A study on ant colony systems with fuzzy pheromone dispersion

Louis Gacogne
LIP6
104, Av. Kennedy, 75016 Paris, France
gacogne@lip6.fr

Sandra Sandri
IIIA/CSIC
Campus UAB, 08193 Bellaterra, Spain
sandri@iiia.csic.es

Abstract

We describe here a sensibility analysis to help tune the parameters of an ant colony model where ants leave a fuzzy trace of pheromone to mark their track and its neighborhood. We test it with different parameters on some classical problems.

Keywords: Ant colonies, Optimization, Fuzzy Sets Theory.

1 Introduction

Ant colonies is a heuristic inspired from nature to solve optimization problems. When an ant finds a source of food, it leaves a trail of pheromone on its way back to the nest. The accumulation of small individual reinforcements on the paths allow an ant colony as a whole to find the best way between a source of food and the nest. When the source is exhausted, no new pheromone is deposited and the pheromone trails disappear as the pheromone slowly evaporates.

In the same way, ant colonies systems [3] find the optimal solution to a problem by the reinforcement of good solutions. In ant colonies systems, each ant (or its path) represents a solution, and weights assigned to each solution represent pheromone. A mechanism that increases weights on the solutions correspond to pheromone reinforcement and a mechanism that decreases weights correspond to pheromone evaporation. In some approaches, only one solution receives an increase of pheromone [4] on each turn, as if only the leading ant on a group would be allowed to deposit pheromone. In both approaches, randomly traces of individuals lead, after generations, to a best individual representing an optimal solution.

In [1], an ant colony system is presented in which an ant deposits pheromone not only on its path but also on the paths close to it. In this formalism, the ants are said to leave a *fuzzy trace*, and the closer a path is to the one that represents a good solution, the larger the amount of pheromone it receives. This is a direct analogy to the what happens in nature, as the chemical particles of pheromone disperse themselves in the air on a region around the deposit. The difficulty to use these formalisms in some optimization problems is to find a suitable notion of closeness among paths, such as the case of the Traveling Salesman Problem (TSP) [2].

This formalism is indicated for problems with large search spaces and has been employed successfully in a real world inverse problem, the estimation of chlorophyl profiles in offshore ocean water [6]. This problem is made harder by the fact that the cost of the computation of the goodness of each candidate solution is very high. The use of the ant colony systems with fuzzy pheromone dispersion allows the search space to be well explored with only a small amount of solutions being actually examined.

Here we are interested on the task of tuning the parameters of ant colony systems with fuzzy pheromone dispersion on some optimization problems. We address the solution of a symbolic problem, the Gauss Queens Problem, and of two numerical problems, using well-known functions in optimization literature, the Rastrigin and tripod functions. We focus in a dispersion func-

L. Magdalena, M. Ojeda-Aciego, J.L. Verdegay (eds): Proceedings of IPMU'08, pp. 812–817
Torremolinos (Málaga), June 22–27, 2008

tion that can be modeled by a linear fuzzy set and verify the relation between the number of ants and the radius of the dispersion.

This work is organized as follows. In Section 2 we describe the fuzzy pheromone dispersion algorithm used in this work. In Section 3, we present the results of the experiments made on the Rastrigin, the tripod and Gauss queens problems, and Section 4 brings the conclusion.

2 The algorithm for function optimization

In the following, we describe the fuzzy pheromone dispersion algorithm we use in the remaining of this text, based in [1]. For any positive function f from $[a, b]^n$ to $\mathbb{R}$, it aims at finding the arguments for f that lead to the global minimum.

An ant is characterized by its path, that represents its solution to the problem in hand. Each solution is a vector $v = (v_1, v_2, ..., v_n)$, where the i-th position contains the value for the i-th argument of f according to that solution. The path of an ant is a set of edges that codify the solution, i.e. each edge represents in fact an argument of f. The value for each edge on a path is taken from a discrete set, as described below. The quality of the solution described by vector v is simply calculated as $f(v)$.

Roughly speaking, in any generation, each ant chooses the edges of its path, according to probabilities that are proportional to the amount of pheromone deposited on them. At the end of this step, for each ant k, we take the solution $v(k)$ encoded by its path and calculate its quality $f(v(k))$. Then, each ant allowed to leave pheromone deposits an amount of pheromone on its path that is proportional to the quality of the solution associated to it and, moreover, it leaves smaller quantities of that amount on paths considered to be close to its own path.

This algorithm has thus two important aspects:

1. the better is an ant that is allowed to deposit pheromone, the bigger the trace it leaves;

2. the closer is an edge to the path of an ant that is allowed to deposit pheromone, the more pheromone this edge receives.

Let f be a positive function $[a, b]^n \to \mathbb{R}$, and let ϕ be the dilatation from domain $[0, 1]$ to domain $[a, b]$, defined as $\phi(x) = a + x \times (b - a)$. Let d be a positive integer; we create a discrete set of $d + 1$ values in $[0, 1]$ as $U = \{0, \frac{1}{d}, \frac{2}{d}, ..., 1\}$. Matrix τ contains the amount of pheromone on each edge at any moment of the execution of the algorithm: each position $\tau_{i,j}$ represents assigning the j-th value from U (i.e. $\frac{j}{d} \in U$) to the i-th component of the path. Thus the value stored in position $\tau_{i,j}$ (an amount of pheromone) is a weighing for the choice of taking value $\phi(\frac{j}{d})$ as the i-th argument of f. Here, contrary to the approach used in [1], the initial values stored in matrix τ are obtained at random.

As said previously, an ant deposits a certain amount of pheromone on its path and smaller quantities of this amount on edges close to its path. Here, the notion of closeness is implemented using the discretization in U; the edges considered to be close to an edge (i, j) are those in set $\{(i, j') \mid (|j - j'| \leq dm) \wedge (\frac{j'}{d} \in U)\}$, where the neighborhood radius $dm \in \mathbb{N}$ is a reasonably small natural number ($0 \leq dm << \frac{d}{2}$). Therefore, given that (i, j) is in the ant's path, an edge (i, j') will receive extra pheromone if $\frac{j'}{d}$ is one of the dm values either below or above $\frac{j}{d}$ in set U. Note that usual scheme (no pheromone dispersion) is modeled by $dm = 0$.

We use a fuzzy set G with membership function $G : \mathbb{N} \to [0, 1]$ to model the notion of dispersion of pheromone on the edges in the neighborhood of a path: the closer an edge to the path, the larger its membership to G and the larger the extra quantity of pheromone it receives. Various types of convex fuzzy membership functions can be used for G. In the algorithm below, we use a triangular fuzzy set $< c - dm, c, c + dm >$, with core equal to point value c (i.e., $G(c) = 1 \wedge \forall x \neq c, G(x) < 1$) and support given by values of $\mathbb{N}$ in interval $[c - dm, c + dm]$ (i.e., $\forall x \in [c - dm, c + dm], G(x) > 0$). Thus we have $\forall k \in \mathbb{N}, G_{c,dm}(k) = max(0, 1 - \frac{|c-k|}{dm})$.

In this work we address two mechanisms for the deposition of pheromone. In the first mechanism, all ants deposit pheromone in their corresponding paths and neighborhoods. In the second mechanism, only the ant with the best solution in each generation is allowed to deposit pheromone (this ant can be thought of as the "leader" of its generation).

In algorithm FAC described bellow, each generation consists of a set of m ants. We also have:

- τ is a matrix $n \times (d+1)$ of float numbers, representing all the possible edges.
- $ants$ is a $m \times n$ matrix in $\mathbb{N}$, such that each row contains the path of one of the ants considered in a generation.
- val is a vector of length m, containing the values $f(v(k))$, where $v(k)$ corresponds to the solution obtained through the k-th ant, i.e. $f(v(k)) = f(\phi(\frac{ants_{k,0}}{d}), ..., \phi(\frac{ants_{k,n-1}}{d}))$.
- vm is the best value of val and nv is the number of evaluations of function f.

Algorithm $FAC(m, d, dm, \alpha, \rho, eval_{\max}, \epsilon, tts)$

- m, d and dm are defined as above;
- α and ρ are ant colony parameters indicating the accentuation of probabilities and the evaporation rate, respectively;
- $eval_{\max}$ is the an upper bound for number of evaluations of function f;
- $\epsilon > 0$ is a lower bound for function f;
- tts is a boolean variable; if $tts = true$ then all ants leave pheromone, if $tts = false$ only the leader does it.

Initialize matrix τ with a random value in]0,1]. Initialize vm and nv with 0.

While $vm > \epsilon$ and $nv < eval_{\max}$ do

1. Creation of ant paths:
   for $i = 0$ to $n-1$ do
   - let $normp$ be the normalized vector of probabilities obtained from $(\tau_i)^\alpha$.
   - for $k = 0$ to $m-1$ do create $ants_{k,i} \in U$ according to probabilities $normp$.
2. Evaluation:
   for $k = 0$ to $m-1$ do $val(k) \leftarrow f(v_k)$;
   let vm be the lowest value in val and b the ant evaluated with vm (the best ant).
3. Evaporation:
   matrix τ becomes the product of itself with $(1-\rho)$.
4. Pheromone Updating:
   if tts then *(* updating with all ants *)*

   for $k = 0$ to $m-1$ do

   for $i = 0$ to $n-1$ do

   $x \leftarrow ants_{k,i}$

   for $j = \max(0, x - dm)$

   to $\min(d, x + dm)$ do

   $\tau_{i,j} \leftarrow \tau_{i,j} + \frac{G_{x,dm}(j)}{val(k)}$

   else *(* updating only with the best ant b *)*

   for $i = 0$ to $n-1$ do

   $x \leftarrow ants_{b,i}$

   for $j = \max(0, x - dm)$

   to $\min(d, x + dm))$ do

   $\tau_{i,j} \leftarrow \tau_{i,j} + \frac{G_{x,dm}(j)}{vm}$
5. Incrementation of nv with $nv + m$.

At the end of an execution, the algorithm delivers the best value $(v, f(v))$ and the number of evaluations of f taken to reach it.

3 Tests and results

In the following we present sensibility analyses on the parameters of the algorithm above regarding three problems, two numerical (Rastrigin and tripod functions) and one symbolic (Gauss queens problem). For all the problems, the results presented are the average of 100 trials. Unless stated otherwise, we have used evaporation rate $\rho = 0.1$ and accentuation coefficient $\alpha = .5$.

In all our experiments, the scheme in which only the best ant leaves pheromone fared much worse than the case where all ants leave pheromone and, for this reason, in the figures we only illustrate this second scheme ($tts = true$).

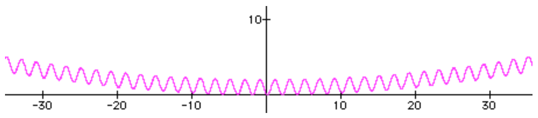

Figure 1: Rastrigin function (single dimension).

3.1 Experiments with the Rastrigin function

The Rastrigin function is defined as

$$f_R(x) = \sum_{i=1,n} [x_i^2 + 10 - 10 \times cos(2\pi x_i)]/400,$$

where $x \in R^n$ (see Figure 1 with $n = 1$).

We performed experiments for this function in dimension 1 with $x \in [-30, 30]$. Using a discretization factor of $d = 100$, the optimal value for f_R is 10^{-4}, reached with $x = 0$.

3.1.1 Neighborhood radius

We studied the behavior of the neighborhood radius dm for function f_R; Figure 2.a (respec. Figure 2.b) depicts the average number of evaluations taken to reach the optimum using between 50 to 250 (respec. 2 and 30) ants. We can see that the best results are obtained with a very small number of ants, and that there is no need to use radius values larger than 8.

In these experiments, the average number of evaluations taken to reach the minimum in the case of no fuzzy trace use ($dm = 0$) is about twice as high as the ones using fuzzy trace, and for this reason this case is not shown in the graphics.

3.1.2 Evaporation rate

In each generation, the pheromone is decreased losing a part ρ of it. Figure 3 shows the average number of evaluations to reach the optimum with $dm = 10$, varying ρ in $[0, 0.9]$. The best results are always roughly around $\rho = 0.1$ to 0.4.

3.1.3 Accentuation of probabilities coefficient

We varied the accentuation of probabilities parameter α for the Rastrigin function in dimension 1, using $dm = 10$. The experiments confirm that the best values are around $\alpha = 0.5$ (see Figure 4).

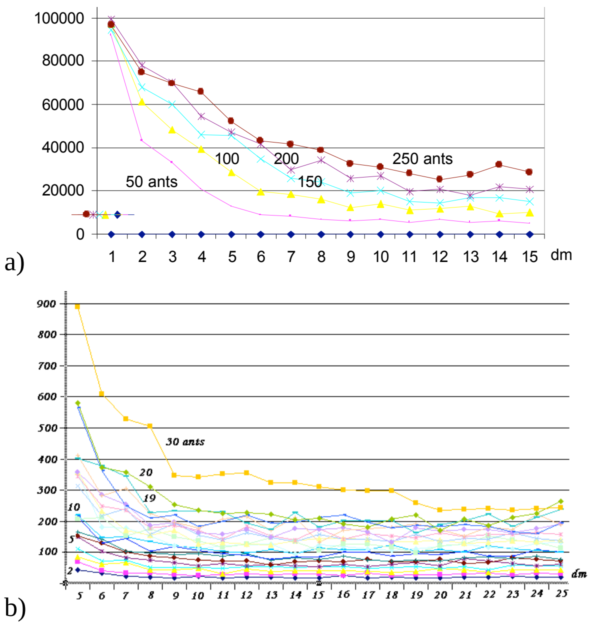

Figure 2: Number of evaluations per neighborhood radius for the 1-D Rastrigin function: a) 50 to 250 ants and b) 2 to 30 ants.

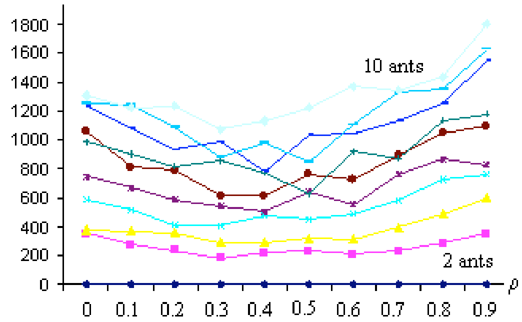

Figure 3: Number of evaluations per evaporation rate for the 1-D Rastrigin function.

3.2 Experiments with the tripod function

The tripod function is defined as:

if $y < 0$

then $f_T(x, y) = |x| + |y + 50|$

else if $x < 0$

then $f_T(x, y) = 1 + |x + 50| + |y - 50|$

else $f_T(x, y) = 2 + |x - 50| + |y - 50|$.

Although very simple, this function has three minima (see Figure 5), and is quite difficult to minimize with stochastic heuristics [5].

Figure 6 shows the number of evaluations taken

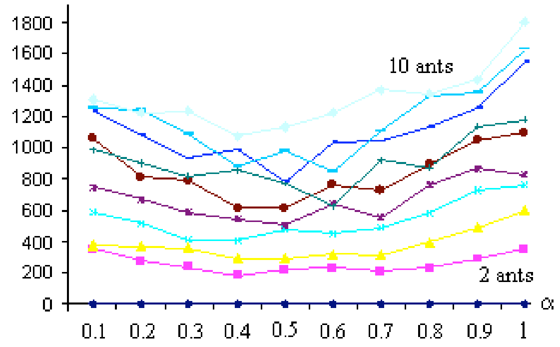

Figure 4: Number of evaluations per accentuation of probabilities for the 1-D Rastrigin function.

by the algorithm to reach the optimum for the tripod function with x and y in domain $[-100, 100]$ and discretization factor $d = 200$, varying the number of ants and the neighborhood radius dm.

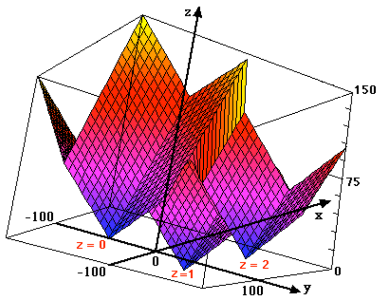

Figure 5: Tripod function for $x, y \in [-100, 100]$.

The best results were obtained with a large number of ants (from 100 to 500) and relatively small values for dm (between 2 and 6). With dm greater than 7, results from 50 to 1000 are quite similar. The use of a small number of ants (2 to 10) did not yield good results, particularly for small values of dm (between 0 and 7).

3.3 Experiments with a symbolic problem

The Gauss queens problem consists in placing n queens on a $n \times n$ chessboard, in such a way that the queens would not be able attack one another, considering the movements the queen is allowed to take in chess.

The possible configurations can be represented by a sequence of n values taken from the set $\{1, 2, ..., n-1\}$, where the value in the i-th position in the sequence indicates the row of the queen

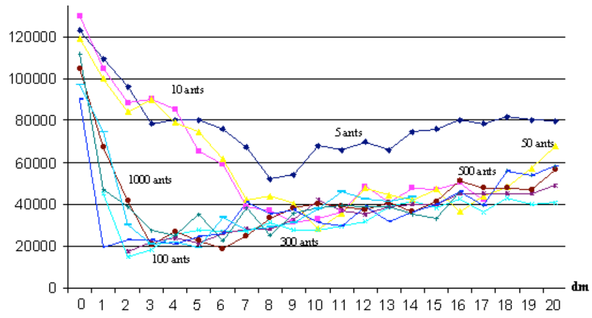

Figure 6: Number of evaluations per neighborhood radius for the tripod function: 5 to 500 ants.

in the i-th column. A set of solutions for n from 4 to 9 are for instance (see Figure 7):

chessboard $4 \times 4 \rightarrow$ (2 4 1 3)
chessboard $5 \times 5 \rightarrow$ (3 1 4 2 5)
chessboard $6 \times 6 \rightarrow$ (3 6 2 5 1 4)
chessboard $7 \times 7 \rightarrow$ (3 1 6 2 5 7 4)
chessboard $8 \times 8 \rightarrow$ (6 3 1 8 4 2 7 5)
chessboard $9 \times 9 \rightarrow$ (4 1 7 9 2 6 8 3 5)

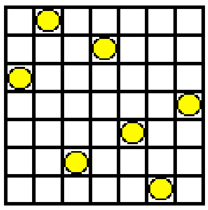

Figure 7: A solution for the 7×7 queens problem.

A solution to the problem can be found by minimizing function f_{gauss}, that computes the number of couples of queens in catching positions on the chessboard. For example, $f_{gauss}(1, 3, 2, 5, 5) = 4$ and $f_{gauss}(2, 7, 5, 3, 1, 6, 4) = 0$. The ants algorithm generates n-dimensional vectors in $[1, n]^n$ and so we have used discretization factor $d = n - 1$.

In Figures 8.a and 8.b, we see that for a 7×7 chessboard, the best results are obtained with a small number of ants and with radius around $dm = 4$ (values for $dm = 0$ and $dm = 1$ are too large and are not plotted).

We see that the use of a larger number of ants does not yield better results, but that even then the best value for dm is still around 4. Note also that a very small number of ants (2 to 10) give results worse than for a large number of ants (100 to 350) in the case of no pheromone dispersion ($dm = 0$).

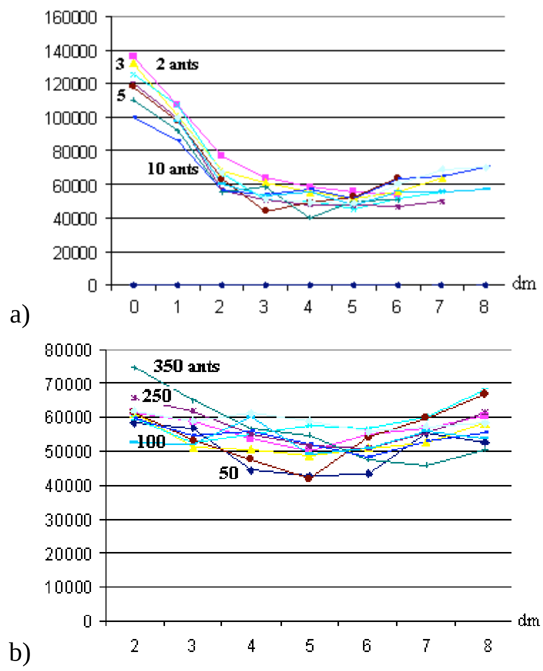

Figure 8: Number of evaluations per neighborhood radius for the 7×7 chessboard f_{gauss} function: a) 2 to 10 ants and b) 50 to 500 ants.

4 Conclusion

We presented a study on the tuning of parameters of ant colony systems with fuzzy pheromone dispersion on some optimization problems. In this algorithm, the closer is an edge to the path in which an ant deposits pheromone, the more pheromone this edge shall receive. Moreover, the amount of pheromone deposited by an ant is proportional to the quality of the solution it represents.

Using a a linear fuzzy set we verify that usually only a relatively small number of ants and small values for the neighborhood radius that characterizes the dispersion are needed to obtain good results. In all experiments, no dispersion of pheromone ($dm = 0$) always yielded inferior results. The scheme in which all ants deposit pheromone showed much better results than the one in which a single ant is allowed to do so.

In the case of Rastrigin function, using here a very small search space, the best results were obtained with the number of ants varying between 2 and 20 and with neighborhood radius around 8. For the tripod function, with a search space of around 40.000 points, one obtains good results with 50 to 100 ants and dm between 2 and 6. For the Gauss problem with 7 queens, with a search space of 7^7 points (> 800.000), the best results are obtained with a small number of ants (2 to 10) and dm around 2 to 6.

As future work, we intend to extend the sensibility analyses. In particular, we would like to verify the sensibility of the algorithm in relation to other types of fuzzy set shapes for pheromone dispersion, such as the Gaussian curve.

Acknowledgements

This work has been partially supported by MULOG2 (TIN2007-68005-C04-01/02) and IEA (TIN2006-15662-C02-01/02) projects and by the Generalitat de Catalunya under grant 2005-SGR-00093.

References

[1] J. C. Becceneri, S. Sandri (2006). “Function optimization using ant colony systems with pheromone dispersion”. In *Proc. IPMU’06*, Paris (Fr), July 2006.

[2] J. C. Becceneri, S. Sandri, E.F.P. Luz (2006). “Ant colony systems with pheromone dispersion for TSP”. In *Proc. 2nd LNCC Meeting on Computational Modelling*, Petrópolis (Br), Aug. 2006.

[3] E. Bonabeau, M. Dorigo, and G. Theraulaz (1999). *From natural to artificial swarm intelligence*. Oxford University Press.

[4] M. Dorigo, G. Di Caro, and L.M. Gambardella (1999). “Ant algorithms for discrete optimization”. In *Artificial Life*, Vol. 5, No. 3, pp. 137-172.

[5] L. Gacogne (2006). “Methods to apply operators in a steady state evolution algorithm”. In *Proc. IPMU’2006*, Paris (Fr), July 2006.

[6] R.P. Souto, H.F. de Campos Velho, S. Stephany, S. Sandri. “Estimating vertical chlorophyll concentration in offshore ocean water using a modified Ant Colony System”. *Submitted*.