A Nonlinear Multi-agent System designed for Swarm Intelligence : the Logistic MAS

Rodolphe Charrier, Christine Bourjot, François Charpillet

A Nonlinear Multi-agent System designed for Swarm Intelligence: the Logistic MAS

Rodolphe Charrier
LORIA - Nancy University
Nancy, France
Rodolphe.Charrier@loria.fr

Christine Bourjot
LORIA - Nancy University
Nancy, France
Christine.Bourjot@loria.fr

Francois Charpillet
LORIA - INRIA Lorraine
Nancy, France
Francois.Charpillet@loria.fr

Abstract

Ant algorithms and flocking algorithms are the two main programming paradigms in swarm intelligence. They are built on stochastic models, widely used in optimization problems. However, though this modeling leads to high-performance algorithms, some mechanisms, like the symmetry break in ant decision, are still not well understood at the local ant level. Moreover, there is currently no modeling approach which joins the two paradigms. This paper proposes an entirely novel approach to the mathematical foundations of swarm algorithms: contrary to the current stochastic approaches, we show that an alternative deterministic model exists, which has its origin in deterministic chaos theory. We establish a reactive multi-agent system, based on logistic nonlinear decision maps, and designed according to the influence-reaction scheme. The rewriting of the decision functions leads to a new way of understanding the swarm phenomena in terms of state synchronization, and enables the analysis of their convergence behavior through bifurcation diagrams. We apply our approach on two concrete examples of each algorithm class, in order to demonstrate its general applicability.

1 Introduction

Bio-inspired algorithms are increasingly important in distributed AI, giving rise to new paradigms and questioning conventional artificial intelligence approaches. Intelligence is no more considered as an individual characteristic only, but also as a swarm emergent phenomenon. This is the rooting principle of the so-called swarm intelligence. The challenge is to understand its mechanisms so as to build algorithms which will artificially reproduce this massively distributed intelligence and control its emergence to a certain extent. This field of research is quite recent, but specially active and prolific [3].

One of the major challenges of this kind of algorithms is to produce self-organization within the system, that is to find a transition from an uncoordinated state to a stable or meta-stable one. The question is therefore how to correlate individual entity behavior to give rise to a collective coordination without any global controller, and under the constraint of individual limited perceptions.

Swarm algorithms have many applications especially in optimization problems : notably, Ant Colony Optimization (ACO) [6] and Particle Swarm Optimization (PSO) [11], two major examples. The former takes inspiration from real ant colonies, and the latter from flocking birds and particle physics models. One can point out the similarity of the scientific genesis in both cases: at the beginning, biological observations lead to mathematical formulations based either on stochastic processes and statistical measures [5] for the ants behavior, or on deterministic rules for the flocking behavior [17]. In the latter case, birds are replaced by particles in the PSO developments, and deterministic rules are mixed with stochastic variables. However, these two models are quite different, and no work to our knowledge tends to make them join together. Our assumption in this paper is that an abstract and generic model exists that unifies these two approaches.

Contrary to previous approaches, our approach is a theoretical one, due to the origin of the model. We start with an abstract model, namely the nonlinear Coupled Map Lattice (CML), which has its origin in deterministic chaos theory. Questioning this field leads us to propose a derived model which is a reactive situated Multi-Agent System (MAS) called Logistic Multi-Agent System (LMAS). In this paper, the LMAS will prove to be able to implement the two main classes of swarm intelligence phenomena in a unified theoretical formulation. Compared to the previous approaches, it radically changes the way of interpreting the involved mechanisms by connecting them to state synchronization processes, namely a universal concept in nonlinear sciences [16]. The LMAS changes also the way of visualizing the dynamics using bifurcation diagrams.

To begin with, we explain the LMAS in section 2; we then implement it on a basic flocking problem in section 3 which is an example of a self-organized system, and -in a deeper way- on the ant binary bridge experiment in section 4, which is the origin of ant algorithms and an example of self-adaptive system. These concrete problems demonstrate the main aspects of our model.

2 Logistic MAS Conception

2.1 Rooting principles

The logistic MAS (LMAS) takes its inspiration from the coupled map lattice (CML) models used by physicists for studying spatiotemporal chaos phenomena, in particular in the field of hydrodynamics (simulation of turbulent flows) or condensed matter physics. Currently these computational models are of great interest in theoretical physics. We are especially interested in CML using nonlinear quadratic maps, namely logistic maps. The physicist Kaneko have widely explored these types of CML since 80's [10]. He has also intended to extend its use to the general study of complex systems. The next three subsections introduce the CML, the logistic map features, and finally how and why the logistic MAS derives from the CML.

2.1.1 Definition of a logistic CML

A CML is a discrete time and space computation model in which states take their values in a continuous domain. It is sometimes considered as a cellular automaton with continuous states. A minimal CML based on the local map f and a nearest-neighbors diffusive coupling is expressed in the usual form given by :

$$x_i(n+1) = (1-\epsilon)f(x_i(n)) + \frac{\epsilon}{N(V_i)} \sum_{j \in V_i} f(x_j(n)) \quad (1)$$

where $x_i(n)$ is the state variable in site i at time n, ϵ is the diffusive coupling coefficient, V_i is the fixed neighborhood of site i which includes $N(V_i)$ other sites. When f is the classical nonlinear parametric logistic map, it is defined on the interval $[0, 1]$ by the recursive relation :

$$x(n+1) = f(x(n)) = 4a\, x(n)(1 - x(n)) = f^{n+1}(x(0)) \quad (2)$$

If the parameter $a \in [0, 1]$, then $[0, 1]$ is invariant through f. This nonlinear map is completely described by its bifurcation diagram, which is shortly explained in the next section. Nevertheless, after many iterations (about thousands) the resulting CML generates a wide range of behaviors from a full synchronization of cell states to a full chaotic and desynchronized behavior of cell states according to the ϵ and a parameters values. We let the interested reader refer to the bibliography for further information about this subject.

2.1.2 Some features on the logistic map

The logistic map is a well known polynomial mapping, very easy to compute, but with very complex outcomes. It comes from a demographic model created by Pierre François Verhulst, and was popularized in 1976 by the biologist Robert May. The logistic map has been widely studied in its standard form (2), where a is the control parameter. This form is derived from the population equation which is a macroscopic formulation. Nevertheless, this map, as well as its conjugated forms (deduced from the original one by a variable transformation), are widely studied and used nowadays for their intrinsic mathematical properties, independently of their population aspect, notably in the CML studies. The figure 1 shows the bifurcation diagrams of two conjugated logistic maps. They have been calculated for 500 iterations with 500 samples in the interval $[0, 1]$. Although the bifurcation diagram differs from one form to another, the global qualitative behavior is the same. The quantitative behavior follows some universal laws (like for example the definition of Feigenbaum constants) [19]. Bifurcation diagrams show $x(\infty) = f^\infty(x(0))$ according to the control parameter: as a decreases from 1 to 0, the map evolves respectively from a completely chaotic behavior (densely dark bands) to a very precise one, associated to a single fixed point. This specific behavior from complete randomness to complete order under the influence of a single parameter is very useful, and semantically rich. We will exploit these mathematical properties as a generator of agent behavior, with the first map for the flocking modeling, and the second one for the ant algorithm.

2.1.3 Genesis of the logistic MAS

We have to point out some characteristics of the CML approach, which differs from an agent-based approach designed for swarm intelligence. On the one hand, the CML is a field approach where cells are fixed in a given topology, as artificial neural networks are, whereas the agent-based approach is similar to a particle approach, where entities may be mobile in a independent space called the environment. Consequently, the neighborhood of agents depends on time and space. On the other hand, the CML model is designed as a closed system, whereas a multi-agent system is often an open system which has to adapt to external influences mediated by the environment. The induced changes are listed below:

- the transformation of the CML cells into reactive agents which can move in space implies the encapsulation of the coupling and control parameters inside the agent. These parameters become therefore local internal variables, in contrary to their global aspect in CML. A logistic map governs as well agent decision making.

(a) Map 1 : $x_{n+1} = 4ax_n(1 - x_n)$

(b) Map 2 : $x_{n+1} = 1 - a(1 - 2x_n^2)$
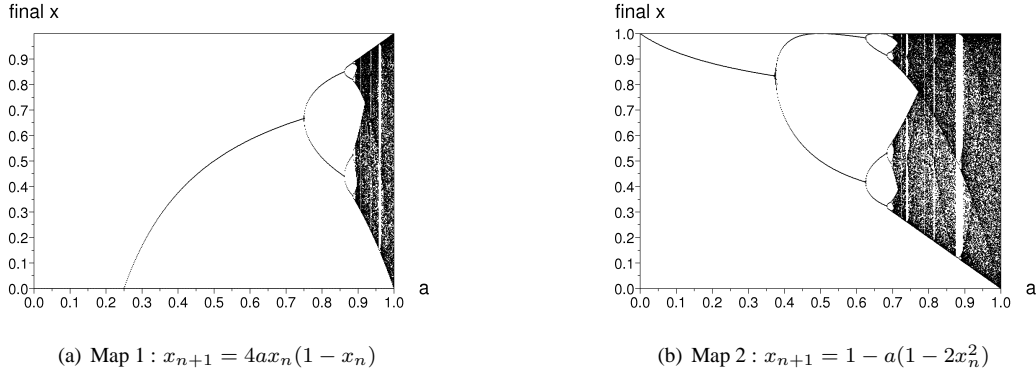
Figure 1. Bifurcation diagrams over $[0, 1]$ after 500 iterations

- the open aspect of the system produces a distinction between entities and their environment, and increases the importance of the adaptation processes and the flows of exchanged data. This implies the dependence of agent state variables on environment perceptions.

Although these differences, many variations on the CML model are close to our case studies, notably the globally randomly coupled map lattice [12], which can be considered as an approximate situation of moving cells. This assumption is not the current issue of this paper, but this specific CML provides theoretical and numerical useful results we think available for LMAS. In addition, numerical simulations are quite a common procedure in CML studies, since mathematical proofs are hard to do in most cases. So are LMAS studies, as the two concrete swarm applications will show it in this paper. To conclude, the LMAS model has a lot to share with the CML one, and can take advantage of existing results on CML.

However, contrary to CML, the concepts of MAS hold an appropriate semantics and structure for handling swarm intelligence. This paper is devoted to present the LMAS and to show its potential.

2.2 Mathematical formalism

The logistic multi-agent system (LMAS) belongs to the reactive MAS class of systems. We take inspiration from the influence-reaction scheme of Ferber-Müller [8] to describe the LMAS.

2.2.1 Global system design

The LMAS is composed of a set A of N_a agents situated in a discrete space called the environment, denoted Env, which is the medium of all the interactions between agents. Our interaction model is indeed an entirely indirect one. We denote by $t \in \mathbb{N}$ the time step variable. Let us identify some

usual definitions: $s_i(t)$ is the state of agent i, and $\sigma(t)$ the state of the environment at time t. The state transition equation of the complete system is resumed by the following equation system:

$$\begin{cases} s_i(t+1) & = F(s_i(t), \sigma(t)) \; \forall i \in \{1, \cdots, N_a\} \\ \sigma(t+1) & = G(\sigma(t), s_1(t+1), \cdots, s_N(t+1)) \end{cases}$$
(3)

The first equation expresses the fact that agents perceive their environment before changing their internal state; the second one expresses the change of the environment state through the combined influences of agents derived from their updated states. The explicit coupling between these two parts of the system makes a natural distinction between the local and the global level and makes clear the computation schedule.

2.2.2 The logistic agent model

The logistic agent is a reactive agent whose internal behavior is governed by a logistic map (fig.1). So as to simplify the description of the involved mechanisms, and because we do not need more for the present, we consider only scalar quantities. But it is obvious that the same design principles apply in a vector space. The agent internal state s is a tuple of the following variables: $s = \langle x, a, \epsilon \rangle$.

Let D denote the interval $[0, 1] \subset \mathbb{R}$.

- $x \in D$ is the decision variable for the agent to perform some actions

- $a \in D$ is the internal control variable, because it operates as the control parameter in the agent logistic map

- $\epsilon \in D$ is a coupling variable for an agent with other agents in its neighborhood; it is similar to the diffusing coupling parameter in CML modeling

The internal state thus takes its values in a continuous domain. D stays invariant through the used map. The agent

state transition is expressed in the equation system:

$$\begin{cases} a(t+1) &= F_a\left(\sigma(t)\right) \\ \epsilon(t+1) &= F_\epsilon\left(\sigma(t)\right) \\ x(t+1) &= F_x\left(x(t), \sigma(t), a(t+1), \epsilon(t+1)\right) \end{cases} \quad (4)$$

This system makes clear how state variables are interwoven and shows the computation steps. $\sigma(t)$ appears explicitly in the equations because it is the source of the agent perceptions. One can notice that a and ϵ transitions arise only from agent perceptions on the environment expressed by the σ dependency.

Let us expand the transition function F_x for the x variable, which is a compound of several operators: one form of the $a(t+1)$-parameter logistic map f_a, an $\epsilon(t+1)$-coupling operator I_ϵ, and a perception function p. This is summarized in the formula:

$$\begin{aligned} F_x\left(x, \sigma, a, \epsilon\right) &= f_a\left(I_\epsilon(x, p(\sigma))\right) \\ \text{where } \forall(y, z) &\in D^2, I_\epsilon(y, z) = (1 - \epsilon)y + \epsilon z \end{aligned} \quad (5)$$

If we write down $u(t) = I_\epsilon(x(t), p(\sigma(t)))$ and choose the logistic map defined in equation (2), the transition equation for the internal x variable becomes:

$$\begin{cases} u(t) = (1 - F_\epsilon\left(\sigma(t)\right))x(t) + F_\epsilon\left(\sigma(t)\right)p(\sigma(t)) \\ x(t+1) = 4F_a\left(\sigma(t)\right)u(t)(1 - u(t)) \end{cases} \quad (6)$$

This latter transition equation makes explicit the agent decision making based on environment perceptions. The action step then is the process in which the decision variable x is interpreted and used to effect an action. In the next section, we propose a basic structure for the environment which allows the agents to perceive it.

2.2.3 The environment and its fields structure

Our environment concept is close to the multi-layered one described in [1], where computational fields play an important part. One then defines fields on the environment Env. A field is a mapping from the environment to a numerical domain D_{field}. With regard to our model, this domain is often the interval $D = [0, 1] \subset \mathbb{R}$. As the environment is a discrete space, it can be defined as a set of sites (let K be the number of sites): $Env = \{site_k | 0 \leq k \leq K\}$.

A field maps therefore a site k of Env to D_{field}:

$$\begin{aligned} field: \quad Env &\rightarrow D_{field} \\ site_k &\rightarrow field(site_k) \end{aligned}$$

The state σ of the environment is defined as the tuple of all existing fields at time t. We are now able to define the agent perception processes.

2.2.4 Agent perception and action principles

In our model, agents perceive some fields of the environment. An agent perception function thus concerns a particular field and achieves a mapping of this field to the agent's state definition domain, namely D. An agent perception applies to a local neighborhood because of its limited capabilities. Perception functions depend on the handled problem and have to be specified after the environment structure has been completely defined.

The environment update concerns the second equation of the global system (3) and occurs after the update of each agent internal state. In particular, agent actions generate some modifications in the environment. They are of two main types. The first one consists in environment fields modifications: an agent can change values of a field at its current location. The second one is the agent location modification: an agent moves relatively to its current location. This latter information is stored as well in a dedicated environment field.
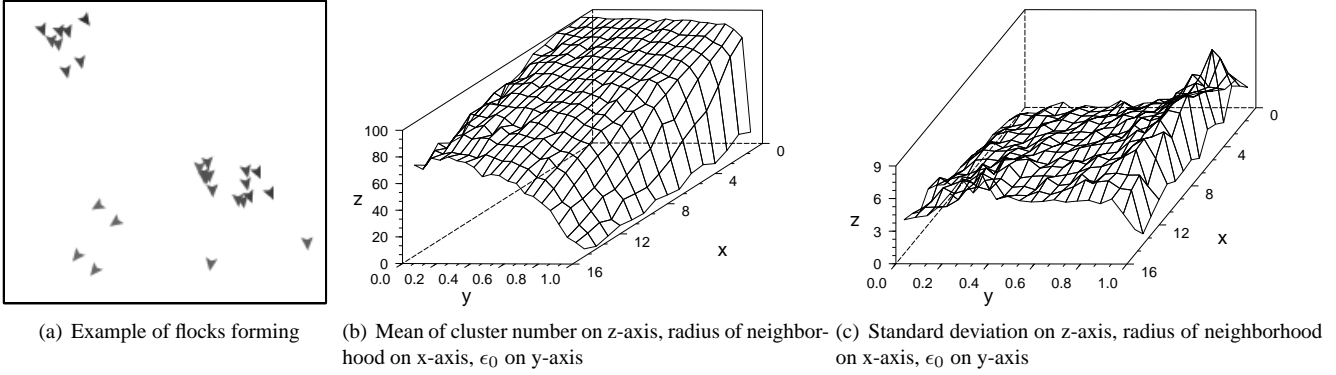
The model structure we have just described is abstract and generic. In the next sections, we will show how it is implemented to a basic flocking instance, and afterward to an ant algorithm problem, with two different ways for analyzing it. This enables to make clear the involved mechanisms in each case.

3 Flocks modeling with LMAS

3.1 Introduction on flocking algorithms

Reynolds proposed in [17] an algorithm to simulate the phenomenon of flocking birds which is considered as an emergent phenomenon example in complex systems. The algorithm principles are based on three deterministic behavioral rules: maintain a minimum distance from others, match velocities with others in its neighborhood, and move toward the perceived center of mass in its neighborhood. Since that time, many other algorithms have been devised to improve its performances, and stochastic versions arose so as to perform optimization problems. This led to the field of particle swarm optimization [11].

A particular interesting instance is the simulation of moving human crowds presented in [18]. The x and y spatial coordinates are computed by means of separate logistic maps. This allows to control the whole crowd with the single control parameter of the map. This parameter is used as a global one, and the resulting behavior can thus not be self-organized. Moreover, there is no coupling factor between agents. The aim of the authors was to achieve a credible virtual simulation, but this specific way of using logistic maps has to our knowledge not been more explored nor theorized. This however is precisely our goal as the following

(a) Example of flocks forming

(b) Mean of cluster number on z-axis, radius of neighborhood on x-axis, ϵ_0 on y-axis

(c) Standard deviation on z-axis, radius of neighborhood on x-axis, ϵ_0 on y-axis

Figure 2. Cluster number anlysis according to the coupling coefficient and to the neighborhood radius with $N = 100$ agents

will demonstrate it.

3.2 A basic flocking model with LMAS

In our context, flocking means the way an agent population gets self-organized into clusters of similar agent behaviors. We will not yet study the stability of such clusters but their distribution according to time and agent perception capacities. We present here the most simple implementation of the LMAS for flocking. Let us list its specifications:

- Agent state: the state of the agent i at time t is expressed in the following tuple: $s(t) = \langle x_i(t), a_i, \epsilon \rangle$.
  The x variable depends on the considered agent and on time. The a variable is determined randomly as a constant value in $[0, 1]$ according to a uniform distribution; it depends on the considered agent. The ϵ variable remains here a constant factor independent of the considered agent. This case proceeds from a perception of a constant field. Finally, the logistic map is the one which corresponds to the first bifurcation diagram 1(a).

- Environment and fields : the environment is a 2D discrete torus with three fields denoted X, N, and E storing respectively the cumulated x variables of each agent on a given site k at time t, the cumulated number of agents on site k at time t, and the ϵ-field. Let us notice that several agents can be located in the same site at the same time, and that the environment has no memory; this means that these fields equal zero before the environment update occurs at time t. Their definitions are expressed in the following formula on a site

k at time t (an agent is indexed by i):

$$N_k(t) \quad = \quad \sum_{i \text{ on } k} 1 \qquad (7)$$

$$X_k(t) \quad = \quad \sum_{i \text{ on } k} x_i(t) \qquad (8)$$

$$E_k(t) \quad = \quad \epsilon_0 \qquad (9)$$

ϵ_0 is given as a constant in the interval $[0, 1]$.

- The perception function p_i^x of an agent i corresponds to the mean of the X field over a neighborhood denoted $V(i)$ centered on the site where the agent i is located. The perception function p_i^ϵ is associated to the ϵ-field. This is expressed by the formula:

$$p_i^x(t) \quad = \quad \frac{\sum_{k \in V_i} X_k(t)}{\sum_{k \in V_i} N_k(t)} \qquad (10)$$

$$p_i^\epsilon(t) \quad = \quad E_{k(i)}(t) = \epsilon_0 \qquad (11)$$

It is easy to verify that $p_i(t)$ always belongs to $[0, 1]$ since it operates as a weighted mean on variables belonging to the same interval.

- The internal state update is described in (6)

- Moving action: since velocity is defined as a 2D vector, the speed or the magnitude of velocity remains here a constant whereas we modify its direction according to the updated $x(t + 1)$; if α denotes this direction, then we write down $\alpha = 360 \times x$ to get α in degrees for example. We denote by k' the agent new site location.

- Updating action: at first, the environment fields are reset. All agents then update them: an agent i modifies

the fields on its new site location k' according to the following equations:

$$N_{k'}(t+1) = N_{k'}(t+1) + 1$$
$$X_{k'}(t+1) = X_{k'}(t+1) + x_i(t+1)$$

3.3 Results and discussion

A computer simulation with $N = 100$ agents in a 30×30 torus environment shows in fig.2(a) that groups of agents arise to form flocks (a small part of the whole is shown). This flocking phenomenon increases as ϵ tends to 1. This latter issue is made clear in the chart in fig.2(b). Let us explain first how it is build: the measure used here consists in counting the number of agent clusters formed at each time step. Two agents belong to the same cluster if and only if their internal x differ from a given threshold and they belong to the same neighborhood. The threshold is set here to 0.01 which corresponds to one percent of the maximum value of x. Let $N_c(t)$ denote this number of clusters. The measure average out this on 1000 consecutive time steps from 1000 to 2000 after the transient period, expressed by:

$$\bar{N}_c = \frac{1}{1000} \sum_{t=1001}^{t=2000} N_c(t) \qquad (12)$$

The chart on fig.2(b) shows the variations of $\bar{N}_c$ according to $\epsilon \in [0,1]$ and according to the neighborhood radius belonging to $[1, 15]$. The maximum neighborhood radius is the half of the environment size. The diagram on fig.2(c) shows the standard deviation on the same series of numbers. Let us now consider the issues of those two diagrams. The flock forming is non-existent, that is $\bar{N}_c \approx 100$, ϵ and the neighborhood radius get closer respectively to 0 and 1; it means that each agent belongs to a cluster formed by itself. As asserted before, The flock forming is more and more important, that is $\bar{N}_c$ is small, as ϵ and the radius get closer respectively to 1 and 15. Let us notice the shape of the chart : a very sharp decreasing arises when ϵ is close to 1 and the neighborhood radius small. It gets smoother as the radius gets bigger. It is reinforced by the fact that the corresponding standard deviation on fig.2(c) is high in this area; that reveals some perturbations in stability for the bigger formed clusters. The initial uniform distribution of the control variable inside the agents is partly the reason of this instability: the bifurcation diagram on fig.1(a) shows that about 10 per cent of the agents has got a chaotic behavior, which implies instability.
These results confirm an intuitive feeling on the flocking phenomenon, "Birds of a feather flock": flocking emerges easier as the radius of interaction and the level of similarities between agents increase. The fundamental change in this mechanism with respect to the classical algorithm, is

that the flocking occurs because of agent state synchronization and not because of appropriate rules. This is indeed a more universal concept.

4 Ant binary bridge modelling with LMAS

4.1 A short ant algorithms survey

The paradigm of ant colony is currently studied and applied a lot in swarm intelligence as well as in other scientific fields [7]. However, dynamical and convergence analysis of ant algorithms [20, 9] are very recent and appear after many years of experimental issues. The stochastic approach succeeds in simulating the global phenomenon but is not sufficient to really explain specific features of ant colony mechanisms, like the symmetry break which occurs in the binary bridge experiment for example, or like the specific involved dynamics of the colony. Merkle's recent work [13] has been a step forward in this regard, but with a deterministic model of ant dynamics.
To analyze these phenomena in a simple way, we resume the modelling of the binary bridge experiment, since this experiment shows the basis of the collective choice mechanism of ant colonies without any optimization aspects. It is also the simplest instance of a graph-based ant algorithm from which all other instances derive. In addition, we will handle the problem with the LMAS model described in section 2.2 and demonstrate this new modeling approach for simulating the binary bridge experiment and providing a new visualization of its dynamics.

4.2 The original binary bridge model

Deneubourg and al. published an experiment together with a mathematical model in [5], explaining the trail following behavior of ants when foraging. They use the argentine ant species which collectively selects a path to a food source. They particularly studied the self-organization process that achieves this goal. More precisely, the experimental protocol consists of a binary bridge A and B, connecting the ant-nest to the food source. Branches have the same distance, so the problem is not to select the shortest path; ants have only to choose one of the branches to get to the food source. Along the way to the food source, and while returning back to the nest, they lay down pheromones. We would like to point out the underlying assumptions : the bridge has got two equal long branches, the pheromone does not evaporate, and the amount of pheromone on a bridge is proportional to the number of ants crossing the bridge. The experiments are of short duration so that the amount of pheromone evaporation is negligible. Finally, ants are supposed to cross the bridge one by one. This leads to the following model [5]:

- A_i (respectively B_i) is the number of ants choosing the branch A (respectively B) after the ith ant crosses the bridge : i acts as the global time step of the algorithm. An amount of pheromone of value 1 is laid down by each ant on the chosen branch.

- P_A (respectively P_B) is the probability the $i + 1$th ant chooses A (respectively B) :

$$P_A = 1 - P_B = \frac{(K + A_i)^n}{(K + A_i)^n + (K + B_i)^n} \quad (13)$$

This formula has been obtained by a statistical analysis on the experimental protocol. The best fit to the experimental data is given with parameter values $K = 20$ and $n = 2$.

This stochastic model produces the expected results: all the ants end up choosing the same branch after a while, even if neither branch is favored *a priori*. This branch is called the winning branch. Moreover it is not a problem of distance optimization since the branches are equally long, but a pheromone optimization.

After this short recall, a question remains unsolved : *what is the cause of the global symmetry break at the local ant level?* And if this question cannot be answered completely, can we model it differently to obtain a better understanding of the phenomena? This question has indeed already been attempted in [15, 2], where the authors study it from a global point of view by using coupled differential population equations. Using such techniques, one succeeds in explaining the symmetry break in ants behavior, while providing a global visualization of the phenomenon. However, this approach is an analytical one, not a constructive one. Our aim is therefore to pursue this approach on the local level of ants and to provide a link between local and global description levels. In short, the major issue for us consists in determining in which way local deterministic mechanisms could produce such a global phenomenon? We will show that our abstract model of logistic MAS is appropriate to achieve this objective with some adapted specifications. This is the topic of the next section.

4.3 Algorithm design

4.3.1 Environment definition

The environment $Env = (V, Ed)$ is a small graph, where V is the set of nodes and Ed the set of edges, composed of only three nodes and two edges. One of the nodes is the nest, and the others correspond to the binary bridge destinations A and B (fig. 3). We define two main fields in the environment: a pheromone field τ [14], and an agent state field X. We will distinguish in our model two types of agent influences on the environment, which are indirect interactions, each relating to one of these fields: a pheromone
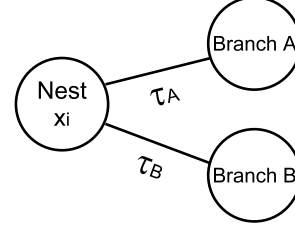

Figure 3. Scheme of the binary bridge

deposit on edges of the graph has a long range interaction effect stored by the field τ, and an internal state deposit on the current node has a short range interaction effect stored by the field X. These fields are defined in the specific case of a finite graph topology by:

$$
\begin{aligned}
X : \quad V &\to D = [0, 1] \\
v_k &\to X_k \quad (14) \\
\tau : \quad Ed &\to \mathbb{R}^+ \\
e_{kl} &\to \tau_{kl} \quad (15)
\end{aligned}
$$

Each node is indexed by an integer and so is X. We denote by X_0 the nest. Each edge is indexed by two integers, and so is τ. Moreover, these fields evolve with time since ants change their value. On the contrary, a third field $Ek = \epsilon_0$ is involved but remains constant like in the flocking case. Agent perception functions depend on these fields.

4.3.2 Agent perception-decision-action loop

We now describe the perception-decision-action scheme of the agent's internal mechanism. In this case, the time step of the algorithm corresponds to the event of one agent crossing the binary bridge.

- Perception : at time t a new agent i, leaving the nest, perceives the environment fields. The perception functions are expressed in the following way for each state variable:

$$
\begin{cases}
p_x(t) = X_0(t) = x_{i-1}(t) \\
p_a(t) = max\{\tau_a, \tau_b\} \\
p_\epsilon(t) = \epsilon_0
\end{cases} \quad (16)
$$

An agent perceives the internal x-value of the preceding agent on the current node (the nest). It perceives the amounts of pheromone by reading the field τ, and by capturing the maximum amount of this field. It selects the edge where the field is maximum as the base edge for its moving decision. Finally, it perceives as well the constant ϵ-field. ϵ_0 may represent the individual communication characteristics of the given ant species.

- Decision making: the transition of the internal state variables (4) is given by the following equations:

$$x_i(t+1) = f_a^k\left((1-\epsilon_0)x_i(t) + \epsilon_0 x_{i-1}(t)\right) \quad (17)$$

$$a_i(t+1) = \frac{\alpha}{\alpha + p_a(t)} \quad (18)$$

The formula (17) describes the individual communication level between agents : if $\epsilon_0 = 0$, no local contact between agents occurs. We are then in a mass recruitment situation as described in [3]: only the pheromone trail can influence the agent. Usual ant algorithms are based on this assumption. However if $\epsilon_0 \neq 0$, individual information exchange occurring in the agent neighborhood is taken into account: $x_{i-1}(t)$, laid down by the preceding agent, is involved in the computation. We use the second form of the logistic map $f_a(x) = 1 - a(1-2x^2)$ (fig.1) with k iterations to get the behavior described in the bifurcation diagram. The a variable depends on time according to the formula (18) in a decreasing way: a varies from 1 to 0 as the amount of pheromone varies from 0 to $+\infty$. The α parameter governs the analytic function form. Finally a threshold enables to make the decision:
if $x_i(t+1) > threshold$ choose the branch with τ_{max} else choose the other.
As we are in a simple case with only two edges connected to the nest node, it is convenient to choose a threshold set to $\frac{1}{2}$. This rule is relative to the branch where the pheromone amount is maximal, since the agents perform here a pheromone amount optimization. The deviation with the highest amount expresses the agent autonomy.

- Action: this corresponds to the update of the environment, that is the update of its two fields according to the following equations:

$$\begin{cases} X_0(t+1) = x_i(t+1) \\ \tau_{chosenedge}(t+1) = \tau_{chosenedge}(t) + x_i(t+1) \end{cases} \quad (19)$$

The pheromone updating is a cumulative process without evaporation. Indeed, the original model for simulating this experiment did not implement any evaporation process because of the short experimental time scale which did not allow evaporation to proceed.

Initial conditions correspond to the experimental ones: the initial amount of pheromone is zero and the initial control parameter is close to 1, it means that randomness is high at first.

4.4 Implementation and results

The resulting algorithm is **Algorithm 1**. The simulation results can be viewed in figure 4. On the same graph, the ratio of ants traversing the branch, averaged over ten runs, is plotted for both the winning and the loosing branch. The mean of the control variable over ten runs is plotted as well. To understand those charts and their associated variations, we must bear the bifurcation diagram of fig.1(b) in mind.

In the three cases, we identify the symmetry break at the point where the winning and loosing curves begin to converge respectively to 1 and 0. In each case, the symmetry break occurs exactly when the control variable a reaches the value 0.5. This result is not very surprising regarding the logistic bifurcation diagram in figure 1(b): before this point, in the area $a > 0.5$, both branches are explored: this implies that the agent system oscillates quite identically between x-values below and above 0.5. After this point, in the area $a < 0.5$, x still oscillates for a while but always above 0.5. In this area, the decision remains invariant and the symmetry break is complete. The ant ratio curves in the graph 4(a) fits well with the experimental data curves[5]. But this comparison remains a qualitative one because of the lack of experimental data. The k-value is currently not significant enough.

The two graphs 4(b) and 4(c) with $\epsilon_0 = 0.5$ are surprising because they only differ in the iteration number of f. One can notice that the graph 4(b) is quite the same as 4(a) with an odd logistic map iteration number, whereas graph 4(c) shows a long term symmetry break with an even iteration number k. The following section demonstrates the impor-

---

Algorithm 1 Binary bridge with LMAS

Initialization
$\tau_A, \tau_B \leftarrow 0$
$N_A, N_B \leftarrow 0$
for ant $i = 1$ to N **do**
    init : $x_0 \leftarrow random(0,1)$
    Perception
    read $\tau_{max} \leftarrow max\{\tau_A, \tau_B\}$
    $index_{max} \leftarrow argmax_{\{A,B\}}\{\tau_A, \tau_B\}$
    $index_{min} \leftarrow argmin_{\{A,B\}}\{\tau_A, \tau_B\}$
    Decision
    $threshold \leftarrow \frac{1}{2}$
    $a \leftarrow \alpha/(\alpha + \tau_{max})$
    $x_i \leftarrow f_a^k(x_0, x_{i-1}, \epsilon_0)$
    if $x_i < threshold$ **then**
        Action 1
        $N_{index_{max}} \leftarrow N_{index_{max}} + 1$
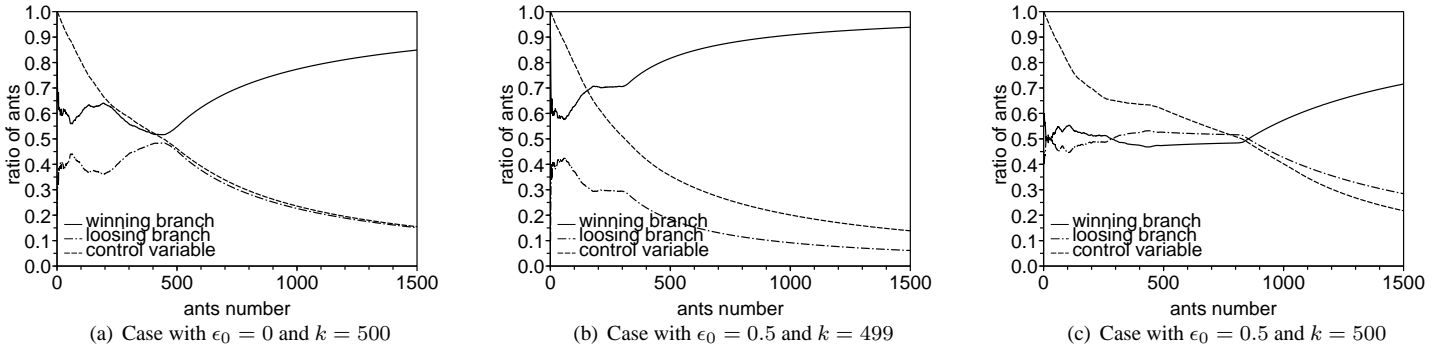 $\tau_{index_{max}} \leftarrow \tau_{index_{max}} + x_i$
    else
        Action 2
        $N_{index_{min}} \leftarrow N_{index_{min}} + 1$
        $\tau_{index_{min}} \leftarrow \tau_{index_{min}} + x_i$
    end if
end for

(a) Case with $\epsilon_0 = 0$ and $k = 500$ (b) Case with $\epsilon_0 = 0.5$ and $k = 499$ (c) Case with $\epsilon_0 = 0.5$ and $k = 500$

Figure 4. Simulation results of the binary bridge : mean curves over 10 runs with 1500 ants

tance of a collective bifurcation diagram building to explain this, and to provide some qualitative convergence analysis.

4.5 Qualitative convergence analysis and discussion

An explanation of the case 4(c) can be given by considering the bifurcation diagram in fig.5. To start with, let us consider how it is made. The graph is not an individual one as it was for the logistic agent, but a collective one: we have plotted every (x, a) pair of values for each agent crossing the bridge at time t. The graph is to read from right to left because a is decreasing in time. If we apply this process on the case 4(b), we obtain a diagram very similar to the individual bifurcation diagram 1(b). In the case 4(c), the achieved diagram fig.5 can be analyzed as follows: when the number of iterations k of each individual logistic map is even, the system only selects the lower branch of the diagram, because of the even oscillation period. But this is not the single cause of the long term symmetry break. One observes that the control variable levels off (fig.4(c)) at the bifurcation cascade points, which delays the symmetry break.

This collective diagram thus provides a very useful tool to follow the whole dynamics, since it gives a qualitative analysis of the global algorithm convergence. As a matter of fact, it shows that the ant algorithm become stabilized as the colony converges to a global state. This global state corresponds to single fixed points, belonging to the left branch of the collective bifurcation diagram. It means therefore that a global agent state synchronization occurs on those fixed points. Contrary to the flocking case, this synchronization proceeds from the pheromone field, which is collectively build by the agents: we are faced with a self-adaptive system. The binary bridge experiment is an appropriate experiment to display this synchronization, and the collective bifurcation diagram is an appropriate tool to point this phenomenon out. This global visualization is made possible
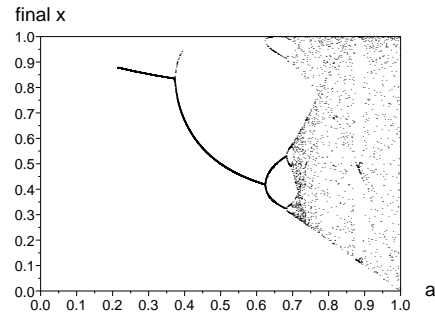
Figure 5. Bifurcation diagram with $N = 1500$ **ants,** $\epsilon_0 = 0.5$ **and** $k = 500$

because of the same map distributed to each agent. One can as well get information about the system, given by the distribution of the a variable at time t. For instance, one can determine whether the collective behavior is focused near a particular a value or widely distributed, how dispersed the agent behaviors are, and how far the system is away from ending fixed points.

5 Final discussion and conclusion

Results showed that the logistic MAS applies itself to both flocking simulation and binary bridge experiment. The LMAS approach is an elegant and relevant way to demonstrate the synchronization process in these two main primordial swarm phenomena, by state coupling in the flocking case, and by a strong decentralized influence on the environment in the ant case. Expressing them in such a common modelling framework, enabled us to make clear the main difference between the two processes, which has never been done to our knowledge. State synchronization argues

for the generality of our model. The choice of the logistic map may nevertheless appear as a limitation of the scope of LMAS. Firstly, some previous developments in CML studies lead us to confirm this choice: cluster synchronization, and thus the flocking phenomenon, would not occur with every nonlinear map. A conjecture in CML studies [4] actually postulates that the map used has to exhibit some periodic windows in the chaotic area of its bifurcation diagram, so as to let cluster synchronization emerge. The logistic map is consistent with this conjecture contrary to the tent map for example. Secondly, the population equation aspect of the logistic map fits well with the swarm phenomena, since these phenomena may be considered as instances of population problems. Consequently, the fact that a piece of this global population effect may be found again as a cause at the local level of entities and within entities, seems totally coherent.

To conclude, this paper presented the logistic agent, which uses a deterministic nonlinear map, namely the logistic map, to govern its own internal state. This agent is part of a logistic multi-agent system supposed to be a generic modeling for swarm intelligence. The local control variable and coupling variable inside the agent are novel features. These variables decentralize control and govern the chaotic level of agent decisions by means of perceptions. The behavior profile according to the control variable is known *a priori* through a bifurcation diagram and it is the same for each agent. This approach provides theoretical tools and visualization tools from the modern dynamical system theory, so as to follow and analyze the dynamics of the whole multi-agent system.

The interpretation of both flocking phenomenon and ant colony behavior in terms of synchronization processes is the most salient contribution of this paper. This approach has to go deeper, theoretically speaking, about cluster stability, global convergence, and nonlinear map choices. The presented LMAS has also to be matched with more applied problems. In this way, we intend to match it with optimization problems like other swarm-based algorithms. We have tested it on some small instances of the symmetric TSP with promising results. LMAS is a promising modeling tool for swarm intelligence.

References

[1] S. Bandini, S. Manzoni, and C. Simone. Dealing with space in multi–agent systems: a model for situated mas. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 1183–1190, New York, NY, USA, 2002. ACM Press.

[2] R. Beckers, J. Deneubourg, S. Goss, and J. Pasteels. Collective decision making through food recruitment. *Insectes sociaux*, 37(3):258–267, 1990.

[3] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence. From Natural to Artificial Systems*. Studies in the Sciences of Complexity. Oxford University Press, New York Oxford, 1999.

[4] M. G. Cosenza and J. Gonzalez. Synchronization and collective behavior in globally coupled logarithmic maps. *Progress of Theoretical Physics*, 100(1):21–38, 1998.

[5] J. Deneubourg, S. Aron, S. Goss, and J. Pasteels. The self-organizing exploratory pattern of the argentine ant. *Insect Behavior*, 3:159–168, 1990.

[6] M. Dorigo, V. Maniezzo, and A. Colorni. The ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics*, 26(1):1–13, 1996.

[7] M. Dorigo and T. Stützle. *Ant Colony Optimization*. The MIT Press, 2004.

[8] J. Ferber and J. Müller. Influences and reactions : A model of situated multiagent systems. In *Proceedings of ICMAS96*, 1996.

[9] W. J. Gutjahr. A graph-based ant system and its convergence. *Future Gener. Comput. Syst.*, 16(9):873–888, 2000.

[10] K. Kaneko. *Theory and applications of coupled map lattices*. John Wiley & Sons, 1993.

[11] J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *Proc. of the IEEE Int. Conf. on Neural Networks*, pages 1942–1948, Piscataway, NJ, 1995. IEEE Service Center.

[12] S. C. Manrubia and A. S. Mikhailov. Mutual synchronization and clustering in randomly coupled chaotic dynamical networks. *Phys. Rev. E*, 60(2):1579–1589, Aug 1999.

[13] D. Merkle and M. Middendorf. Modeling the dynamics of ant colony optimization. *Evolutionary Computation*, 10(3):235–262, 2002.

[14] H. Parunak. Go to the ant: Engineering principles from natural multi-agent systems. *Annals of Operations Research*, 75:69–101, 1997.

[15] J. M. Pasteels, J. L. Deneubourg, and S. Goss. Self-organization mechanisms in ant societies. i: Trail recruitment to newly discovered food sources. *Experientia, Suppl.*, 45:155–175, 1987.

[16] A. Pikovsky, M. Rosenblum, and J. Kurths. *Synchronisation, a universal concept in nonlinear sciences*, volume 12 of *Cambridge Nonlinear Science Series*. Cambridge University Press, 2003.

[17] C. W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics*, 21(4):25–34, 1987.

[18] N. Saiwaki, T. Komatsu, T. Yoshida, and S. Nishida. Automatic generation of moving crowd using chaos model. *IEEE Int. Conference on System, Man and Cybernetics*, 4:3715–3721, 1997.

[19] H. G. Schuster and W. Just. *Deterministic Chaos : An Introduction*. WILEY-VCH Verlag GmbH and Co. KGaA, Weinheim, 2005.

[20] T. Stützle and M. Dorigo. A short convergence proof for a class of ant colony optimization algorithms. *IEEE Trans. Evolutionary Computation*, 6(4):358–365, 2002.