$. Both simulations were created using the following parameters: $\phi_{\min} = 255$, $\Delta\phi = 0$, $C_S = 100$, $\tau = 8$, $B = \{0.135, 0.031, 0.008, 0.004\}$ and the simple fork algorithm for 1500 steps.

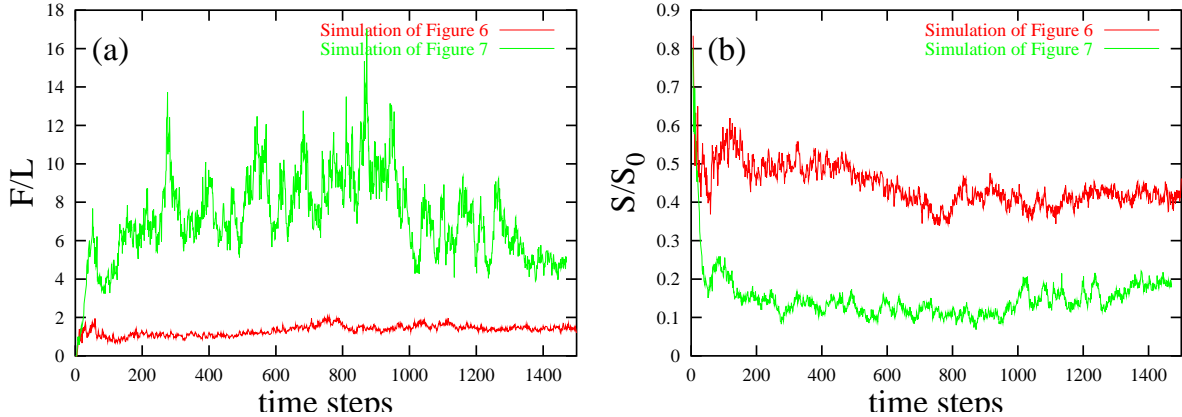


Figure 8: Normalized entropy as a function of time for the simulations presented in Figures 6 and 7.

This leads to the following definition for the “normalized” entropy of the system:

$$\frac{S}{S_0} = \frac{\log_8(8^L + F)}{\log_8(8^T)} = \frac{L + \log_8(F)}{T} \quad (15)$$

where S is the entropy, $S_0 = T$ is the maximum entropy, L is the number of lost ants, F is the number of follower ants, and T is the total number of ants. From this definition, the maximum normalized entropy, 1, is attained when all ants are lost ants.

This measure of the self-organization capability of the system can be tested for the two simulations presented in Figures 6 and 7. If the definition of the entropy is correct, one can expect the entropy to be greater for Figure 6 than for Figure 7 since the former contains more lost ants wondering around the trail network. A graph of the normalized entropy for both Figures has been computed using (15) and is presented in Figure 8. As expected, the entropy of Figure 6 is greater than that of Figure 7. This is consistent with what can be seen from the screen shots — in Figure 6 there are lots of lost ants wondering around the trail network while in Figure 7 almost all the ants are on trails.

One might have noticed that the values of F and L presented for my simulation in Figures 6 and 7 are not in agreement with the ones from [7]. This is due to mistakes in the original code that were corrected in my simulation. The mistakes and their effect on the results are discussed at length in my Advance Computational Physics final project [1]. Please refer to it for more information.

6 Conclusion: Cellular Automata as an Extension of Statistical Mechanics

It has been seen that statistical mechanics and CA have a lot in common. In fact, it has been shown that a CA can be described in statistical mechanics terms and that CA are thus a great tool for modeling statistical mechanic problems. CA also allow one to simulate systems that are not possible to study using statistical mechanics. Systems, such as MASs — systems for which the interactions between the different constituting agents cannot be neglected — can very rarely be solved using a statistical mechanic approach. Those systems, however, are perfect candidates for CA modeling.

Computer simulations such as CA have proven extremely useful in modeling those kinds of systems and

usually give very acceptable results. The beauty of CA is that with a minimum of simple local rules, one can reproduce most behaviors exhibited by the highly complex system being modeled. The level of interactions involved in MASs is usually quite well described by CA. In fact, it has been shown that CA seem like a very promising approach to studying and understanding systems where the interactions between the various agents are the key to the complex behaviour exhibited.

What one can hope to do in the future is to uncover the secrets of MASs. Through the simulation of many MASs, using CA, one can maybe be able to gain intuition as to how a particular set of local rule will influence the global dynamics of a system. The ultimate goal would be to be able to deduce global dynamical equations for the system given its local rules. Until such a formalism exists, CA remains a wonderful tool for the study of such systems. Its increasing popularity in a wide variety of domains including biology, physics, chemistry, computer science, engineering and social sciences is certainly a good indication of its very expandable use.

References

- [1] C. Beauchemin. A visualization program for the study of multi-agent systems. 2002. Final Project in PHYS699: Advance Computational Physics. FIXME.
- [2] E. Bonabeau, G. Théraulaz, and J.-L. Deneubourg. Group and mass recruitment in ant colonies: the influence of contact rates. *J. theor. Biol.*, 195:157–166, 1998.
- [3] J.-L. Deneubourg, S. Aron, S. Goss, J. M. Pasteels, and G. Duernick. Random behaviour, amplification processes and number of participants: How they contribute to the foraging properties of ants. *Physica*, 22D:176–186, 1986.
- [4] Harcourt. Ap dictionary of science and technology. Available at: <http://www.harcourt.com/dictionary>.
- [5] C. E. Hecht. *Statistical Thermodynamics and Kinetic Theory*. W.H. Freeman and Company, New York, 1990.
- [6] R. H. Landau and M. J. Pàez. *Computational Physics: Problem Solving with Computers*. Wiley-Interscience, New York, 1997.
- [7] J. Watmough and L. Edelstein-Keshet. Modelling the formation of trail networks by foraging ants. *J. theor. Biol.*, 176, 1995.
- [8] S. Wolfram. Statistical mechanics of cellular automata. *Reviews of Modern Physics*, 55, 1983. Available at: <http://www.stephenwolfram.com/publications/articles/ca/83-statistical/>.
- [9] S. Wolfram. Cellular automata as models of complexity. *Nature*, 311, 1984. Available at: <http://www.stephenwolfram.com/publications/articles/ca/84-cellular/>.
- [10] S. Wolfram. Cellular automata. *Los Alamos Science*, 9, Fall 1983. Available at: <http://www.stephenwolfram.com/publications/articles/ca/83-cellular/>.