Simulation of Artificial Ant's Behavior in a Digital Environment

Paulo E. Merloti
padu@ieee.org
Department of Computer Science
Graduate Seminar in Artificial Intelligence
Evolutionary and Adaptive Computation

Abstract. Ants present a very good natural metaphor to evolutionary computation. While their individual computational power is small compared to more evolved species, it is the power of their colonies that inspire computer scientists. This paper presents "Ants Box Simulator", software that allows the simulation of ants in a restricted environment. We demonstrate how some parameters affect their performance on finding a solution to a given problem.

1 INTRODUCTION

Ants are said to be very simple beings. With their small size and small number of neurons, they are not capable of dealing with complex tasks individually. The ant colony in the other hand is many times seen as an "intelligent entity" for its great level of self-organization and the complexity of tasks it performs. In this paper, we will focus on one of the resources ant colonies use for their achievements, pheromone trails.

One way ants communicate is by using chemical agents and receptors. For example, one ant is capable of distinguishing if another individual is a member of its own colony by the smell of its body. One of the most important of such chemical agents is the pheromone. Pheromones are molecules secreted by glands on the ant's body and once deposited on the ground, they start to evaporate. Ants inside the pheromone's cone of evaporation [1] are attracted by that pheromone. Foraging ants leave a trail of such scent, which stimulates the stigmergetic behavior of other ants to follow that trail and drop pheromones while doing it so. Such autocatalytic process will continue until a trail from the ant colony to the food source is established. Ants don't have the goal to create a trail that has shorter distance from nest to food source. Their goal is to bring food to the nest, but most of the time the pheromone trails they create are highly optimized.

In laboratories, several studies have explored how pheromones are used by ants. In a very inspiring experiment, Deneubourg et al. [3] demonstrated how ants use pheromones to optimize the roundtrip time from nest to food source. In this experiment denominated "The double bridge", ants in the nest are separated from the food location by a restricted path with bifurcations that lead to branches of different lengths. As time goes by, the experiment shows that pheromone concentration is higher on the shorter path, and consequently almost no ant chooses the branch that is not the shorter one.

Although, it is naïve to think that pheromones are the only resource ants use to navigate. In another experiment, Bethe [4] proves that pheromone trails are not polarized, as everybody thought it was at that time. The experiment consists of a nest, a food source and a pivot table in between food location and the ant's nest. After a pheromone trail is formed over the pivot table, an ant is released from the nest. While the ant is following the pheromone trail to the food location, the pivot table is turned 180º. If the ant keeps its direction, it would end up in the nest again, but amazingly, the ant also turns its direction and end up in its original destination. This experiment demonstrated that ants also depends on other senses to navigate, such as position of the sun in the sky (or a strong enough light source), gravity, slope and reference objects.

1.1 Existing work

It seems like the Artificial Intelligence community has been shifting its paradigms to techniques based on evolutionary computation. Inspiration comes from several natural fields such as Genetics (GA), Metallurgy (Simulated Annealing) and Mammal Immune System (AIS). Ant Colonies and Swarm algorithms are one more paradigm of such algorithms.

Marco Dorigo leads the research on Optimization techniques using Artificial Ant Colonies [5]. Since 1998, Dorigo organizes a biannual workshop on Ant Colony Optimization and Swarm Algorithms at the *Université Libre de Bruxelles*. Victorino Ramos [6] and Semet [7] use the Ant Colony approach to perform image segmentation. Heusse et al. [8] applies some concepts of Ant Colonies on routing of network packages. A more detailed summary of these studies can be found on a summary paper from this author in [9].

1.1 Introduction to Ants Box Simulator

The *Ants Box Simulator* idea started with the author's curiosity about ant behavior and their amazing feats. The concept of the simulator is of a two dimensional digital box where ants are represented by small objects that move around a delimited area in a random fashion. They are inserted in the environment around a spot denominated "IN Hole" and their goal is to find the "OUT Hole". Ants have a limited sensorial radius to "smell" pheromones or to detect the OUT Hole. The simulator is essentially graphical, and the probability of one ant finding the exit of the box is directly related to screen resolution (area of the box) and the distance from IN and OUT holes.

The next sections of this paper demonstrate a series of experiments using the simulator by varying parameters and strategies that affect ant's behavior.

2 ARE PHEROMONES GOOD ENOUGH?

As stated previously, ants heavily rely on pheromones to guide themselves. In computer science literature, most of Ant Colony algorithms utilize the pheromone analogy together with a highly parallel architecture to solve hard problems. It is true that being computer scientists seeking inspiration to solve engineering problems, we don't need to be locked on a high fidelity model of an ant colony, but it is very important to keep in mind that our inspiration comes from a non-complete biological model of how ants navigate.

The fundamental question we try to answer here is: "Can we solve the problem or searching for the OUT hole using only the pheromone concept?" We know that this model is a simplification of how real ants navigate, but is it enough to use only pheromones to solve the proposed problem?

We know that in other lines of research, the pheromone concept has proven useful, achieving excellent results in combinatorial optimization problems for example [5].

3 SIMULATOR ARCHITECTURE

The Ants Box Simulator is an application that runs on Windows 32 platform. Due to its demand for graphic computation, a Pentium IV or greater with a fast video card is recommended. Also, it relies on platform specific DirectX technology, version 7 or higher. The software was developed using Borland's Delphi, a Pascal based language. The main reason for this choice was high productivity and fast performance of native code offered by this tool.

The application was built with expansion in mind. An object model was built in order to implement not only the Ant concept, but a digital environment where objects and even other insects could be placed together in later experiments. Following is a brief explanation of such object model.

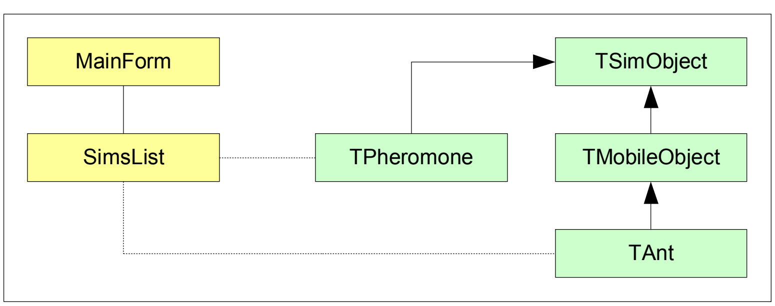

Fig. 1: Ants Box Simulator Object Model

The *Main Form* is responsible for drawing, creating and managing all kinds of simulation objects, as well as presenting a user interface for interaction with parameters. It keeps a dynamic list of simulation objects on *SimsList*. All simulation objects are descendants of *TSimObject*. *TSimObject* is responsible for basic aspects of the object, such as position, size, identification and a virtual method for drawing. A *TMobileObject* implements basic animation methods such as wall collision, collision with other objects, current speed and direction. An ant (*TAnt*) is a specialization of a *TMobileObject* that refines and implements navigation techniques using stochastic patterns and pheromone following algorithms. Ants may release pheromones (TPheromone) into the environment. Pheromones have the property to expire after a certain amount of time, and their strength is proportional to their remaining life time.

There are two types of ants in the Ants Box Simulator, Regular Ants and Searching Ants. When regular ants find the OUT hole, they exit no matter what. Searcher ants will always leave a pheromone trail, and if the OUT hole is found, then they will trace back their trail reinforcing their own pheromones (positive feedback). If they find a wall instead, they will destroy their trail (negative feedback).

4 EXPERIMENT DESIGN

In this section we will describe the experiments we performed using the Ant Box Simulator. This paper is not a full presentation of how the Ant Box Simulator was created, but in order to fully understand the experiments and its results, we feel that a brief introduction on the models applied to the simulator engine is necessary.

The digital environment where the experiments were performed is a two dimensional array of pixels representing the surface of the box with an approximate area of 644,496 squared pixels. The IN hole is located at the center of the box, and the OUT hole is located at the right bottom corner, with a distance of 535 pixels. Experiments are run for 300 seconds (5 minutes). Performance is measured by comparing the number of ants in the box during the time allotted for the experiment. All the experiments start ticking the clock when the first batch of 100 regular ants is released through the IN hole.

Our first experiment (*xp1*) is a random search; an initial batch of 100 ants is released into the environment to run their random walk. Some of the ants will find the OUT hole by chance. *xp1, xp2* and *xp3* were performed 3 times each. In *xp2*, batches of 10 searching ants are released 10 seconds apart, until 250 seconds of box time. Experiment 3 (*xp3*) is very similar to *xp2*, except that it releases only one searching ant every 10 seconds until the 250 seconds mark is reached. Experiment 4 (xp4) and 5 (xp5) are based on *xp2* and *xp3* respectively with the addition that regular ants will also drop pheromones when they find a pheromone trail. For reasons to be seen later, these two experiments were run only one time each.

4.1 Parameter Definition

Table 1 offers a brief description of each parameter and the values defined for these experiments.

Parameter	Description	Value
Ant direction max variance (***maxVar***)	This parameter imposes a limit for the random choice of an ant's new direction.	80º (-40º to +40º)
Pheromone duration	Gradient of pheromone expiration. Pheromones are stronger when they are just released and weaken towards its expiration.	30 sec.
Pher. Reinforcement	If a pheromone is reinforced, this parameter indicates how many more seconds its expiration is postponed, consequently increasing its strength.	40 sec.
Sensorial radius	Ants have a limited range of space around them that can be sensed. This parameter defines such radius.	45 pixels
Searching radius	Searching ants are allowed to have a different sensorial radius than regular ants.	90 pixels
Walk before pause (***wbp***)	Defines a limit for segment lengths of straight walk before calculating a new random direction to take. Segment lengths are randomly drawn every time a new direction is chosen.	45 pixels
Searcher follows reinforced trail	As it will be presented on the conclusions of this paper, this is one of the fundamental parameters for success. If enabled, ants searching for the exit will give up their random walking pattern in favor of reinforced trails within their sensorial radius.	True
Drop pheromone while following trail (**drop_pher**)	Experiments 4 and 5 (*xp4, xp5*) are designed to test this parameter. If enabled, regular ants will drop pheromones if they are following a trail.	Depends on experiment
Stochastic pheromone following	If enabled, ants will navigate using a probabilistic method to decide which direction to take if pheromones are within their sensorial radius. If disabled, ants will always take the direction of the stronger pheromone around.	True

Table 1: Ants Box Simulator main parameters

4.2 The Mathematics of Navigation

Before we proceed to the experiments, let us explain one very important aspect of our digital ants, how navigation and pheromone following is implemented. When ants are released into the box, they immediately start their random walk inspired on *"Brownian movement"* [10]. Ants will follow a straight line of random length (limited by ***wbp*** parameter) by some initial random direction. When they reach the end of this segment, a new direction is chosen according to the following formula:

$$NewDirection = CurrentDirection + \theta \quad (1)$$

Where

$$\theta = \text{Random}(maxVar) * \text{Random}(-1:1) \quad (2)$$

When one or more pheromones are within the ant's sensorial radius, the new direction is calculated using a stochastic pheromone following technique described below.

From the ant's point of view, there are three attributes related to a pheromone: direction, distance and strength. Biological ants don't have perfect sensors, and natural factors may contribute to irregular pheromone sensing (i.e. wind), therefore a mechanism to introduce some noise is necessary. The second supposition is that ants have a better sense for pheromones if they are in front of them, therefore a pheromone that is aligned with the ant's current direction have a greater probability to affect the ant's decision on choosing the new direction than one located somewhere behind the ant.

Figure 2 depicts a situation where the given ant detects three pheromones within its sensorial radius. Each pheromone vector is attenuated and/or expanded given its strength (diameter of the blue dots) and the angle between the ant's current direction and pheromone direction. The new calculated direction is the sum of all of these pheromone vectors.

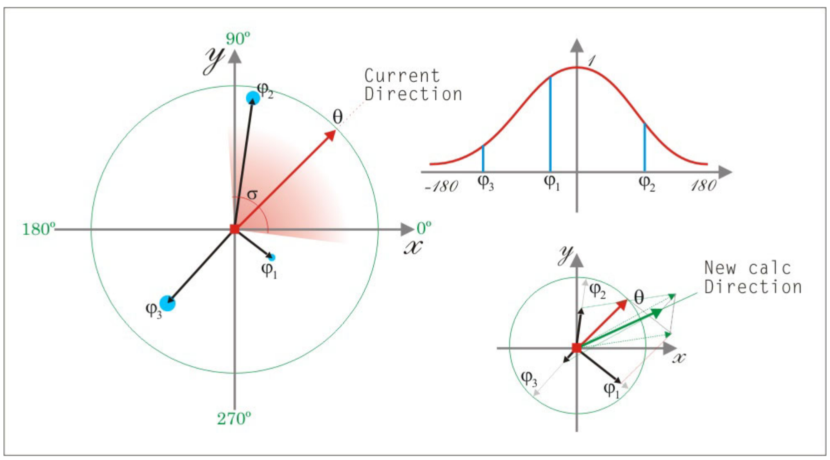

Fig. 2: Illustration on how pheromones affect the ant's choice for a new direction

Also, in order to simulate the uncertainty factor found on nature, we must introduce some random noise into the system. The new direction can be calculated as:

$$d_{new\_dir} = d_{calc} + d_{noise} \quad (3)$$

$$d_{calc} = \sum_{i=1}^{N} \varphi_i . s_i \quad (4) \text{ and } d_{noise} = \rho . e^{-\frac{1}{2}\left(\frac{2\rho}{\sigma}\right)^2} \quad (6), \textit{where}$$

$$s_i = \text{strength}(\varphi_i).e^{-\frac{1}{2}\left(\frac{2\delta}{\sigma}\right)^2} \quad (5)$$

σ is the **maxVar** parameter, ρ is a random angle in the range [random(360)-180] and δ is the angle between φ_i and the current direction.

5 EXPERIMENTAL RESULTS

As stated somewhere else in this paper, the ants of the following experiments have a primary goal of finding the exit of the box in the least amount of time. The evolutionary goal is to create a near optimum (a straight line) pheromone trail from the IN hole to the OUT hole. This way ants inserted after the trail was established would have a reference on how to get out of the box as soon as they enter it.

Figure 3 shows a graphic comparing the results of such experiments designed in section 4. Table 2 helps summarizes what attributes distinguishes one experiment and the other.

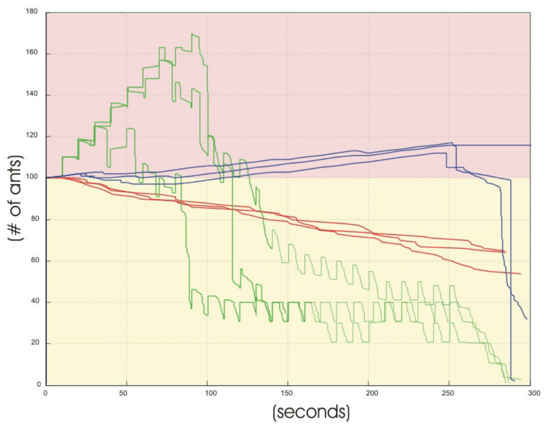

Fig. 3: Results of experiments. Red is random search (xp1), green is xp2 and blue is xp3

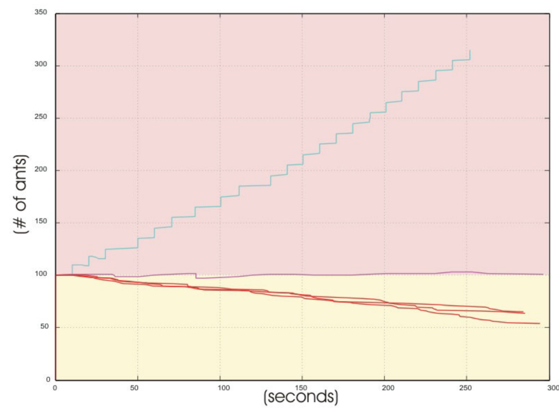

Fig. 4: Results of experiments xp4 (cyan) and xp5 (magenta). Red is random search (xp1).

Experiment	Description	Color
XP1	Random Search	Red
XP2	Batches of 10 searching ants every 10 seconds up to t=250	Green
XP3	One searching ant every 10 seconcs up to t-250	Blue
XP4	XP2 + Regular ants drop pheromones while following trail	Cyan
XP5	XP3 + Regular ants drop pheromones while following trail	Magenta

Table 2: Summary description of experiments

Two aspects must be considered when analyzing the performance of a given experiment. One is how soon the exit if found by a considerable number of ants, observed in the graph as a big drop in the number of ants. The second aspect is the dynamics of the number of ants in the arena. In random search (*xp1*) for example, the total number of ants never went higher than 100 for obvious reasons (no searching ant is ever introduced).

Experiment 2 (in green) was the best performing one if we are after shorter time to find the exit. As observed, at earlier steps a greater number of ants will be on the box due to the introduction of the extra searching ants. Another point observed in this experiment is that even after the exit was found, there were a good number of ants still on the box, mainly doing path maintenance. This experiment also demonstrated that as long as searching ants are kept doing trail maintenance, it will slowly converge to the quasi-optimum solution, a straight line from the IN box to the OUT box.

Experiment 3 (in blue) tells us that if we reduce the input flow of searching ants, it takes much longer to find and establish the trail to the OUT box, but it is the more economical solution in terms of number of ants in the box. In this case, the solution was found at the fringe of the 5 minutes time limit, and only in two of the three repetitions of the experiment. One should always keep in mind that as the number of searching ants increase, so does the number of pheromones in the box and consequently the computation cost of the experiment.

When we allow non-searching ants to release pheromones when they find a trail (enable parameter **drop_pher**), the situation goes out of control. Ants will tend to form circular groups, where one ant follows the fresh pheromone of the other and vice versa. At this point, pheromone starts to be deposited at very high rates and results are even worse than random search. In random search, ants are free to randomly walk the box. Figure 4 shows the results of *xp4* and *xp5*. Note on *xp4* how the number of ants in the box grows indefinitely as no ant is able to find the exit of the box.

The Ant Box Simulator can be found on the Author's webpage at http://www.merlotti.com/EngHome/Computing/AntsSim/ants.htm, including the source code in Delphi 6.

Figure 5 shows a screenshot sequence of *xp2* demonstrating the patterns on how a trail is created. Figure 6 shows *xp4* and the circular grouping pattern.

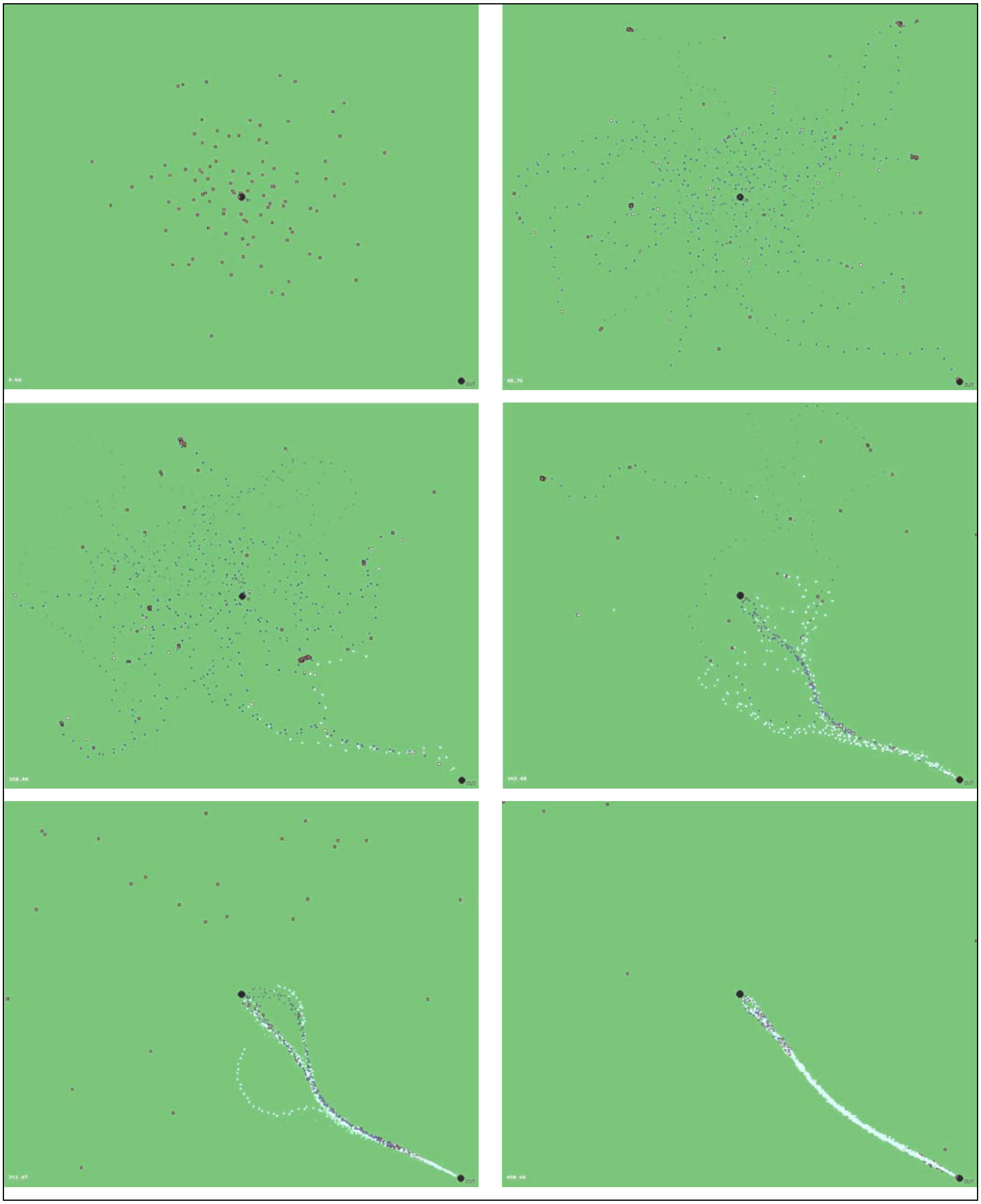

Fig. 5: Screenshot sequence about Experiment 2

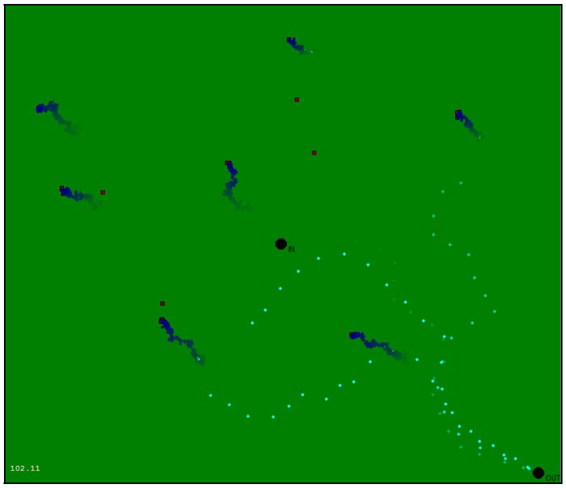

Fig. 6: Screenshot of experiment 4, ants start on a circular pheromone following

6 CONCLUSIONS

In this paper we introduced the idea of an Ant Box simulator, a software program that simulates the autocatalytic behavior of ants when faced to an artificial problem, finding the exit of a two dimensional box.

After running the experiments, we were able to show that pheromones are really useful if used together with a "good" strategy. Also, we were able to see that different strategies may serve different purposes.

As in other studies about ant algorithms, our artificial ants use only the pheromone concept to guide themselves, and as we saw on *xp4* and *xp5,* this approach can lead to catastrophic behavior sometimes. The main reason why *xp4* and *xp5* experiments failed is related to the discussion of section 1. Biological ants don't rely only on pheromones to navigate. It would be interesting to research the creation of a framework of ant colony algorithms that include methods inspired on other resources used by biological ants such as gravity, light sources and vision.

To conclude this paper, we share our feeling that the toughest problem we faced dealing with evolutionary techniques such as ant algorithms is finding the right parameters in order to direct the system to solve a specific problem. If we find a way to pressure the population of such system to change its own parameters and naturally evolve into a body capable of solving a specific problem, then our task would be defining problems in such ways that would be understandable for our population. Perhaps genetic algorithms would be a good approach. Would it then materialize our dreams of a machine capable of solving problems with no necessity of being pre-programmed? Would it eliminate the brittleness problems found on many approaches to artificial intelligence?

These questions are the main focus of research on many AI studies, and in the author's point of view, biological inspired ideas have a great probability of success; after all, for many problems we still don't know how to solve using machines, nature has proven methods that work everyday almost effortlessly.

References

1. Matthews, R. W. & Matthews, J. R., *Insect Behavior*. Wiley-Interscience, University of Georgia, New York, 1942.

2. Gordon, D., *Ants at Work: How an Insect Society is Organized*. The Free Press, New York, 1999.

3. Goss, S., Aron, S., Deneubourg, J. L., Pasteels, J. M., *Self-organized shortcuts in the Argentine ant*. Naturwissenschaften, 76:579–581, 1989.

4. Bethe, A., *Recognition of nestmates, trails.* Arch. Gesamt. Physiol., 70, 15-100, 1898.

5. Dorigo, M., Di Caro, G., Gambardella, L. M., *Ant Algorithms for Discrete Optimization,* Artificial Life, Vol.5, No.3, pp. 137-172, 1999.

6. Ramos, V., Almeida, F., *Artificial Ant Colonies in Digital Image Habitats – A Mass Behavior Effect Study on Pattern Recognition.* Proceedings of ANTS'2000, 2nd International Workshop on Ant Algorithms, pp. 113-116, Brussels, Belgium, Sept. 2000.

7. Semet, Y., O'Reilly, U., Durand, F., *An Interactive Artificial Ant Approach to Non-Photorealistic Rendering.* Springer-Verlag, K. Deb et al. (Eds.): GECCO 2004, LNCS 3102, pp. 188-200, Berlin, 2004.

8. Heusse, M., Guérin, S., Snyers, D., Kuntz, P., *Adaptive Agent-driven Routing and Load Balancing in Communication Networks.* Technical Report RR-98001-IASC, Départment Intelligence Artificielle et Sciences Cognitives, ENST Bretagne, 1998.

9. Merloti, P., *Optimization Algorithms Inspired by Biological Ants and Swarm Behavior.* San Diego State University, Artificial Intelligence Technical Report – CS550, San Diego, 2004.

10. Ford, B. J., *Brownian Movement in Clarkia Pollen: A Reprise of the First Observations.* The Microscope, Volume 40, Fourth Quarter: 235-241, Chicago, Illinois, 1992.