

SELF-ADAPTIVE CONTROL WITHIN A CHAOTIC ANT SYSTEM FOR OPTIMIZATION

Rodolphe Charrier

LITIS-ISCN

Normandy University

France

Rodolphe.Charrier@univ-lehavre.fr

Cyrille Bertelle

LITIS-ISCN

Normandy University

France

Cyrille.Bertelle@gmail.fr

Abstract

This paper investigates the possibility of using parametric one-dimensional chaotic maps within a swarm of ant-like agents to perform optimization tasks. Ant algorithms derive from a stochastic modeling based on specific probability laws. We consider in this paper a full deterministic model of chaotic ants which uses a one-dimensional chaotic map –like the quadratic map– to govern the decision behavior of ant-like agents. Chaotic maps can produce pseudo-randomness for simulating stochastic behaviors with specific distributions. The main advantage of this approach lies in the possibility of controlling the chaotic properties of the iterated map through a single parameter per map. We deal actually with a decentralized swarm of agents individually controlled by only one parameter. This resulting chaotic swarm is driven by a “pheromone field”, that is a stigmergic process as classical ant algorithms do. This field modifies each individual control parameter by feedback. Finally we prove the efficiency of this approach on a Travelling Salesman Problem (TSP) instance, and compare the results with the original Ant System algorithm. To summarize, the algorithm submitted in this paper operates a distributed parametric control on agent’s internal decisions by means of a global pheromone field. This provides also a new insight into the stigmergic processes involved.

Key words

Swarm Intelligence, Ant Algorithm, Chaotic Map, Optimization.

1 Introduction

Ant algorithms constitute a family of stochastic models mainly based on a probability function used by artificial ants(?). This function modelizes the stigmergic process involved by ants in a stochastic way. Stigmergy is a concept firstly coined by Pierre-Paul Grassé in 1959 when he studied termite behaviors(?). Very briefly described, Stigmergy occurs when the outcome

of a collective work acts as a feedback loop on the individual’s behavior. Let’s get back to the point, the stochastic decision function used by artificial ants when they move on a graph-based environment is as follows:

$$p_{ij} = \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum_{l \in \mathcal{N}_i} (\tau_{il})^\alpha (\eta_{il})^\beta} \quad (1)$$

where τ_{ij} denotes the amount of pheromone on the edge (i, j) linking node i to node j , and η_{ij} denotes an extra heuristics which corresponds with the problem —as the inverse of the edge distance in the TSP case—, and \mathcal{N}_i is the set of other existing edges from node i . The exponents α and β are greater or equal to 1, but commonly equal 2 to get the best performances. This stochastic decision enables a colony of agents (artificial ants) to find or approximate good solutions for many hard combinatorial optimisation problems (?), like the Travelling Salesman Problem (TSP) we deal with in this paper. Basically a TSP problem consists in finding the shortest Hamiltonian cycle in a connected graph, that is the shortest cycle composed of all nodes occurring only once. This famous problem belongs to the NP-hard class of computing problems.

This paper focuses on the stochastic foundations of the ant algorithm modeling: more precisely our concern is with the existence of an alternative deterministic model which would exhibit the dynamical aspects of the involved stigmergic processes (?). Some works have been published in this way of modeling, by requiring the hypothesis of chaotic dynamics within ant behaviors (?; ?). But this approach has not yet led to optimization processes. This is precisely our purpose in this paper, by proposing a deterministic model of chaotic ants. This model is inspired by theoretical studies on iterated nonlinear maps, namely logistic maps or quadratic maps, which are well known to produce chaotic time series (?) with specific distributions. The main advantage of this approach lies in the possibility of controlling the chaotic properties of the iterated

map through a single parameter that modulates the dynamics. We deal therefore with the deterministic chaos theory to “replace” the probability theory. This field provides tools like bifurcation diagrams to monitor the processes. This change of the modeling paradigm — deterministic versus stochastic— leads to a novel insight into the internal mechanisms involved during the searching and optimizing process of ants. This proves to be a decentralized control process on each ant-like agent, driven by the whole pheromone field. Some authors speak of a “self-adaptive” process in this case.

Some work has been done using chaotic ant swarms applied on optimisation problems (?; ?), but it was only applied on continuous problem spaces. We tackle here graph-based problem spaces and combinatorial problems, and our modeling principles are quite different. Our chaotic ant model has been already applied on the binary bridge experiment and has proven to simulate the expected symmetry break of the problem, by producing the same shape of data series as the experimental ones (?). We aim at applying it on combinatorial optimization problems, as the original ant algorithms. Other approaches have involved chaotic maps as random generators in the particle swarm field. In particular the Chaotic Particle Swarm Optimisation approach seems to be very efficient for many optimization problems (?; ?).

The presentation of our chaotic ant model constitutes the first section of this paper. We intend then in the the second section to validate our approach by comparing the chaotic ant model to the “Ant System” algorithm (?) —one of the first ant algorithm instance in the family— on a Travelling Salesman Problem (TSP) benchmark with 48 nodes. The objective is at this stage to prove that the concept is relevant for combinatorial optimization, not to deal with hard TSP. The last section is dedicated to discuss our simulation results we think encouraging, and to open perspectives.

2 The chaotic ant system (CAS) for TSP

This section is devoted to the design of the chaotic ant system (CAS). In fact this system is a reactive multi-agent system (MAS) composed of an environment, that is the problem space, plus many chaotic ant-like agents. The agents interact through the environment within a pheromone field which enables stigmergy(?), that’s why we call them “ants”. However, the difference between chaotic agents and stochastic ants lies in the decision process of chaotic ants which is governed by a deterministic one-dimensional chaotic map —the quadratic map here—, in contrary to the stochastic law of the original ant system (1).

2.1 Metaheuristic principles of ant algorithms

Let us specify this in the case of a TSP problem where the objective is to find the shortest Hamiltonian cycle in a weighted graph¹. The generic considered graph is

denoted $G = (V, E)$ where V is the set of $|V| = n$ vertices and E the set of edges. In our case the graph is symmetric and has $\frac{n(n-1)}{2}$ edges.

Before describing the different parts of the logistic ant system, let us recall some rooting principles of ant algorithms. We use the same global metaheuristic method using a pheromone field to perform optimization on a symmetric TSP, that is:

N ants forming a colony are involved at the same time on a given TSP.

The algorithm proceeds in a global loop composed of global steps.

During a global step, each ant achieves individually an hamiltonian cycle from a random initial position and marks it by an amount of pheromone depending on the path distance.

Each global step ends with a pheromone reinforcement of the best cycle, when all ants have finished their own cycle.

In this paper, the elementary discrete time step t corresponds to the process time needed to achieve a “local loop”, that is a loop where all ants in the colony has moved into a new vertex of the graph. A global step lasts therefore $T = n - 1$ time steps for all ants to cover an Hamiltonian cycle in parallel. We keep the naming of local loop relative to the vertex and global loop relative to the graph, to distinguish the different levels of scheduling in the algorithm. The global loop lasts until a fixed limit of time steps is reached: this is the criterion to stop the algorithm in this study. The best optimization performance among the colony is then recorded.

2.2 The environment design

We use the concept of environment of the MAS paradigm to include all the data related to the graph problem. In this way, our MAS environment is composed of a basic geometric space denoted E and many fields on it. E corresponds to the $2D$ discrete space $\mathbb{N}_n^* \times \mathbb{N}_n^*$ where $\mathbb{N}_n^* = \{1, \dots, n\}$ and n is the number of vertices in the considered graph. So E can be stored as a matrix of $n \times n$ cells. Each field is an additional layer of this matrix.

2.2.1 The notion of field The field concept is the means we use to structure the data in the environment and to describe dynamically all the processes within the CAS. A field is defined as a mapping between the geometric space E and the real set. It may also be seen as a data layer of the environment, but remains mathematically a function that maps a cell of the matrix E to a value in the field domain . In all the following, (k, l) denotes a pair of coordinates in the space E , it corresponds also to the indices of a cell in the linked matrix.

2.2.2 Main fields The environment is composed of the following main fields:

¹we will only consider symmetric TSP in this paper

The adjacency field \mathcal{A} corresponds simply to the connectivity matrix of the graph. The connectivity is complete here, \mathcal{A} is therefore full of 1 except on the diagonal.

The weight field stores data relative to the given distances in case of a TSP. We use a formula to transform all distances into the interval $[0, 1]$ so as to get a generic design, scale-free and independent of the units used:

$$\mathcal{W}(k, l) = \frac{\min_{(i,j) \in E} d(i, j)}{d(k, l)} \quad (2)$$

where $d(i, j)$ gives the distance of the edge (i, j) . The pheromone field \mathcal{T}^t is dynamically build by the ant colony. It is therefore a cumulative endogenous field: The pheromone field is initialized to 0. It is characterized by a cumulative updating process by ants and an evaporation coefficient ρ .

2.2.3 Other fields Other fields are needed in our environment modeling:

A field of visited vertices and edges denoted by \mathcal{H}_i^t stores, for an ant i at each time t during a global step of the algorithm, the taboo list of edges and nodes already visited by the ant.

A field of ordered edges \mathcal{O}^t maintains a list of edges related to each node of the graph, ordered by the amount of pheromone. This field is needed because of the deterministic nature of the choosing decision process of logistic ants (cf section 2.4.3).

A field of influence: the pheromone field is a dynamical field, and it has to be updated after each ant has achieved a hamiltonian cycle. That is why we involve also an ‘‘influence’’ pheromone field denoted $\tilde{\mathcal{T}}$, which is a temporary field for the global update of the pheromone field.

2.2.4 Final remarks The use of so many fields may be seen as a heavy way to describe the data structure, but it is a systematic way and a very generic means to describe the manipulated data whatever the considered environment might be. The field concept is moreover well adapted to system dynamics formulations.

The next section explains the chaotic ant system.

2.3 Design of the CAS

2.3.1 Rooting principles A chaotic ant is composed of an internal state with two components and a parametric chaotic map using these internal components.

Internal state definition of chaotic ants. Let us describe now the internal behavior of a chaotic ant. A chaotic ant i is a reactive agent with an internal state $s_i^t = \langle x_i^t, a_i^t \rangle$ at time step t where:

$x_i \in [0, 1]$ is the decision variable of the ant i ,

$a_i \in [0, 1]$ is the internal control variable of the ant i .

The interpretation of the internal variables becomes clear within the decision function of the ant-like agent, that is the function which gives the next value of x , computed by the following quadratic map:

$$x_i^{t+1} = f(x, a) = 1 - a(2x_i^t - 1)^2 \quad (3)$$

Here, both x and a belong to the interval $[0, 1]$: x is the main variable and a the control parameter of the map. For each a value, one plots the result of the computation of x after a fixed number of iterations and one obtains a bifurcation diagram (Fig. 1). This recurrence lead to three type of results: a fixed point, a periodic cycle or a chaotic (aperiodic) numerical series. We will use the characteristics of this map, which is governed by a single parameter a , in order to control the way a given chaotic ant choses the next node to visit.

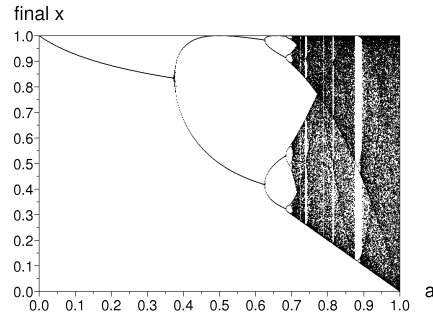


Figure 1. Bifurcation diagram for the quadratic map. Each point is plotted after 500 iterations.

Parametric control for exploration and exploitation.

More precisely the proposed algorithm consists in using the dynamical properties of the quadratic map — described in the bifurcation diagram— by modifying dynamically the a value as the algorithm runs and in respect to the pheromone field perceived locally, to achieve the final optimisation objective. Chaos occurs on the very right part of the diagram (Fig.1) that we will make correspond to the exploration phase, whereas fixed points occur on the left part of the diagram that we make correspond to the exploitation phase and the final convergence of the algorithm. This modulation of a is achieved during the perception phase (see the next section). Globally the algorithm leads at each local level of ants to converge from an exploration phase to an exploitation phase towards the best trajectories in the problem space, that is from a high a -value ($a \simeq 1$) to a low a -value ($a \simeq 0$). We may compare the control parameter a with a decentralized temperature driven by the local pheromone field.

2.4 Algorithmic process

The internal processing of a chaotic ant-like agent follows a sensorimotor scheduling —typical of a cybernetics approach—, that is a perception-decision-action process. This scheduling is achieved during an elementary time step of the algorithm, in parallel within each ant of the colony.

2.4.1 Perception process Let us consider an ant i on a given vertex k of the graph, let $V_i(k)$ denote the set of all vertices connected with vertex k and not yet visited by the ant i (not belonging to the taboo list of i). The perception operator acts on the pheromone field according to the formula:

$$P_i(k) = \max_{l \in V_i(k)} \{ \mathcal{T}(k, l) \} \quad (4)$$

This perception returns simply the maximum amount of pheromone from a given vertex.

2.4.2 Decision process The decision process performs the transition of the internal state of the ant i locally on the node k . This is formalized as a dynamical system between two time steps t and $t + 1$:

$$\begin{cases} a_i^{t+1} = \frac{1}{1 + e^{\alpha (P_i^t(k) - \tau_0)}} \\ x_i^{t+1} = f(x_i^t, a_i^{t+1}) \end{cases} \quad (5)$$

The updating of a_i regulates the adaptation behavior of chaotic ants by means of a sigmoid function which fixes the envelop of the decreasing variation of the control variable in function of the perception. Then the new decision variable is computed in respect to the just updated value of a . In the beginning of the algorithm, the pheromone field is low, consequently the a value equals 1 and the dynamics given by the quadratic map is chaotic. In the end, the pheromone field may be high, consequently the a value is low and the decision is always the same near a fixed point close to $(0, 1)$ in the bifurcation diagram. This summarizes the decentralized and distributed way this algorithm converges. The sigmoid function in this process enables to smooth the variations of the many dynamical modes generated by the bifurcations of the chaotic map.

2.4.3 Action process After updating the decision variable, each chaotic ant effects local actions:

Choose an edge among the ordered edges list (through the field \mathcal{O}) from the current vertex in proportion to the value of the decision variable x and move on the chosen vertex, denoted l^* .

Update the set of visited edges and vertices (the taboo list), that is the field \mathcal{H} .

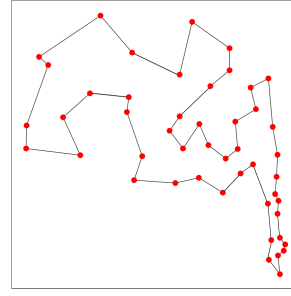


Figure 2. Chosen TSP instance from the TSPLIB Library. Optimal cycle equals 10628.

Update the influence pheromone field, that is the contribution part of the ant i to the pheromone field, by the following formula:

$$\tilde{\mathcal{T}}_i^{t+1}(k, l^*) = x_i^{t+1} \mathcal{W}(k, l^*) \quad (6)$$

2.5 Pheromone field updating

The updating of the pheromone field occurs once at the end of each time step when the N ant local loops are made up, and once at the end of each global step to reinforce the best cycle.

At the end of each time step the global pheromone field is updated with the contributions of all the colony according to the formula:

$$\mathcal{T}^{t+1} = \mathcal{T}^t + \sum_{i=1}^N \tilde{\mathcal{T}}_i \quad (7)$$

At the level of a global step, the pheromone field is updated according to the best cycle among the colony. Let L_i denote the distance of the ant i 's cycle and let G_{min} denote an inferior bound of the minimal distance for a cycle — it is defined by summing the minimal distances from every edges—. The amount of pheromone $\Delta\tau \in [0, 1]$ which reinforces the best cycle is given by:

$$\Delta\tau = \frac{G_{min}}{\min_i \{ L_i \}} \quad (8)$$

This formula is independent of the distance units used and does not depend on any parameter. The updating of the pheromone field according to the evaporation coefficient ρ is then very similar as the Ant system algorithm (?).

3 Simulation and results

Simulations have been performed on 20000 elementary time steps on the “att48” TSP instance from the TSPLIB library (?) for which the optimal cycle equals 10628 (cf. fig. 3).

Different ant number N and evaporation coefficient $\rho \in \{0, 0.01, 0.02, \dots, 0.1\}$ have been tested. We

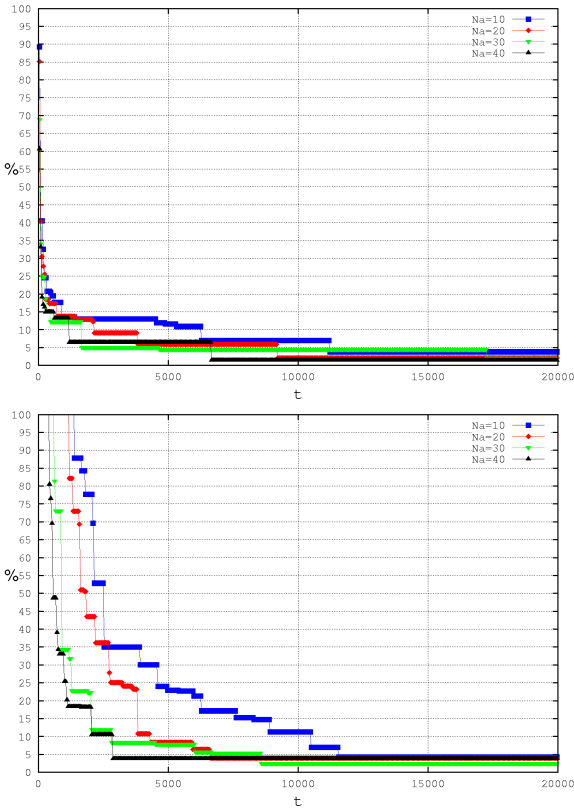


Figure 3. Evolutions of the best results (expressed in percentages of the optimal cycle) in function of time for the AS algorithm in the upper figure and for the CAS algorithm in the lower figure.

have compared the best results between the classical Ant System (AS) algorithm (?) and our chaotic ant system (CAS) on 5 runs for each initial configuration. The comparison of the best results is given in table 1. The sigmoid function used by chaotic ants has been fixed to the following parameter: $\alpha = 0,04$ and $\tau_0 = 30,0$. In terms of computation time, the running time of the

| Number of ants | AS | CAS |
|----------------|---------------|---------------|
| 10 | (0.0, 11028) | (0.01, 11074) |
| 20 | (0.06, 10845) | (0.0, 11049) |
| 30 | (0.06, 10847) | (0.02, 10894) |
| 40 | (0.08, 10777) | (0.01, 11026) |

Table 1. Comparison of best results between the “Ant System” (AS) algorithm and the Chaotic Ant System algorithm (CAS). Each cell gives a pair (ρ_{best}, L_{best}) .

CAS algorithm is in average twice the running time of the AS algorithm with the same hardware conditions (cf. fig. 3).

4 Discussion

Both algorithms have failed in finding the optimal solution in the limited laps of time. Some points have to be mentioned however:

1. Best results are very close for both algorithms, but CAS takes twice more computation time than AS.
2. The AS algorithm converges very fast towards good solutions, whereas it takes more time for the CAS algorithm to find good solutions. This feature of the CAS algorithm may be an advantage to avoid local optima.
3. The increase of the ant number has low benefits on the results in the AS case, whereas it speeds up the convergence in the CAS case.
4. The evaporation coefficient has to be near 0.1 to give the best results in the AS case, whereas much lower values are needed in the CAS case. This seems to make CAS algorithm less sensitive to the evaporation mechanism.

Consequently we consider that the performances of the chaotic ant system are sufficiently convincing to carry on developing this model for optimization. We foresee some advantages of this approach:

The deterministic nature of our chaotic ant model enables to make clear the underlying dynamical mechanisms involved. We have shown indeed that the pheromone field modifies the internal control variable of each ant-like agent and impacts its future decisions through a nonlinear decision map. The pheromone field reveals to be a decentralized control field in the control parameter space. This approach enable to follow the convergence process both locally and globally. In the end of the convergence, most of the chaotic ants are no more chaotic at all. They all get synchronized on the same path corresponding to the left part of their bifurcation diagram.

This model constitutes therefore a metaheuristic approach, because of the genericity of the convergence principle: the convergence profile is given for each ant by the bifurcation diagram of the chaotic map used. This convergence principle is simple and “graphical” on the bifurcation diagram of the map. Other maps may be used to improve convergence speed.

The CAS algorithm seems to be improved by the growing number of ants. This may be a good point in case of massive parallel processing.

The main drawback we may see in the CAS model lies in the theoretical impossibility to prove at the present time the asymptotic convergence to optimal solutions, instead of the original ant colony algorithm family, even if the theoretical proof does not help in practice when handled problems are very large. We intend to compensate for this drawback by keeping several permanent chaotic agents in the environment to maintain a minimum exploration level.

5 conclusion

We have shown in this paper a way to build a full deterministic model of ant algorithm by means of quadratic maps as decision functions within ant-like agents. The results obtained by the Chaotic Ant System reveal to be comparable to the Ant System algorithm on the same TSP instance, and this fact consequently validates our approach. The main advantage of this deterministic model lies in the mechanism principles it involves: the stigmergic process derives here from a internal parametric control of ants through the global field of pheromone, which is build from the cumulative contributions of all ants. This indirect parametric control operates in a distributed and decentralized way making a link between a bifurcation diagram and an optimization process. The whole ant colony ends up synchronizing on the same internal state and correlatively in the search space on the same good path of the TSP graph. This model constitutes an example of using a self-adaptive and dynamical control mechanism within a colony of ant-like agents in the field of optimization. This provides therefore a novel dynamical and distributed perspective on stigmergic processes involved in ant algorithms.