Study of Self-adaptation Mechanisms in a Swarm of Logistic Agents

Rodolphe Charrier, Christine Bourjot, François Charpillet

Study of Self-adaptation Mechanisms in a Swarm of Logistic Agents

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

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

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

Abstract

We are interested in addressing the problem of coordinating a large number of simple agents in order to achieve a given task. Stated in this way, the question leads naturally to the Swarm Intelligence field. In this paper we use a new type of model, directly inspired by Kaneko's coupled map gas model which we have adapted to the multi-agent system paradigm, so as to tackle this generic objective. This model is called a logistic multi-agent system (LMAS): it is composed of reactive situated agents whose individual behavior is governed by a logistic map or more generally a quadratic map. The collective behavior results from couplings between agents and local controls on agents adjusted by local environmental conditions. This way of modelling reveals to enable a wide range of pattern formations and various forms of adaptation to the environment. This paper focuses on the way to design the constitutive mechanisms of LMAS –particularly the perception and action processes– and on the way a self-adaptation process may result from these mechanisms. This study is illustrated with experiments on the predators-prey pursuit problem, in which a set of agents (predators) has to encircle a moving prey. We show that coupling the internal states of agents leads to amplifying the predator aggregation around the prey, whereas altering the internal control variable in each agent through environment perceptions modifies the predator sensitivity to the prey. We finally complete this study by relating the concept of adaptation with concepts of the dynamical system theory: a qualitative dynamical analysis of the capturing process leads to view the prey as a dynamical fixed point of the system.

1. Introduction

The root principle of Swarm Intelligence consists in considering intelligence no more as an individual characteristic only, but also as an emergent phenomenon. This type of artificial intelligence aims at being robust, fault-tolerant and to provide some efficient meta-heuristics to large sets of problems. Furthermore Swarm Inelligence meets intrisically some scalability aspects, since a swarm involve usually from hundred individuals. The challenge is on one hand to understand the involved mechanisms of natural swarms –this constitutes the research program of ethologists for examples [1]–, and on the other hand to design artificial mechanisms –maybe nature-inspired– so as to build algorithms which aims at reproducing this massively distributed intelligence and controling its emergence to a certain extent. Our concern about that research field lies in dynamical approaches of swarm intelligence, viewed as a distributed cognition issue. These type of approaches are usual in the complex system field [2] or as well in the neural network field [3]. The rooting principle of this approach is to use deterministic models based on a dynamical system formulation, in order to take advantage of the theoretical tools and analysis methods of this mathematical theory.

The originality of the model we use to explore swarm phenomena lies in the way it takes inspiration from the study of the Coupled Map Lattices (CML) family of computational models, notably the ones involving logistic maps [4] in non-linear sciences. This model is a reactive multi-agent system (MAS) called Logistic Multi-Agent System (LMAS) [5]. In this paper we explore both organizing and adaptive capabilities of the LMAS. To achieve this goal, we study the way to design the variation functions associated with each state variable of the agents. These functions are divided in three groups: perception functions, decision functions and action functions, which fits exactly with the linked perception-decision-action loop. The term "function" is justified by the fact that we deal with sets of real numbers. Modifying these local functions lead to change the global characteristic of the system in terms of pattern formation. We apply these principles firstly to a flocking case to show the effect of the coupling on the formation of groups, and secondly to a case of predator-prey pursuit which involves a "swarm" of predators flocking round one or many preys. This latter case focuses on adaptation processes and is difficult since a large number of moving agents –with limited perceptions and without direct communication– have to flock round the same moving location (the prey) in a decentralized way. Beyond this application and the important feature of that study lies in the link and interpretation we can state in the dynamical system field: we show that the mechanism design is to be dynamically interpreted as the construction of a fixed point with a limited basin of attraction in the phase space of the agent.

The paper is organized as follows. The first section recalls what a CML is and how the LMAS derives from it. We then recall as well in Section 3.1 how a flocking formation may appear within a LMAS, as an illustration of the self-organization processes occurring through coupling in this MAS. The next sections show the effects of specific changes in the perception functions of the agents to get adaptation capabilities. We apply this model to a predators-prey pursuit to prove the efficiency of the approach. We finally discuss the results regarding the dynamical system field and we link the chosen application with the field of optimization problems.

2. The LMAS model

This section begins with the description of the model of coupled map lattice, which is the underlying mathematical basis of the LMAS model. The global coupled instance of the model is presented shortly as well as the main mathematical results on it. We then show how to transform it into a multi-agent system, in which agents are mobile in space. The second part of the section presents the resulting general formulation of the LMAS.

2.1. A swarm derived from CML

CML models are used by physicists to study spatiotemporal chaos phenomena in the field of hydrodynamics (simulation of turbulent flows) or condensed matter physics, and more generally in theoretical physics, as a computational model. Our interest focuses on CML using nonlinear quadratic maps, like the logistic map. These types of CML have been widely explored by Kaneko since the 80's [6].

A CML is a discrete time and space cellular model, in which cells are located in a fixed connection topology and in which cell states x take values in a continuous domain –a CML is often considered as a cellular automaton with continuous states–. First studies focused on simple one-dimensional CML based on quadratic maps and a local diffusive coupling with the two nearest neighbors. A "mean-field" type instance of CML called the Globally Coupled Map (GCM) consists in coupling each cell with all other cells. A GCM can be expressed by the following master equation:

$$x_i(t+1) = (1-\epsilon)f(x_i(t)) + \frac{\epsilon}{N}\sum_{j=1}^{N} f\left(x_j(t)\right) \quad (1)$$

where $x_i(t)$ is the state variable of the cell on site i at time t and ϵ is the diffusive coupling coefficient and N is the total site number in the lattice. CML and GCM have been widely studied when f is the well-known logistic map usually defined on the interval $[0,1]$ ($[0,1]$ is invariant through f) by the following recurrent equation:
$x(t+1) = f(x(t)) = 4\ a\ x(t)(1-x(t)) = f^{t+1}(x(0))$
where $a \in [0,1]$ denotes the control parameter of f—we will use the notation f_a as well when we want to stress the role of the parameter a—. The control parameter a governs the type of series the map can generate: this map is known to generate chaotic series —that is pseudo-random series— in particular if $a=1$, or to converge to some fixed points if $a < \frac{3}{4}$ or to periodic cycles and chaos if $a > \frac{3}{4}$. However other quadratic maps may be used as well and have the same qualitative dynamical properties since they are conjugated with the original logistic map, i.e. there exists a homeomorphic transformation between the two maps. In particular in this paper we will use the following quadratic map because of its particular bifurcation diagram (on Fig.1)[1]:

$$x(t+1) = f(x(t)) = a\ (2\ x(t) - 1)^2 \quad (2)$$

Let us return to the GCM model properties. A GCM displays a full synchronization phenomenon which corresponds by definition to a global stable state where all sites have the same $x(t)$ series from a given time t_0 on. This phenomenon occurs from a particular threshold of the coupling parameter ϵ on. When coupling is not symmetric anymore, or when coupling becomes local or random, full synchronization turns into many different synchronization clusters according to the chosen map. Each synchronized cluster corresponds to a set of units which produces the same sequence of x-values. Quadratic maps belong precisely to the class of maps which enables clustering situations to emerge. This is a crucial point for arguing the choice of this kind of map for modeling swarm phenomena, particularly in the flocking modeling: a spatial flock of agents corresponds in that modeling to a cluster of almost-synchronised agents (see section 3.1).

Shibata and Kaneko anticipated the evolution of the CML model so as to deal with the Swarm Intelligence field. They have proposed in [7] a specific CML instance called the Coupled Map Gas (CMG). A CMG is a CML-like model in which entities are "motile" –that is capable of motion– and therefore free to move on the lattice. Local couplings between entities may also depend on time in that case, since two cells may be too far to interact. This evolution means that the model goes from a field design in the CML case to a particle design in CMG. The authors of [7] assert that the CML gas may have some Swarm Intelligence applications without really exploring concrete cases: they kept an abstract formulation close to CML and studied a specific case where a gradient-type force is responsible for the cell moving behavior. But despite its promising potential, the CMG has not been investigated further. We have considered that CMG

1. this diagram plots $x(\infty) = f^{\infty}(x(0))$ as a function of the control parameter a

lays the foundation of a multi-agent system as a grounding calculus model. Moreover the MAS paradigm is more appropriate to deal with swarm intelligence in our opinion: we intend to divide clearly the model in two parts –agents and environment– interacting according to the "influence-reaction" scheme [8] and to guide the mechanism design according to perception-decision-action loops. To summarize our approach, we may generalize the formula 1 into the following master equation where i indexes a moving agent:

$$x_i(t+1) = (1-\epsilon(t))f_{a(t)}(x_i(t)) + \frac{\epsilon(t)}{N(V_i(t))}\sum_{j\in V_i} f_{a(t)}(x_j(t)) \quad (3)$$

The main changes we introduce in the CML model are the following ones:

- All parameters may now depend on time because of perceptions in the environment. Consequently, their mathematical kind will change from "parameters" to "variables" in our modeling.
- Secondly the mean field is reduced to the local neighborhood $V_i(t)$ of an agent i containing $N(V_i(t))$ other agents. This master equation gives the most generic formulation of the computational model we use.

The next section describes the LMAS model, starting from an influence-reaction formulation.

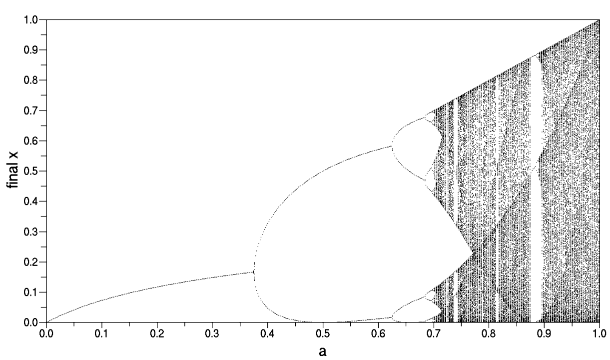


Figure 1. Bifurcation diagram of the quadratic map $x_{t+1} = a\ (2\ x_t - 1)^2$, calculated with 500 iterations on 500 samples of the interval $[0, 1]$

2.2. Formal writing of LMAS

The LMAS is composed of N reactive agents called logistic agents, because they contain a logistic or quadratic map as internal decision function. They correspond to the moving cells in preceding CMG. The environment is denoted by Env. It is a continuous or discrete space and constitutes the medium of all interactions between agents. The interaction model is indeed an entirely indirect one, which implies that all exchanged information is stored during a time step in some layer of the environment which we call a field. A field is defined as a map $\mathcal{F}$ from Env to $\mathbb{R}$:
$\mathcal{F} : Env \rightarrow \mathbb{R}$
A field may vary with time and may be discretized –a field is implemented on the computer as a 2D-array of data in all the cases we handle in this paper–. The state σ of the environment is therefore defined as the tuple of all existing fields at time t –we denote by $t \in \mathbb{N}$ the time step variable–:

$$\sigma(t) = \langle \mathcal{F}(t), \mathcal{G}(t), \ldots \rangle \quad (4)$$

The state transition equation system for the most generic case derives from the influence-reaction scheme of Ferber and Müller [8]. It is described by a dynamical system formalism and may be summarized by the following equation system –this is a simplified version of the original formulation of Ferber and Müller– where $s_i(t)$ denotes the state of the agent i at time t.:

$$\begin{cases} s_i(t+1) & = F(s_i(t), \sigma(t))\ \forall i \in \{1, \cdots, N\} \\ \sigma(t+1) & = G(\sigma(t), s_1(t+1), \cdots, s_N(t+1)) \end{cases} \quad (5)$$

The first equation expresses the fact that agents perceive their environment before changing their internal state, that is the "reaction" phase of the environment. The second one expresses the change of the environment state through the combined influences of agents: the agents perform actions derived from their updated states. The explicit coupling between these two parts makes a natural distinction between local and global levels and makes clear the entire computation schedule.

2.2.1. The Logistic Agent. The logistic agent is a reactive agent whose internal behavior is governed by a logistic or quadratic map. As mentioned in the comment of the formula 3, the internal state s of a logistic agent is composed of three variables as a general rule: $s = \langle x, a, \epsilon \rangle$, where:

- x is the decision variable which governs the action process: the action processes will depend only on x
- a is the control variable which is the control parameter of the internal logistic map: the control variable governs the dynamics type of the agent, that is it selects the fixed point, cyclic or chaotic mode of the agent according to its value. The chaotic mode enables the agent to explore an environment, the cyclic mode restricts the dynamics to a confined area, and the fixed point mode implies a constant behavior.
- ϵ is the coupling variable for each agent which make the agent synchronize with others.

The control as well as the coupling variable may have the same value for the whole agent set, or be specific to each agent. All these three variable values belong to the real domain $D = [0, 1]$, so $s \subset D^3$. In order to simplify the formulations and as ϵ will not be an individual variable but always a global constant parameter in this paper, we chose to omit ϵ and to remove it from the agent's state variables. But the same kind of mechanisms might be involved with ϵ. The computation process for the agent is clearly divided in three parts –perception-decision-action– in the MAS approach, which contributes to explain the data flow. The environment state $\sigma(t)$ is the source of the agent perceptions through the internal state variables of the logistic agent. Perception and action functions constitutes the way for the agent to tranform the values from Env to D^3 –which is the space of agent states– and vice versa. An agent's perception function may concern only a particular field of the environment and achieves a computation with the data stored in this field. The perception is restricted to the local agent neighborhood due to the limited agent's perception capabilities.

The decision process consists in updating the decision variable x. During the action phase, the decision variable x is interpreted and used to perform actions like moving and updating locally the state of the environment, that is some field values. The global transition from the time step t to the next time step regarding the agent state is summarized in the following expression:

$$\begin{cases} x(t+1) & = F^x\left(x(t), \sigma(t)\right) \\ a(t+1) & = F^a\left(a(t), \sigma(t)\right) \end{cases} \tag{6}$$

These equations define the transition for the two components of the agent state (remind ϵ is not considered as a state variable anymore). One can notice that a does not depend on x but only on agent perceptions expressed by the function F^a of σ only. Their explicit formulation depends on the problem. When agents are not adaptive to the environment, σ does not play any role.

Let us expand the main transition function F^x for the agent, which expresses the ϵ-coupling between the logistic map f with parameter a, and a perception function p^x specific to the x component of the agent state. F^x is actually a rewriting of equation 3 into the following formulation so as to make clear the perception process:

$$F^x\left(x(t), \sigma(t)\right) = (1-\epsilon) f(x(t)) + \epsilon\ p^x(\sigma(t)) \tag{7}$$

Actually the perception function $p^x(\sigma)$ will typically be the local mean field on the perception neighborhood of the agent, which leads to the following definition:

$$p^x(\sigma(t)) = \frac{\sum_{j \in V_i} f_a(x_j(t))}{N(V_i(t))} \tag{8}$$

Now that we have merged the computational model of CML within a MAS approach, the goal is to take advantage of the intrinsic dynamics of the iterating quadradic map which governs the agents, in other words to use the bifurcation diagram of the map in order to provide adaptation and organization capabilities to the MAS. The next sections clarify these points by showing how the design of the perception and action phases are essential to reach the global goal defined in the problem.

3. Self-organization and adaption in LMAS

3.1. Self-organization in the flocking phenomenon

In this subsection, we describe a simple implementation of the LMAS which leads to flocking simulations. This implementation shows a simple example of self-organization capabilities of the LMAS resulting from the synchronization processes occuring between the internal states of agents. The action phase here will simply "express" this synchronization in the environment space to give flocking, that is a group of coordinated moving agents. This implementation has been already presented and synchronization has been analyzed in [9]. Let us shortly recall its specifications and explain how the perception-decision-action loop is defined:

- The state of agent i at time t turns to the tuple: $s(t) = \langle x_i(t), a_i \rangle$
  a does not depend on time: a is randomly chosen for each agent. There is also a global coupling parameter ϵ_0 which is the same constant factor for each agent – this may be considered as an intrinsic characteristic of the whole population– .
- The environment is a 2D continuous torus where agents are located with two spatial coordinates.
- Two discrete fields denoted $\mathcal{X}$ and $\mathcal{N}$ are considered: the field $\mathcal{X}_k(t)$ stores the $f(x)$ values of the agents on a given site k at time t, and $\mathcal{N}_k(t)$ is the number of agents located on the same site k at time t.
- The perception function p_i of an agent i corresponds to the mean of the $\mathcal{X}$ field over the agent neighborhood denoted $V(i)$:

  $$p_i(t) = \frac{\sum_{k \in V_i} \mathcal{X}_k(t)}{\sum_{k \in V_i} \mathcal{N}_k(t)} \tag{9}$$

  $p_i(t)$ always belongs to $[0,1]$ since it proceeds an local average over the field $\mathcal{X}$. This corresponds precisely to the expression of the diffusive coupling in the equation 3 by means of field perceptions.
- The decision is achieved through the internal state update in (6).
- Moving and updating actions: the updated $x(t+1)$ indicates the new direction of the velocity —$2\,\pi\,x$ gives an angle in a simple way—, the velocity magnitude remaining a constant here.

The final master transition equation for the x component is summarized in the following expression:

$$x_i(t+1) = (1-\epsilon_0) f_a(x_i(t)) + \frac{\epsilon_0}{\sum_{k \in V_i} \mathcal{N}_k(t)} \sum_{k \in V_i} \mathcal{X}_k(t) \tag{10}$$

Figure (2) shows a snapshot of such a flocking simulation involving 50 agents whose initial control has been chosen randomly in D and with $\epsilon = 0.96$. The size of the environment is 100×100, knowing that the magnitude speed is fixed to 1.0 per time step. We notice some clusters of partial synchronization in the shape of agent flocks, which are unstable: they split as soon as they cross the path of another cluster. The conclusion of this flocking simulation have to stress two points: firstly we show that the geometrical self-organization process –indicated by the flocking formation– is related to a dynamical synchronization process occuring in the internal state of the agents [9]. Secondly we controled the move in a $2D$-space with only a $1D$-quantity, that is the decision variable x which is a noteworthy feature of this implementation. The following part of the paper is dedicated to more sophisticated perception-action phases so as to make possible adaptation processes in LMAS, through a predators-prey pursuit application.

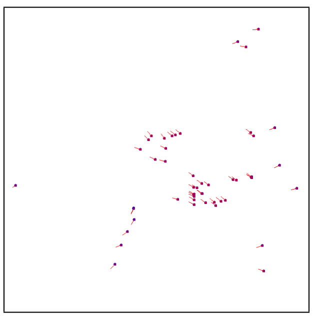


Figure 2. Snapshot of a flocking simulation: $n = 50$, $\epsilon = 0.96$, radius of neighborhood$= 20$

3.2. Adaption processes in LMAS

In the preceding flocking simulation, we stressed the group of agents: each agent was influenced by the others in its neighborhood and no "external" data stepped in. Now we intend to show a specific effect of the influence of initial data marked down in the environment. The objective is to perform adaptation processes through variations on a, which is the control parameter of the quadradtic map used for the transition on x. The environment plays this way the role of a strong stimulus for agents by modifying their internal dynamical behavior indirectly through the control parameter of the quadratic map. Whereas the a-variable was a stationary variable in the flocking situation, the perception functions from now on induce changes on both components: we distinguish the perception for the x component of the agent state from the perception for the a component. The associated perception neighborhoods may be different as well: they will be differenciated by mentioning the field involved: $V^{\mathcal{X}}$ for the x-perception and $V^{\mathcal{D}}$ for the a-perception. As before, ϵ remains an identical constant for the whole group of agents. The capability of agents is reduced to read environment data but not modify them. They keep writing however their own internal state information in the $\mathcal{X}$ field for the synchronization process as before.

Let us summarize the model improvements:

- The state of the agent i at time t turns to the tuple: $s(t) = \langle x_i(t), a_i(t) \rangle$
  where each component depends on time, whereas the coupling parameter ϵ_0 remains the same for all agents.
- The environment holds a new discrete field denoted $\mathcal{D}$ for "data", which is a grid whose cells are initialized with values in $[0, 1]$. These data are persistent in the field.
- The perception function associated with the a component of an agent i is denoted p_i^a. It consists in reading the data in the $\mathcal{D}$ field of the environment in a vicinity of the agent i. This perception function acts in fact as an operator (average operator or min operator) on the $\mathcal{D}$ field.
- The transition on the a variable for the agent i is expressed by:

$$\left\{ \begin{array}{ll} a(t+1) & = F^a(a(t), \mathcal{D}(t)) \\ & = (1-\epsilon_0)a(t) + \epsilon_0 \; p_i^a(\mathcal{D}(t)) \end{array} \right. \tag{11}$$

In the next section we specify an adaptation process specific to the prey-predator pursuit problem.

3.3. The predators-prey pursuit case

We present in this section a derivative case of predators-prey pursuit. This type of problem has been formulated for the first time by [10]. We propose to slightly different problem: the system is composed of one prey and many predators (from 20 to 50 here). The goal of predators consists in flocking round the prey and stay around to stop it moving: the prey speed is inversely proportional to the number of predators surrounding it. This moving rule of the prey is very simple and does not correspond to a realistic biological situation —it may depend on the species involved too…— : on the contrary one would have expected the prey to speed up or to change its moving direction. But at this stage of the study, we focused only on the capacity of the group of predators to aggregate on the prey in a dynamical basic case. We define a terminating state as follows: if 10 predators surrounds a prey –that is they stay in the Moore neighborhood (8 sites) of the prey– this prey is considered to be captured, which corresponds to the end of the pursuit (and of the simulation run). After this description of the problem, let us explain some specifications of the proposed modeling:

- The $\mathcal{D}$-field is initialized by the value 1.0 everywhere. This value corresponds to a chaotic regime for the predators: when there is no prey, predators explore the space (see explanations on the moving rule for predators).
- The prey marks down the environment in the $\mathcal{D}$ field which reveals its presence to the predator, since the predator can perceive it by means of its p^a perception

(a) Simulation at $t = 0$

(b) The prey has been captured at $t = 500$

(c) The whole group of predators has flocked round the prey at $t = 10000$

Figure 3. Three snapshots of a prey-pursuit simulation with $N = 50$ predators. The prey is represented as a circle. Parameters are $\epsilon_0 = 0.2$, radius of prey vicinity$= 5.0$, radius of predator perception$= 1.0$

function. The $\mathcal{D}$-field is modified by the prey in this way: a prey marks locally the environment in the $\mathcal{D}$-field to diffuse a vicinity of "presence" around itself (see fig.3). This marking process is not persistent in the environment (in other words, this field has no memory) but follows the prey at each time step. This marking is achieved by setting the sites of the prey vicinity to values from 1.0 on the border to 0 in the center of the vicinity. This gradient enables predators to slow down near this vicinity and maybe to turn to the prey and stop on its location. These values have to be interpreted regarding the bifurcation diagram of the quadratic map on Fig.2, which gives for each a-value the corresponding agent dynamical behavior from chaotic ($a = 1.0$) to static ($a = 0$).

- the moving behavior of the prey is very simple here: a straight direction choosen randomly and an initial speed slower than predators. The velocity magnitude of a prey is constant –it is set to $V_{prey} = 0.05$ unit per time step in the simulations without any predator– and decreases with the number of predators in its presence-vicinity according to the formula:

$$v \leftarrow \frac{v}{\sum_{k \in V_{prey}} \mathcal{N}_k(t)}$$

  where $\mathcal{N}_k$ denotes the number of predators in site k of the presence vicinity. The prey almost stops as soon as about ten predators are located in its vicinity.
- the perception function F^a of a predator is then expressed as follows:

$$a(t+1) = (1 - \epsilon_0)a(t) + \epsilon_0 \min_{k \in V_i^{\mathcal{D}}} \{\mathcal{D}_k(t)\} \qquad (12)$$

  The $\min$ operator enables to select the lowest values in the $\mathcal{D}$-field so as to make predators react more quickly near the prey.
- About the moving action of predators:
  the velocity magnitude is proportionnal to the internal updated x_i value for each agent-predator i, while the direction of velocity follows a cumulative process according to:

$$\left\{ \begin{array}{ll} \theta_i(t+1) & = \theta_i(t) + (0.5 - x_i(t))\ \delta \\ v_i(t+1) & = x_i(t+1) * v_0 \end{array} \right. \qquad (13)$$

  with $v_0 = 1$ unit per time step in our simulations. The cumulative process on θ_i prevents absolute directions in the move, as it was the case in the flocking simulation. The δ angle is an ad-hoc parameter to adjust the predator moving for capture –δ is in the size of 20 degrees–. At the beginning of a simulation run, every predator has a chaotic dynamics. This chaotic dynamics is transfered to the angle through the above formula. This formula enables therefore predators to explore the environment when no prey is perceived, and converge on the prey in a reactive way when the perception of the $\mathcal{D}$-field is decreasing. Consequently, the agents can no more form a flocking pattern, they move more like flying bees than like flocking birds.

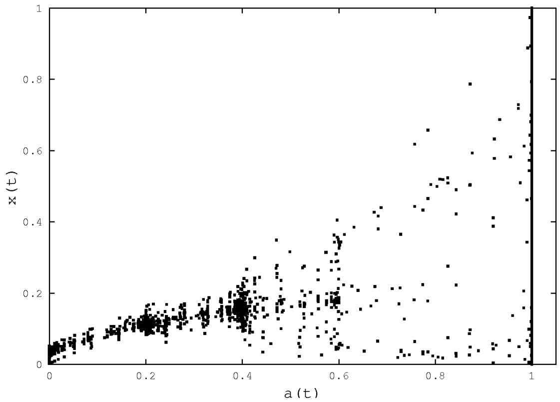
Finally the internal transition equations for a predator may be globally written as:

$$\left\{ \begin{array}{ll} a(t+1) & = (1-\epsilon_0)a(t) + \epsilon_0\ \min_{k \in V_i^{\mathcal{D}}} \{\mathcal{D}_k(t)\} \\ x_i(t+1) & = (1-\epsilon_0) f_{a(t+1)}(x_i(t)) + \\ & \frac{\epsilon_0}{\sum_{k \in V_i^{\mathcal{X}}} \mathcal{N}_k(t)} \sum_{j \in V_i^{\mathcal{X}}} f_{a(t+1)}(x_j(t)) \end{array} \right. \qquad (14)$$

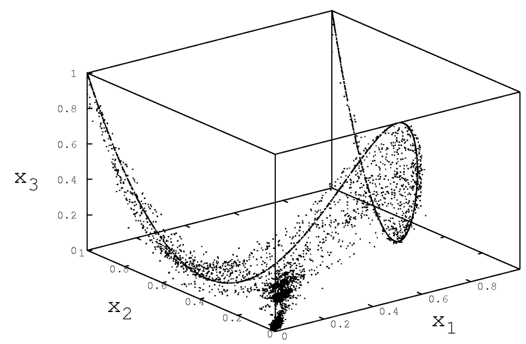
All the processes detailed here aim at a single objective: to lead the agent internal state to the point $(0, 0)$ when the agent meets the prey. In the next sections, simulation results are analysed and discussed.

3.4. Simulation results

At first, we set the values of some parameters in the simulations as follows:



(a) Evolution of the linked bifurcation diagram



(b) $X_1 = x(t)$, $X_2 = x(t+1)$, $X_3 = x(t+2)$ – 3D-reconstruction of the attractor for the considered agent-predator.

Figure 4. Evolution of the behavior of a predator which found the prey on 10000 time steps.

- The environment is a continuous torus of size 100.0×100.0 as before (the agent location is defined by two double values), containing the discretized fields (grids of size 100×100) we described in the latter section — the $\mathcal{D}$ field is initialized to 1.0 in each site of the grid—.
- The performed simulations involve 40 or 50 predators.
- Initial conditions: the initial values of the state variables in each agent are set to 1.0 for the control variable and at random for the decision variable.
- The neighborhood of presence of the prey –denoted V_{prey}– has been set to a radius of 4.0 or 5.0, which corresponds to about the twentieth of the environment size.
- The predator's neighborhood for perceiving the $\mathcal{D}$-field –denoted $V^{\mathcal{D}}$ in the formula (12)– has a lower radius: it has been fixed to 1.0 to enable predators to get as close as possible from the prey.
- In fact both mentioned neighborhoods are involved in the same mechanism: the detection of the prey by the predator. The presence of the prey marked in the environment is required by the indirect kind of the interaction. All the interactions have to be marked in the environment through specific fields. Agents have just to read and interpret the informations in the environment.

Figure 3 presents three snapshots of a simulation from the beginning to the end of the run. The left image shows the prey with many predators and its presence neighborhood. The central snapshot shows a prey about to be captured. The right image shows a captured prey: there are more than ten predators in its Moore neighborhood. At this stage, the prey does not move anymore on screen, but the velocity of the prey never equals zero because of the formula above. Although the prey keeps moving very slowly, predators in the prey's neighborhood are in a stable state so they will stay indefinitely there. With the parameter values used in this simulation, the time to capture a prey is about a few hundreds of time steps. When a capture fails –see below the range of parameters for which the capture may fail– because there are not enough predators around the prey for example, predators leave the prey and either return to a chaotic state —that is a random motion— or keep following the prey for a while.

As a capture may fail, a batch-processing of simulations has been performed in order to show or select the best set of parameters. Although this modeling intends to be as simple as possible, several parameters may improve the global performance. For instance:

- the coupling factor ϵ_0 between agent's x-internal states
- the radius of the linked perception neighborhood $V^{\mathcal{X}}$ which appears in equation 14 –this neighborhood has an attracting effect caused by the synchronization process involved inside–
- the angle increment δ in the motion rule (13)

To measure the global performance of the set of parameters we have recorded the catching time when the prey catching occurs before 10000 time steps. After this time limit, the catching time is set arbitrarily to 10000 so as to not hamper the data visualization. The simulation results are presented on figure 5 where the catching time is quantified by means of a colormap. The dark red is linked with high catching times and thus poor adaptive behaviors, whereas the dark blue reveals the best performances of the multi-agent system. The catching time is the average catching time on 5 runs with identical initial conditions. The three previously listed parameters –coupling, radius, angle increment– are the x-,y-,z-coordinates of the 3D-diagrams. This batch-processing of simulations leads to the following conclusions:

- global performances are improved by the number of predators. Other simulations not reported here confirm this tendency but we could not infer any simple law from these data

- best performances give a catching time less than 1000 time steps
- the coupling factor don't need to be very high –the perfect value reveals to be about 0.1–
- the radius of the perception neighborhood $V^{\mathcal{X}}$ seems to be a crucial factor since results show that a large neighborhood do not induce automatically an improvement of performances
- the angle increment δ has to be greater than or equal to $30°$in order to enable catching with good performances.

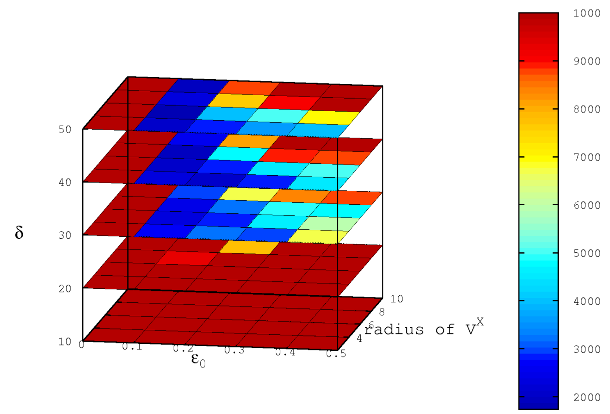
These results may guide future developments of the design of decision and action processes. The variations of these parameters may be governed by some supplementary perception functions which will improve the global adaptation process. Finally other simulations have been achieved involving more than one prey: with several preys the process does not disrupt, as the snapshot on figure 6 illustrates this fact. The next section aims at explaining and discussing the involved mechanisms by using some specific visualization tools.

4. Discussion

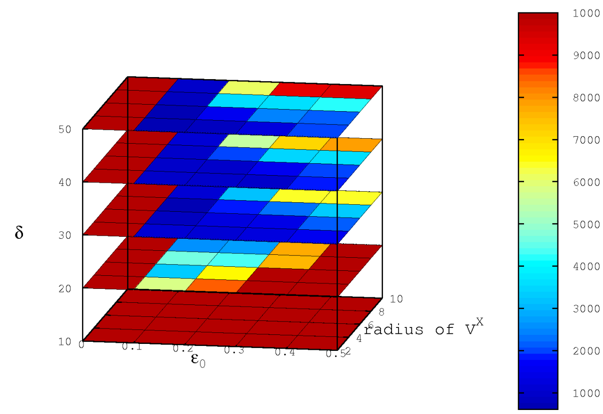
The following discussion deals with two important points: firstly the self-adaptation process and its mechanism, secondly the dynamical qualitative interpretation by means of visualization diagrams. To begin with, let us summarize the capture process: a capture can occur because predators can perceive the presence neighborhood of the prey by reading locally the $\mathcal{D}$-field of the environment. When a value different from 1 is detected, the predator will slow down there and begin to curve its trajectory, because its internal control variable decreases with the perceived values in this area. This low control value implies that the x-values decrease as well according to the bifurcation diagram of the considered quadratic map and consequently the velocity magnitude of the predator. As this predator is coupled with others in its neighborhood, this predator influences also the other predators. All these cumulative processes constitute the global self-adptation process which "propagates" within predators.

Now let us interpret this behavior according to the dynamical system view. We can see on figure 4 a view of the global dynamics of a predator during the catching of the prey. Both charts show the same dynamics of the same predator in different ways. The left one is achieved in the same way as the bifurcation diagram of the quadratic map, and shows the effect of the prey detection on the evolution of the predator's control variable a. The right one shows the attractor of the chaotic behavior of the predator in 3 dimensions by means of the "delays method". Let us analyze these two charts. The figure 4(a) shows the way a predator "travels" inside its bifurcation diagram. The graph is to be read from right to left because a capture proceeds in this way, that is from the value $a = 1.0$ to the value $a = 0$ which corresponds to the time when the predator is "fixed" on the prey since its decision variable $x = 0$. The prey detection by a predator induces indeed a decreasing of the a variable of this predator which consequently lowers the internal x decision variable of the predator. This effect is geometrically expressed via the action processes by a reduction of the predator's speed and by the predator turning round the prey. On the other view 4(b) of the same agent dynamics, the attractor of the dynamics is reconstructed in a 3D-space according to the delays method [11]: according to this theorem, a 1D dynamics may be embedded into a 3D-space. One can easily distinguish on the one hand the global initial chaotic attractor, which is a parabola formed into a curve with many points around due to the interactions between agents, and on the other hand a cloud of points close to the origin-point $(0, 0, 0)$, occuring later in the time dynamics. This cloud represents the catching phase where the predator finally stays on the prey. One may interpret this mechanism as a means to set the prey as a dynamical attractor for predators, more precisely a fixed point attractor with a limited basin of attraction around the prey. The moving formula 13 of predators is the onto map which transforms the abstract space of internal x values into the environment geometrical space. In dynamical system terms, the presence neighborhood of the prey constitutes a basin of attraction for predators. This goal is achieved through two convergent processes: firstly the decreasing of the a control variable of the predator near the prey, and secondly the specific moving rule which results in a rotary motion inside the presence neighborhood of the prey. The coupling effect increases the size of this basin of attraction. Finally we may assert as a conclusion that the self-adaptation process here has resulted from a mechanism design based on the construction of a basin of attraction in the dynamical phase space of the whole system. This type of analysis is made possible because of the dynamical approach we followed in the LMAS design.

Some important remarks have to be mentioned to finish with this section. Firstly the LMAS has only reactive mechanisms. Although we didn't need any explicit potential field to achieve this, predators are attracted by the prey in a nonlinear way, even if the prey is moving. It is important at this stage to notice that all decisions and consequently all actions are generated from the internal state variable x which lives in a 1D-space whereas the environment is a 2D-space. Moreover data flows are completely defined for control, coupling and decision processes. Secondly, this prey catching looks like some optimization problems actually. Predators find indeed some zero points in the environment and react automatically on them by flocking round and stopping on. In the Particle Swarm Optimization field, many problems consists in finding the minima of nonlinear functions in high dimensional



(a) Simulations with 20 predators



(b) Simulations with 50 predators

Figure 5. Simulation results for different number of predators. The presence neighborhood of the prey has a radius of 5 in all these runs. The catching time is represented by colors and is calculated as the average catching time on 5 runs.

spaces. The De Jong's set of functions [12] gives some characteristic functions used for comparing the algorithms. Some of these functions may correspond to the $\mathcal{D}$-field of the environment with a prey as we defined it: Easom's function or Ackley's Path function (see [13]) are particularly close to our predator-prey problem. Although the LMAS algorithm needs to be develop for this specific context, these predator-prey simulations have shown new potentialities of the algorithm, especially when the environment is a dynamic one in which the zeros points (preys) are moving.

Figure 6. Snapshot of the simultaneous catching of 3 moving preys

5. Conclusion

This paper has proposed to describe the evolution of the LMAS model to tackle self-adaptation issues. From a flocking case, the LMAS could be modified by adding a perception function for governing the control variable through the environment, and indirectly governing the internal decision variable in each agent. Then by giving agents new perception and action capabilities, the system became more adaptive to the environment fields and enabled the cathing of preys in a predator-prey pursuit problem. The design of perception and action functions reveals to be crucial in this reasoning. The set of mechanisms involved for this adaptation process achieves the self-adaptive property of the group of agents. However, the self-adaptation process may be improved as the results of the batching-process shows: some parameters could be fitted automatically by some appropriate perception functions. Finally we showed that this self-adaptation had an explanation in terms of dynamical system theory. To be self-adaptive, the properties of the global phase space must be changed: for instance in the predator-prey problem, the prey has to become a "fixed point" for the predator's dynamics. To reach this objective, the mechanism design has consisted in linking the bifurcation diagram of the predator's internal logistic map with the geometrical phase space of the environment through appropriate perception and action functions. As a result of which the prey acts as the center of a basin of attraction for predators. The novelty of this mechanism lies in the replacement of potential field approaches which one could have used for this type of problem, by a bifurcation control approach. The modern dynamical system theory has powerfull tools and accurate expressiveness for explaining natural phenomena. We stress here that dynamical approaches have to be developped in the swarm intelligence field so as to deeply understand the involved mechanisms. Future prospects with LMAS will deal with optimization problems –as it was discussed at the end of the paper– and with swarm robotics. We expect indeed that LMAS could be a very appropriate algorithm for swarm robotics because of its simple basis and its low computational cost.

References

[1] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence. From natural to artificial systems*, O. U. Press, Ed. Berlin Heidelberg New York: Springer-Verlag, 1999.

[2] K. Kaneko and I. Tsuda, *Complex systems: chaos and beyond*, L. T. Asakura Publishing Co., Ed. Berlin Heidelberg New York: Springer-Verlag, 2001.

[3] B. Doyon, B. Cessac, M. Quoy, and M. Samuelides, "Control of the transition to chaos in neural networks with random connectivity," *International Journal of Bifurcation and Chaos*, vol. 3, pp. 279–291, 1992.

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

[5] R. Charrier, F. Charpillet, and C. Bourjot, "Deterministic nonlinear modeling of ant algorithm with a logistic multi-agent system," in *AAMAS '07, International Conference on Autonomous Agents and Multiagent Systems,Hawaii, 14-18 mai*, 2007.

[6] K. Kaneko, "Clustering, coding, switching, hierarchical ordering, and control in network of chaotic elements," *Physica D*, vol. 41, pp. 137–172, 1990.

[7] T. Shibata and K. Kaneko, "Coupled map gas: structure formation and dynamics of interacting motile elements with internal dynamics," *Physica D*, vol. 181, pp. 197–214, 2003.

[8] J. Ferber and J.-P. Müller, "Influences and reactions : A model of situated multiagent systems," in *Proceedings of 2nd International Conference on Multiagent Systems (ICMAS-96)*, T. M., Ed. The AAAI Press, December 10-13 1996, p. 72–79.

[9] R. Charrier, C. Bourjot, and F. Charpillet, "Flocking as a synchronization phenomenon with logistic agents," in *ECCS'07, European Conference on Complex Systems, Dresden*, Dresden, October 1-5 2007.

[10] M. Benda, V. Jagannathan, and R. Dodhiawala, "On optimal cooperation of knowledge sources - an empirical investigation," Boeing Advanced Technology Center, Tech. Rep., July 1986.

[11] F. Takens, "Detecting strange attractors in turbulence," in *Proceedings of the Symposion on Dynamical Systems and Turbulence,University of Warwick, 1979-1980*, D. Rand and L. Young, Eds. Springer, 1981.

[12] K. De Jong, "An analysis of the behaviour of a class of genetic adaptive systems," Ph.D. dissertation, University of Michigan, 1975.

[13] H. Pohlheim. (2006) Geatbx: Genetic and evolutionary algorithm toolbox for use with matlab. [Online]. Available: http://www.geatbx.com/docu/fcnindex-01.html