Proceedings of EPNACS 2011 within ECCS'11
Emergent Properties in Natural and Artificial Complex Systems
Vienna, Austria - September 15, 2011

TOWARDS A SWARM OPTIMIZATION ALGORITHM WITH "LOGISTIC" AGENTS.

Rodolphe Charrier *

Abstract. Adaptation and self-organization are two main aspects of swarm mechanisms and collective intelligence. We adress here the global question of controlling the coordination of groups of mobile agents in order to achieve optimization processings. We study in this paper a simple case involving "logistic" agents –whose internal decision is governed by a logistic map, that is a discrete parametrized quadratic map– slaved to a stochastic environment through their control parameter. The proposed algorithm enables agents to find local minima in the environment. We show that the adaptation process on the control parameter leads to this local optimization. Applications may be envisaged for multi-objective problems.
Keywords. Swarm Intelligence, Reactive Multi-Agent System, Multi-objective Optimization, Logistic Agent, Adaptation, Self-organization, Decentralized Control.

1 Introduction

The root principle of Collective Intelligence consists in considering intelligence no more as an individual characteristic only, but also as an emergent phenomenon involving (self-)adaptation and self-organization processes. This type of intelligence may be massively distributed and seems very robust to perturbations and failures. Research on Collective Intelligence has provided efficient meta-heuristics for approximating large sets of problems, notably in the Swarm Intelligence field [3]. The challenge is to understand and model properly the involved mechanisms so as to build appropriate and efficient algorithms.
The majority of proposals in modeling Swarm Intelligence in particular, is based on specific probability laws used as decision functions like Ant Colony Algorithms for instance [10], or based on randomized algorihtms like Particle Swarm Algorithms [11].

We address in this paper the challenging problem of controlling the collective motion within a reactive multi-agent system to achieve optimization processings. In this way, we have followed the Particle Swarm Algorithms evolution by focusing first on flocking-like phenomena. Flocking has a great interest firstly because of its importance in nature –many species adopt such a coordinated form of collective motion–, and secondly because it may have many applications in complex network studies such as social networks, ad hoc networks, swarm robotics, ... The two most used modeling frameworks for achieving flocking simulations are currently Reynolds's approach [1] and Vicseck's model [2].

Reynolds has based his approach on three main rules –collision avoidance, velocity matching and flock centering– without giving any mathematical specifications. On the contrary, Vicseck has expressed an explicit dynamical system. Both derives from particle systems and provide a synchronization process: Reynolds uses a velocity matching rule while Vicseck involves the average of the neighbors' velocity angles to compute the velocity angle of the current particle.

Our approach to tackle the flocking phenomenon is quite different. On the one hand, our mobile entities are not particles but agents. This enables to distinguish the geometrical/kinematic characteristics from the internal state of entities. The encapsulated internal state gives to agents an autonomy of decision for any type of actions, not only for moving. On the other hand, we stress on the synchronization processes, which must be involved in the internal decision making to our opinion. Our main working hypothesis is that many swarm phenomena may be described and analyzed in terms of oscillator synchronization and decentralized control, in the spirit of [8]. In that way we have proposed a particular model which is a reactive multi-agent system (MAS) model called Logistic Multi-Agent System (LMAS) [5], because of the internal logistic map governing the decision process of each agent. It refers also to the Coupled Map Lattices (CML) family of computational models, particularly the ones involving logistic maps [7]. Logistic maps are one-dimensional deterministic quadratic maps controlled by a single parameter, which can generate chaotic series. These maps are very interesting because of their wide range of dynamical behavior –from complete randomness to single fixed points–, and their facility to be coupled and controlled. This specific dynamical behavior enables to use such a deterministic series generator instead of

*Rodolphe Charrier are with LITIS, University of Le Havre, France. E-mails: rodolphe.charrier@logistic-swarm.net

a probabilistic one. This change opens a new research perspective which constitutes the heart of our studies.

In this paper, we firstly recall the ability of the LMAS to produce flocking simulations by synchronization caused by individual couplings. This constitutes an introduction for the main case we want to tackle. We intend to show how one can apply the self-organization and adaptation capabilities of the LMAS to find minimum values in a stochastic environment. A decentralized and adaptive control mechanism is used to achieve the motion coordination and the convergence towards the local minima. This mechanism is based on control variations of the logistic map, caused by the local agents' perceptions of random data stored in the environment. Simulations show the agregation of agents on the same minimum field locations. The discrete dynamical system theory enables to partially explain this resulting phenomenon. In particular, we stress on the fact that the whole group of agents converges to a decision fixed point. This also shows that dynamical modeling approaches are appropriate for analysing phenomena and for monitoring the dynamics.
The paper is organized as follows. The first section recalls the formulation of the LMAS model. We then some collective motion results by showing how a flocking may appear within a LMAS in a first step. In a second step, we explain the algorithm for finding local minima in a data field of the environment. Then we discuss results and conclude this work.

2 The Logistic Multi-Agent System (LMAS)

This section begins with a short 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. We then show how we have turned it into a multi-agent system.

2.1 A CML foundation

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, namely logistic maps. These types of CML have been widely explored by Kaneko since the 80's [7].
A CML is a discrete time and space cellular model in which cell states x take their values in a continuous domain. A CML is sometimes considered as a cellular automaton with continuous states. Let us focus on a mean-field instance of the CML class, namely the globally coupled map lattice (GCM) [6]. A GCM is governed by the 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 N is the total site number in the lattice, x_i^t is the state variable of the cell on site i at time t and ϵ the diffusive coupling coefficient. 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:

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

where $a \in [0,1]$ denotes the control parameter of f. We intentionally use the notation $f(x,a)$ to stress on the importance of the control parameter a in our case studies. This map is known to generate chaotic series as t tends to ∞, in particular if $a = 1$, or to converge to some fixed points or periodic cycles in function of the value set for a.
The asymptotic behavior of this nonlinear map is completely described by its bifurcation diagram on Fig. 3, which plots $x^\infty = f^\infty(x^0)$ as a function of the control parameter a.

Shibata and Kaneko have anticipated the evolution of this model to fit with the Swarm Intelligence field. They proposed a specific CML instance called the CML gas in 2003 [9]. A CML gas is a CML-like model in which chaotic cells are motile on the lattice. But to our knowledge, there was only one paper about this research perspective. The LMAS may be considered as an agent based evolution of the CML model specifically designed for Swarm Intelligence. The next section describes this evolution.

2.2 Conception of the LMAS

Contrary to CML, a multi-agent system distinguishes the group of agents from the environment. The environment is the space where agents can perceive data and make actions (move for instance). Consequently, each agent encapsulates its internal state, which enables autonomy in acting.

Definition 2.1. The logistic multi-agent system (LMAS) is composed of N agents called logistic agents. A logistic agent is a reactive agent whose internal decision is governed by a logistic map.

The internal state s of a logistic agent is composed of three variables corresponding of the three quantities of CML:
$s = \langle x, a, \epsilon \rangle$

- x is the decision variable computed by the logistic map characteristic of the agent

- a is the control variable of the agent, which is also the control parameter of the logistic map. It governs the type of dynamical series generated by the map given by the bifurcation diagram in Fig. 3.

- ϵ is the coupling variable of the agent which quantifies the strength of the influence of its neighbors' decision.

All these three variable have values in the real interval $D = [0, 1]$. The logistic agents can locally perceive their environment. In the most general case, each of these quantities can be modified by the environment through perception functions, that is why they have a variable status since they may change with time.

Here is summarized the decision-making of the agent i in one master equation derived from the formula 1:

$$x_i^{t+1} = (1 - \epsilon_i^{t+1}) f(x_i^t, a_i^{t+1}) + \frac{\epsilon_i^{t+1}}{|V_i^{t+1}|} \sum_{j \in V_i^{t+1}} f(x_j^t, a_j^{t+1}) \quad (3)$$

where all parameters may depend on time. The influence of the other agents is reduced to the current neighborhood V_i of the agent i, which contains $|V_i|$ other agents at the current time. This master equation gives the rooting principle of the model and the scheduling of the computation: all quantities are computed before the decision variable x during the perception process. This equation corresponds to the decision process.

The action process consists in using the decision variable x to perform actions like moving or updating local data in the environment (a pheromone field for instance). Finally, the computation process is clearly divided in three parts — perception-decision-action — which is coherent with a cybernetic approach. In the next section, we apply this model to collective motion simulations.

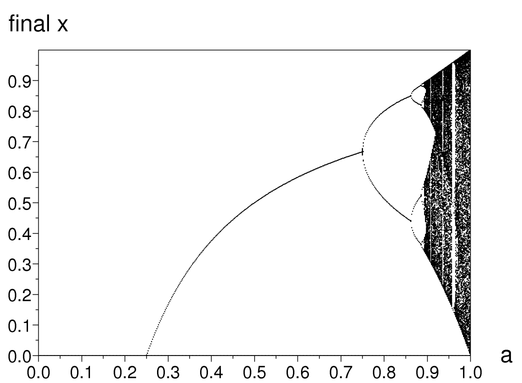


Figure 1: Bifurcation diagram for the logistic map $x_{t+1} = 4\ a\ x_t\ (1 - x_t)$, calculated with 500 iterations on 500 samples of the interval $[0, 1]$

3 LMAS based simulations

3.1 The flocking phenomenon

In this subsection, we describe a very simple implementation of the LMAS which leads to flocking simulations. The root principle is to interpret the decision variable x of the agents as a moving angle in the geometrical space. This case shows a simple example of the LMAS self-organization capabilities resulting from the synchronization processes which occur between the internal states of agents. The action phase of moving will “express” this synchronization in the environment space as a flocking phenomenon, that is a grouping of coordinated agents. This implementation has been completely presented and the synchronization process analyzed in [4].

Let us shortly describe the specifications of the modeling for this case:

- The state of the agent i at time t turns to the tuple: $s_i^t = \langle x_i^t, a_i, \epsilon \rangle$

- a_i and ϵ are constants in the current case.

- The value of a_i is randomly chosen in $[0, 1]$ for each agent, that is each agent may have any type of dynamical behavior according to the bifurcation diagram 3.

- The coupling variable ϵ is a global constant as well, but may be set with various values initially.

- The environment is a discrete 2D torus where agents are located randomly initially.

Let us comment the perception-decision-action processes for the agent i:

- The perception gives the second term of equation 3: $P_i = \frac{\sum_{j \in V_i^{t+1}} f(x_j^t, a_j)}{|V_i^{t+1}|}$

- The decision is calculated by means of equation 3).

- Then the moving action consists in updating the direction of the velocity (cf. fig 2) according to the updated decision value:

$$\begin{cases} \alpha_i^{t+1} &= 2\pi\ x_i^{t+1} \\ \mathbf{r}_i^{t+1} &= \mathbf{r}_i^t + v_0\ \mathbf{u}_{\alpha_i^{t+1}} \\ \theta_i^{t+1} &= \arg(\mathbf{r}_i^{t+1}) \end{cases} \quad (4)$$

  gives the new velocity direction of agent i in a simple way.
  The velocity magnitude remains a constant here.

Thanks to this modeling, we can control the move in a $2D$-space with a $1D$-quantity only, which is an important feature of this model.

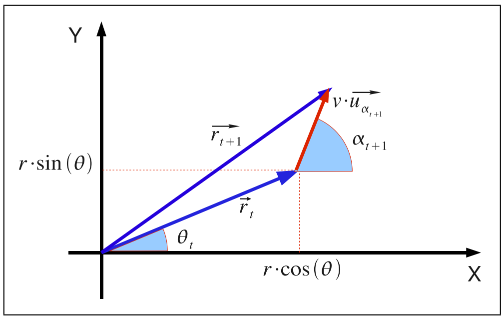


Figure 2: Description of the agent's motion in cylindrical coordinates.

The final master transition equation for the decision x is summarized in the following expression for the agent i:

$$x_i^{t+1} = (1-\epsilon) f(x_i^t, a_i) + \frac{\epsilon}{|V_i^{t+1}|} \sum_{j \in V_i^{t+1}} f(x_j^t, a_j) \quad (5)$$

Figure (3) shows a snapshot of such a flocking simulation involving 50 agents whose initial control has been chosen randomly in $[0,1]$ and with $\epsilon = 0.96$. The environment is a toric grid of 100×100 cells. The velocity magnitude is set to 1.0. Simulations lead to the formation of clusters of partial synchronization which gives the shape of flocks. These clusters are quite unstable: they split as soon as they cross the path of another cluster.
Another type of flocking simulation consists in involving only chaotic agents, namely with $a = 1.0$ in each agent. We let the reader refer to the paper [4] to get the whole analysis on the synchronization process for this type of flocking. The following case study is founded on the same type of formulation and dynamical mechanisms.

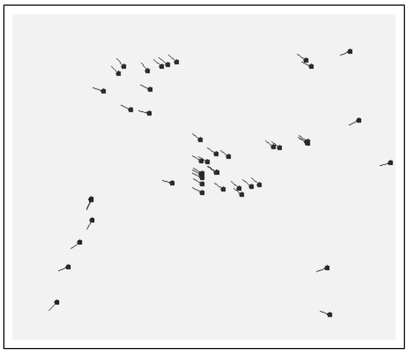


Figure 3: Snapshot of a flocking simulation: $n = 50$, $\epsilon = 0.96$, radius of neighborhood= 20

3.2 Towards optimization

3.2.1 Control by adaptation

In the preceding flocking simulation, we stressed on the group of agents: each agent was influenced by others in a limited 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. Whereas the a-variable was a constant in the preceding case, it is now modified through a perception function. The aim for the agents is now to find local minima of data in the environment by stopping at the corresponding locations. This aim is inspired from some optimization problems in the Particle Swarm Optimization (PSO) domain, where particle final positions indicate the best solutions for finding the zero values of a positive nonlinear function [12].
Let us summarize the main changes compared to the preceding situation:

- The state of the agent i at time t turns to the tuple: $s(t) = \langle x_i(t), a_i(t), \epsilon \rangle$
  a depends now on time. As before, ϵ remains a constant chosen initially.
- The environment holds now a discrete field denoted $\mathcal{D}$, which maps a random value in $[0,1]$ onto each cell of the environment —let us recall that the environment is a discrete grid here—. These data are persistent in the field, that is they are not modified by agents. The field values are indicated in our simulations by color levels from light gray for the value 1 to full white for the value 0.
- The perception function for the a component is quite simple: an agent can read local data in the $\mathcal{D}$ field and returns the mean value of these data in its perception vicinity. The updating proceeds in a convex coupling with the same coupling coefficient ϵ.
- Moving and updating actions:
  - as before the new direction of the velocity is set by the updated decision $x(t+1)$,
  - the velocity magnitude changes as well in function of the decision x according to the updating equations:

  $$\begin{cases} \alpha_i^{t+1} & = \alpha_i^t + 2\pi\delta \; (0.5 - x_i^{t+1}) \\ v(t+1) & = x(t+1) * v_0 \\ \mathbf{r}_i^{t+1} & = \mathbf{r}_i^t + v_0 \; \mathbf{u}_{\alpha_i^{t+1}} \end{cases} \quad (6)$$

  where v_0 is the initial velocity magnitude and δ is a coefficient on angle variations.

The transition system for the agent's internal state may be written as before, added with the updating of the control variable a:

$$\begin{cases} a_i^{t+1} & = (1-\epsilon) a_i^t + \epsilon \; \langle \mathcal{D} \rangle_{V_i^{t+1}} \\ x_i^{t+1} & = (1-\epsilon) f(x_i^t, a_i^{t+1}) + \\ & \frac{\epsilon}{|V_i^{t+1}|} \sum_{j \in V_i^{t+1}} f(x_j^t, a_j^{t+1}) \end{cases} \quad (7)$$

where $\langle \mathcal{D} \rangle_{V_i^{t+1}}$ denotes the average operator on the perception neighborhood V_i^{t+1} applied on the data field $\mathcal{D}$.

The control variable is computed before the decision variable since it is needed for the logisitic map calculation. We can easily verify that a_i always belongs to $[0, 1]$.

3.2.2 Simulation and results

Simulations have been settled as follows:

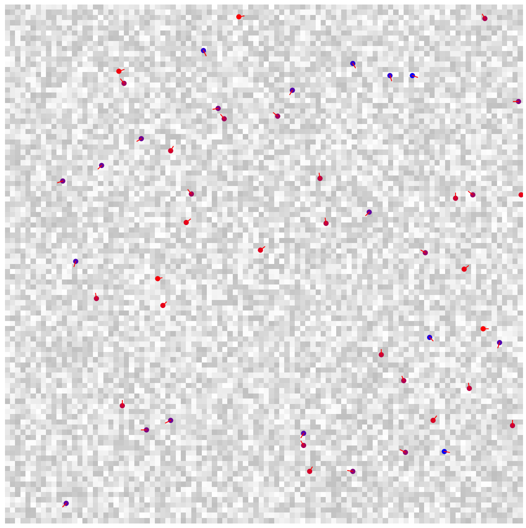
Environment	100×100 torus
Data field $\mathcal{D}$	100×100 grid
Weighted cells	at random in $[0, 1]$
Agent set	$N = 50$
Radius of perception	$R = 1.0$ ($\simeq 8$ cells)
Initial velocity magnitude	$v_0 = 1$
Angle increment	$\delta = \frac{1}{20}$
Initial agent’s decision x	at random in $[0, 1]$
Initial agent’s control	$a_i^0 = 1$
Global coupling	$\epsilon = 0.3$

The figure (3.2.2) shows three snapshots taken at different evolution stages of a simulation leading to three main agregation points. The agents’ coloring reveal their internal decision making: from light red for a chaotic behavior to light blue for a fixed point behavior. At time $t = 0$, every agent has a different velocity direction. Then at time $t = 2000$, three agregation areas begin to emerge. At time $t = 3700$ these agregation groups are definitively fixed and correspond to local minima given in the table (5(c)). Then each agent will progressively join one of these groups untill there are no moving agents left. This is caused by the attraction effect of the agregation of many agents on the same location: each agent extends the attraction area with its coupling potential.

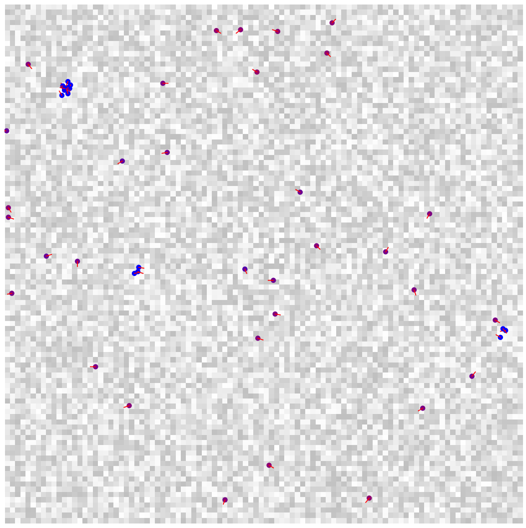
The figures (5(a)) represents the chart of the internal control (in red) and decision (in blue) variables of a randomly chosen agent. Its perception values are also plotted in green, showing the range of variations perceived in the environment. All these series have a mean value of 0.5 at the beginning, that is why the updating equation for the velocity angle (6) involves this particular value. The decision variable x becomes therefore the amount of variations around the initial velocity angle. This moving regime continues untill a minimum value or an agregation group is perceived. Then the agent’s decision converges dramatically towards zero, as it is shown in the figure 5(b) at time $t = 3250$. This evolution is qualitatively the same for each agent, depending on the agregation point.

4 Discussion

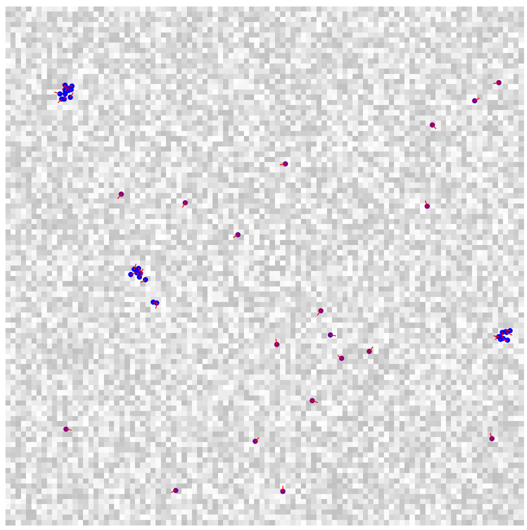
Now we have to comment on the simulation results.
First of all on the convergence process. To explain this convergence, we must look again at the bifurcation diagram of the logistic map (3). As long as the agent’s perception function returns values close to the mean 0.5, the



(a) t=0

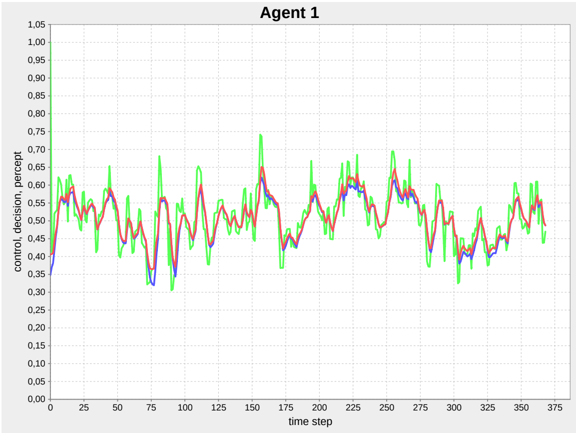


(b) t=2000



(c) t=3700

Figure 4: Snapshots of the simulation at different time steps.



(a) Variations of control (in red), decision (in blue) and perception (in green) with time for agent 1.

(b) Dramatic convergence to a local minimum.

Group 1	minima: 0.02, 0.06
Group 2	minima: 0.008,0.02
Group 3	minima: 0.04, 0.06

(c) Values of the minima detected in the field by each group.

Figure 5: Variations of the internal agent's variables and detection of local minima.

control parameter stays as well in this range of values and so does the internal decision variable of the agent. This is reinforced by the fact that the point $(a = 0.5, x = 0.5)$ is a superstable fixed point of the bifurcation diagram, that is $f'(\frac{1}{2}) = 0$ for the logisitc map. In other words, the variations on a are systematically counterbalanced by the induced variations on x.
But as soon as the perception operator –which is an average operator on about 8 cells– returns values close to 0.25, the agent's decision can converge rapidly towards the fixed point $(a = 2.5, x = 0)$. At this stage, the considered agent does not move anymore because of the moving laws (6), and can not leave this point, except if another agent is sufficiently near to draw it away by the coupling strength.

The second comment is about the other parameters in the simulation. In particular, we may wonder how the tuning on the radius of perception or on the global coupling parameter impacts the performances. This study has not been completely achieved so far. Some testings suggest tendencies:

- the perception neighborhood has to be as smaller as possible to capture the best minima,

- the coupling coefficient is the same for the coupling of internal agents' decisions and for the updating of the internal control parameter (cf. equations (7)). These two couplings are antagonist at first: while the control coupling tends to stop agents rapidly on field sites containing low values, the decision coupling can produce perturbations for already stabilized agents. This antagonism prevents to converge too soon toward unsatisfactory locations. The tuning of this parameter seems therefore essential for the quality of the convergence. Another way we do not have explored yet, consists in tuning the two coefficients separatly so as to give more tuning possibilities.

A last comment may be made on the quality of the minima found. At this stage of the study, our objective is not the absolute performance in the results but the proof of the concept. To improve performances, we have to compare the proposed algorithm with particle swarm approaches on benchmark problems [13].

5 Conclusion

This paper has proposed an algorithm based on the logistic agents' model which derives from the CML model, so as to tend to a multi-objective problem solving. After recalling the genesis of our logistic multi-agent system, we have applied this model to a small and basic problem of finding local minima in a random field within a toric environment. Although we can not conclude to the design of a new swarm optimization algorithm at this stage, we have laid its foundations.

The main interest of our approach lies in its nonlinear dynamical system theory, which provides strange but usefull mathematical maps, like the logistic map. We have exploited the interesting properties of this map, notably by linking the control parameter of the map to the data stored in the environment. The agents' decision process is therefore indirectly controlled and governed by the environment. It is also quite simple to explain by means of the bifurcation diagram of the logistic map.
This algorithm has to be improved to be more efficient to detect the best local minima. Testing on benchmark problems could be a future development strategy. Transforming the algorithm for high dimensional spaces may be a future prospect as well.

References

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

[2] T. Vicsek, A. Czirok, E. B. Jacob, I. Cohen, O. Schochet, ”Novel Type of Phase Transitions in a System of Self-driven Particles”, *Physical Review Letters*,75:1226–1229, 1995.

[3] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence. From natural to artificial systems*, Springer-Verlag, Berlin Heidelberg New York, 1999.

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

[5] R. Charrier, F. Charpillet, and C. Bourjot, “A Nonlinear Multi-agent System designed for Swarm Intelligence : the Logistic MAS”, *SASO'07, International Conference on Self-Adaptive and Self-Organizing Systems, Boston*, 2007.

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

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

[8] S. C. Manrubia, A. S. Mikhailov, and D. H. Zanette, *Emergence of Dynamical Order*, volume 2 of *World Scientific Lecture Notes in Complex Systems*, World Scientific, 2004.

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

[10] M. Dorigo, C. Blum, ”Ant colony optimization theory: A survey”, *Theoretical Computer Science*, 344:243–278, 2005.

[11] J. Kennedy, R. C. Eberhart, ”Particle swarm optimization”, *Proc. of the IEEE Int. Conf. on Neural Networks*, 1942–1948,1995.

[12] R. C. Eberhart, J. Kennedy, ”A new optimizer using particle swarm theory”, *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 39–43,1995.

[13] K. De Jong, ”An analysis of the behaviour of a class of genetic adaptive systems”, *PhD Thesis*,University of Michigan,1975.