# The Incidence of the Controllable Parameters in Routing Algorithms Based on Ant Colony Optimization

**Rolando F. Vallejo**
ITESM-CCM
rovallej@campus.ccm.itesm.mx

**Edgar Vallejo, Roberto Gómez**
ITESM-CEM
{evallejo,rogomez}@campus.cem.itesm.mx

## Abstract

Routing algorithms are used to find routes which are helpful to establish efficient data communication in computer networks. As the size of the network grows, finding routes becomes more complex in a non-linear way. Therefore, some innovative techniques such as Ant Colony Optimization, intended for solving these issues have been presented by several researchers. This paper presents an empirical study consisting of experiments related to the use of such algorithms applied to the routing problem, as a framework for understanding its behavior. Results shown in this paper indicate that some factors involved in the process can be fine-tuned in order to improve the performance of the algorithm.

## 1. Introduction

Some researchers have studied the behavior of social insects. Insect colonies show a high degree of efficiency in their activities and survival capabilities. It is believed that the cause of this efficiency resides in that while each individual in the colony considers its labor like if it where alone, the impact of its activity over the colony seems to be like a centralized coordination.

Stigmergy [1] is a kind of indirect communication, involving individuals of social insects colonies, in which some senses are not needed for the communication to take effect. This communication is very useful to coordinate and divide their efforts, and it is attributed to this the great success the colonies have in the achievement of their goals.

During foraging, ants deposit a pheromone trail in their way. This pheromone acts as a guiding agent to all other individuals, from the nest to the found food source without the need of the sense of sight. Pheromone trail remains on the ground for a limited period of time, so it is obviously more concentrated in short paths. As ants get to their destination and back more quickly than over longer paths, pheromone trail are deposited more frequently.

In addition, ants prefer to move through paths where pheromone trail are more concentrated. In consequence ants generally find the shortest path to get to the food source and back to the nest, although some other times they proceed probabilistically.

While this happens in the natural world, we would like to point to the problem of routing. As defined in [2] a routing algorithm specifies the route by telling each intermediate node on the route on which outgoing edge de message should be sent, depending on the destination.

In [3] Tel represents a network as a graph, where the nodes of the graph are nodes of the network, and there is an edge between two nodes if they are neighbors. The optimality of an algorithm depends on what is called the best path. A best path can be defined in terms of minimum hops, the shortest path or the minimum delay.

As the size of WANs grows, routing process may get very complex, and algorithms implemented on routing equipments may not show the best performance, therefore, distributed behavior of these systems is an important issue to address.

Many works has been presented in order to obtain the best path. They have been presented in order to solve different aspects, from distributed algorithm point of view. In this paper we propose a new approach.

Previous research [4] have shown that it is possible to model some complex problems, such as discrete optimization, quadratic assignment, traveler agent, among others, beginning from the inspiration of social insect behavior.

The aim of this paper is to contribute to the understanding on how ant algorithms work, and present relevant results of a set of empirical experimentations in a seek for optimal parameters, applied to the ANTNet algorithm developed by Dorigo, which intention is to address the problem of path determination optimization using ant algorithms.

We have found a convenient value rank for the variables of the system, and we believe they can be used in most of the network topologies.

The remainder of this paper is organized as follows. In section 2 we describe some related work. In section 3 we presents the model of our proposition. We explain our experiments and results in section 4.Section 5 is dedicated to our conclusions and the future work.

## 2. Related Work

Di Caro and Dorigo [5], have introduced an algorithm for adaptative routing based on the foraging behavior of ant colony. These algorithms explore data networks by the mean of routing table construction, and keep them adapted to network traffic conditions. This algorithm was named *ANTNet*.

Fundamentally, algorithm consists of the creation of artificial ants on each node of a network, represented by a graph with nodes and bidirectional links, at regular time intervals.

- Each node is characterized by the number of neighbors and a routing table, which contains probabilities of choosing neighbor *n* on the next hop, being *d* the destination node.
- Ants choose randomly a destination node and select next node among non-visited neighbor nodes. Probability of selecting a particular node is proportional to the value of the routing table for this node, as well as a local pool, which are created at each node by local traffic.
- Each visited node identifier is pushed into a node stack, which is a part of the ant's data structure. The time the ant has taken to get to the destination node is also stored.
- When an ant reaches destination node, it generates a *backward ant* and transfers the stack content to the new ant and then it dies.
- Backward ant takes the same route than the ant which generated the route, but in an opposite direction. As it goes back, the ant updates routing tables as shown ahead:
    - When the time the ant used to get to the destination node is less than the time memorized in the model, then the ant updates the routing table, in other words, it deposits more pheromone.
    - When the time it took the ant to get to the destination node is larger than the time memorized in the model, then the ant does not change the routing table.

Di Caro & Dorigo algorithm was compared to the following traditional routing algorithms:

- A simplified version of *Open Shortest Path First (OSPF),* the official routing algorithm for Internet.
- A sophisticated version of the distributed and asynchronous algorithm known as *Bellman Ford (BF),* with a dynamic cost metric.

- The algorithm *Shortest Path First (SPF)*, with a dynamic cost metric.
- The algorithm *Q-Routing* proposed by Boyan and Littman.
- The algorithm *Predicitve Q-Routing*, a *Q-Routing* algorithm extension.

The observable parameters in the tests were:

- The delay in delivery of packets
- The throughput, which means the total traffic transported.

With regard to throughput the variations between the different algorithms was minimal, while as much for delay, *ANTNET* showed a performance 4 times better than *Q-routing*, 3 times better than *Predictive Q-Routing*, 2.5 times better than *OSPF* and 1.5 times better than *BF and SPF*.

We believe that the performance of the algorithm could be optimized by appropriate manipulation of the parameters involved in its operation. Dynamic manipulation of these parameters, in function of the state of the system at each time step, is a part of our proposed model.

## 3. The Model

The proposed algorithm consists in the creation of artificial agents (ants and pheromone) which simulate the acting factors in the process of stigmergy. These agents are conformed by a collection of parameters that can be quantitatively varied in order to observe the global performance of the system, and try to find the optimal combination of factors to offer an approximation to the problem in the less time possible.

Determination of mentioned parameters is not a trivial task, since the global behavior of the system is an emergent property of the local interaction of the individual agents. In consequence, it can not be explained from the individual capabilities of these agents.

The model considers a simulated environment of Wide Area Networks (WANs), which are conformed by nodes in a graph and each one of them represents a Local Area Network (LAN). Links between nodes are represented by edges in the graph. See figure 1.

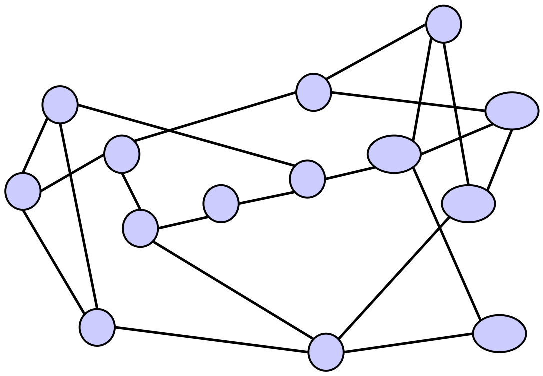


*Figure 1.WAN network conformed by LANs*

The algorithm consists in the creation of an ant for each network node with a random chosen destination node. This first generation of ants chooses next node in the search of the destination node using a uniform distribution of probabilities.

During the selection of the next node, it is necessary to verify that the node has not been visited yet by the same ant, in order to avoid infinite loops. To achieve this, the ant stores each visited node ID, which can produce two possibilities:

- In the case that selected node was previously visited, process of selection of the next node is repeated.
- In the case that selected node was not previously visited, the ant directs to it.

If the ant reaches a dead end, that is, every neighbor node have been visited, the ant dies and a new ant is created departing from this node with a destination node selected randomly, as shown in the figure 2.

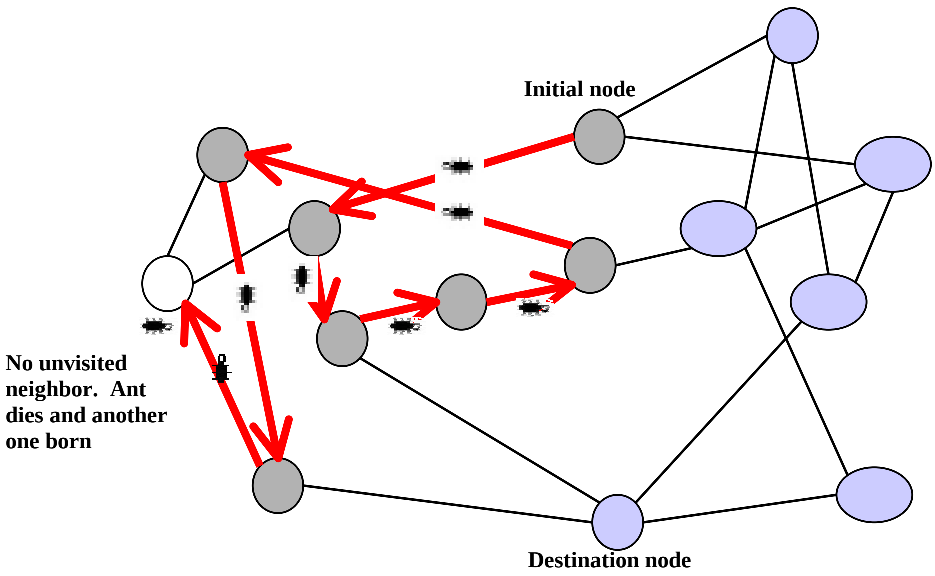



*Figure 2. Ant does not find a node to visit*

Every node has a routing table associated with it, which represents the pheromone concentration that exists at each node.

This table is conformed by a two dimensional array which values are defined by a probabilistic function $P_{nd.,}$ This function is the probability of choosing the link that takes the ant to the neighbor node *n* when destination node is node *d*. In other words, $P_{nd}$ is the degree of *goodness* of this route towards destination node *d*. Initially, this table has values that correspond to a uniform distribution of probabilities, as shown in figure 3.

The operation of the system is performed in discrete steps. At every step, all ants advance forward one node. The born of new generation of ants is made in a iterative way, after certain number of steps and the frequency of new births is a controllable parameter.

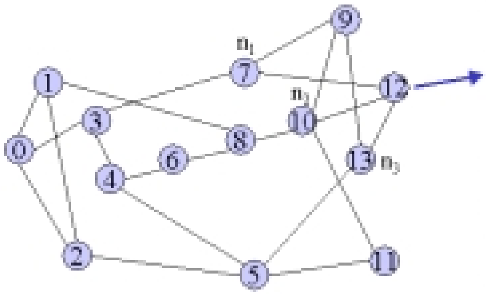
When the ant reaches its destination, it returns to the origin node, depositing a pheromone quantity at every node as it goes back. This action is made in function of the quantity of energy the ant still have in its body; if the ant had to walk more nodes, it has less energy and less pheromone to deposit.

When an ant gets to some particular node, it finds that the routing table have been modified by other ants. This way indirect communication between individuals appear, which is referred to as stigmergy.

In the last generation, we create as many ants as nodes exists in the system. It is intended to get from each node to all other nodes, in order to evaluate the work of earlier generation of ants.

We propose two different schemes for updating routing tables:

- The first one consists in depositing a fixed amount of pheromone on every node, proportionally to the number of nodes previously visited.




| | Origen | | |
|---|---|---|---|
| Destino | 7 | 10 | 13 |
| 0 | 0.33 | 0.33 | 0.33 |
| 1 | 0.33 | 0.33 | 0.33 |
| 2 | 0.33 | 0.33 | 0.33 |
| 3 | 0.33 | 0.33 | 0.33 |
| 4 | 0.33 | 0.33 | 0.33 |
| 5 | 0.33 | 0.33 | 0.33 |
| 6 | 0.33 | 0.33 | 0.33 |
| 7 | x | 0.33 | 0.33 |
| 8 | 0.33 | 0.33 | 0.33 |
| 9 | 0.33 | 0.33 | 0.33 |
| 10 | 0.33 | x | 0.33 |
| 11 | 0.33 | 0.33 | 0.33 |
| 12 | x | x | x |
| 13 | 0.33 | 0.33 | x |

*Figure 3. Initial routing table for node 12*

- In the second one, the ants consider the amount of elapsed steps in the global process, as well the number of visited nodes in order to determine the amount of pheromone deposited until that moment.

Previous work in artificial ants simulation, indicate that the parameters with more incidence on the global system behavior are the pheromone and the sensibility of ants to this agent [6]. This is the reason why we considered the second way of depositing pheromone.

In order to simulate sensibility of ants to pheromone, we looked for an existent variable in the system. The selected variable was the amount of live ants on each step, as a factor which dynamically modifies the quantity of pheromone in the system to evaluate the performance of the algorithm.

As an additional proposal, we generated two approaches to evaluate the system, which have impact directly on the way the last generation of ants choose the next node to visit.

- In the first one, the ant selects next node to visit probabilistically using a probability distribution in the tables.
- In the second one, the ant selects deterministically the link with the highest pheromone concentration, just the way it would be done by a data packet, which behavior should not be necessarily stochastic.

From the previous schemes, we have as a consequence three different implementations of the algorithm:

- Every generation of ants which have found destination node, except for the last one, deposit on their way back a fixed amount pheromone trail, which depends on the route length. The last generation of ants selects the larger value in the table to choose the next node. When an ant find the destination node it dies. We call this version of the algorithm DFP (Deterministic with Fixed Pheromone).
- Every generation of ants which have found destination node, except for the last one, deposit on their

way back a fixed amount pheromone trail, which depends on the route length. The last generation of ants proceeds probabilistically to choose the next node. When an ant find the destination node it dies. We call this version of the algorithm DPP (Probabilistic with Fixed Pheromone).

- Every generation of ants which have found destination node, except for the last one, deposit on their way back a variable amount of pheromone trail, which dynamically varies depending on the quantity of steps elapsed in the system, and consequently, the amount of live ants in the system, besides to the route length. The last generation of ants proceeds probabilistically to choose the next node. When an ant find the destination node it dies. We call this version of the algorithm PVP (Probabilistic with Variable Pheromone).

What is looked for, is that a multititude of ants acting over the system, eventually generate routes with the least possible number of hops, regardless of the source and destination nodes for a data packet.

## 4. The Experiments and Results

It is necessary to mention that the evaluating parameter selected for the tests, was *hop count*, or the number of nodes an ant visits until it gets to its destination node. The reason why we choose this metric is because most of the routing algorithms as the distance vector algorithms, uses this metric. Under this scheme, a route with less hops is considered better than other which requires the ant to travel over more nodes.

As mentioned before, one of this work intentions is to have the possibility of validate this model with representative instances of the problem. We used for the experimentation three different network topologies, known as *Simplenet (Figure 4)*, *NSFNet (Figure 5) and NTTNet (Figure 6)*, every one of them with bi-directional links.

The reason why we used this network instances, is because the first represent a typical complexity network, the second is a well balanced network (on number off nodes and links), and the third is unbalanced.

On the other hand, these topologies where also used by Marco Dorigo in his experiments, and we consider an important issue to find coincidences and deviations between our work and the work of Dr. Dorigo.

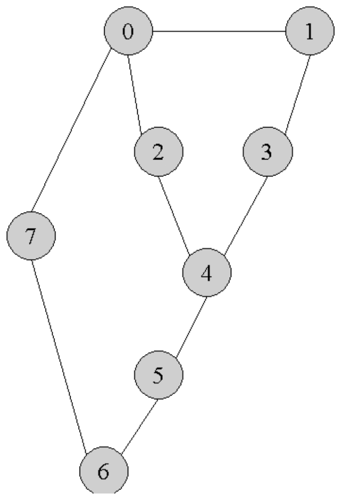



*Figure 4. Simplenet*

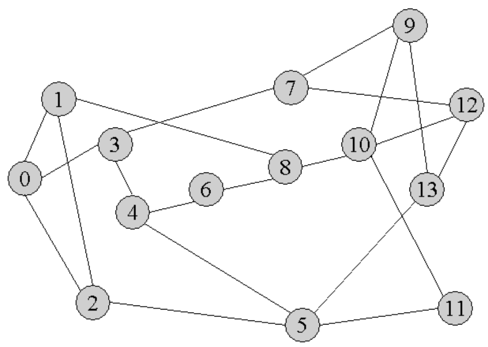



*Figure 5. NSFNet*

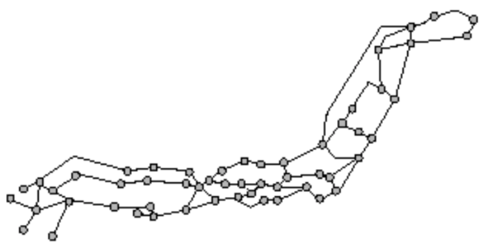


*Figure 6. NTTNet*

**System Parameters Configuration**

An important part of the experimental stage has been the selection of the parameters of the system. Some of them where manipulated in a systematic way, trying to find correlations between those parameters and the impact on the global performance of the algorithm, that is, performing a sensibility analysis.

The parameters to be manipulated where: The number of time steps of the system, the frequency of birth of new generations of ants, and the amount of pheromone of each ant.

Preliminary tests showed that the representative values for executing the different modalities of the program where those what are listed later on. These executions of the program were performed with every possible combination of values showed below:

- Number of time steps of the system: 2048, 4096, 8192, 16384, 32768, 65536
- Frequency of birth of new generations of ants: 64, 32, 16
- Amount of pheromone: 0.001, 0.005, 0.01, 0.05, 0.1

If we combine the different parameters and values, we have 90 different experiments; each one of them where executed 10 times, and we obtained the average results, searching for a significant and consistent set of results; because variations in results could arise, due to the stochastic nature of the model.

This scheme of experiments where taken for the three versions of the program, so there where 2,700 executions of it, only for *NSFNet*, and later some significant values where selected to run over *Simplenet* and *NTTNet* networks.

We thought this was a representative set of tests which lets us distinguish and analyze the incidence of the involved factors and the differences between the versions of the program.

## Analysis and Discussion of Results

In this section some representative examples of the program runs, and their results, which leaded us to establish the most important conclusions of our work.

Most of the tests where made over the *NSFNet,* since we consider it as a typical complexity network and it is well balanced. Another reason was to have the chance to manually calculate optimal results, and to be able to compare these with the system's output.

## Observable Parameters

As we consider a hop count metric, we thought that the most important parameters to measure are:

- Total sum of hops that every one of the ants used in the last generation. If the system works correctly, this value must be near 278 total hops, which is the optimal measure of the system. We call *Global* to this parameter in the future.
- Average of hops for each ant of last generation; this optimal value manually calculated is 3.15 hops per ant. This parameter is called *average* in the future.
- At last, we consider the amount of ants which fail in their search for the destination node; when routing tables have not converge in well defined values, ants roam without finding destination node, they find dead ends and in consequence they die. In the future we call this parameter *failures.*

## Analysis of the Incidence of Controllable Parameters

As exposed before, the manipulated parameters were:

- Quantity of steps the system acts. In the future *steps.*
- Frequency of births of new generations of ants, expressed in quantity of steps, that is, a frequency of 16 means a new generation born every 16 steps. We call it *frequency.*
- Quantity of pheromone. We call it *pheromone.*

Results show an important relationship between the three different controllable parameters. Some combinations of different values for each parameter, produce a system behavior which outputs values close to the optimum, although other combinations produced many failures or long routes.

An important detected observation is that, when it seems that varying one parameter has no effect over the global performance, suddenly it outputs results very close to optimal values.

In previous works with emergent behaviors [6], have been observed that *"order is generated on the edge of chaos"*. In other words, when it seems that the global system acts erratically, suddenly it stabilizes, which let us think there is a very thin line separating good from bad results, and some times, changes could be abrupt.

Below we analyze every one of the controllable parameters and its incidence over the global performance.

## Steps

Experiment results indicate that the quantity of steps the systems acts is a very important factor to the global performance of the algorithm.

In the models PFP and PVP, for similar values of *frequency* and *pheromone*, as we increment the number of steps, we observe how the quantity of *global* substantially descends as shown in figure 7.

.

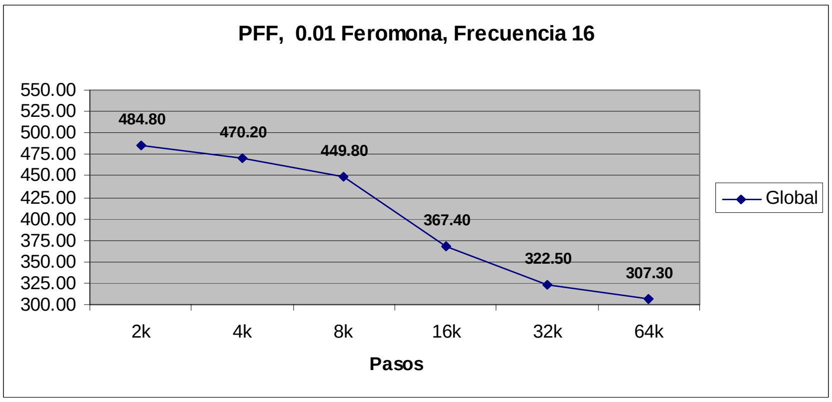



*Figure 7 PFP, 0.01 pheromone, frequency 16*

In the model DFP, for similar values of *frequency* and *pheromone* to the past experiment, we observed a *global* value very close to the optimal value (278), independently of the *steps* value managed as shown in figure 8. We conclude there is a more anticipated system convergence than in the model PFP

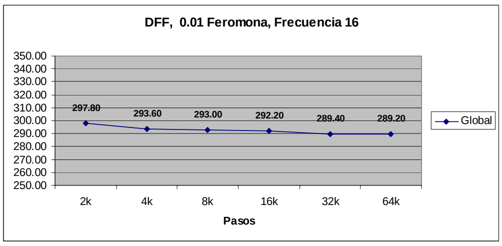



*Figure 8 DFP, 0.01 pheromone, frequency 16*

## Frequency

Increasing the frequency of birth of generations of ants, has a similar effect over the system than increasing the number of steps, but the time needed to get to the system convergence is significantly lower. At the end, the relationship between steps and frequency, have effect over the number of live ants in the system. We observed that a similar amount of *failures* and *average* for opposite values of *frequency* and *steps,* but similar values of *pheromone,* presented similar outputs. This was true for PFP and PVP.

For similar values of controllable parameters than the last example, but with DFP, values of *average* do not change considerably when incrementing *pheromone* and as we can see in figure 9, these get close to optimal values on most of the cases, no matter how much *pheromone* is considered in the experiment.

On the other side, quantity of *failures* in the system is practically equal to zero for any value of *pheromone.*

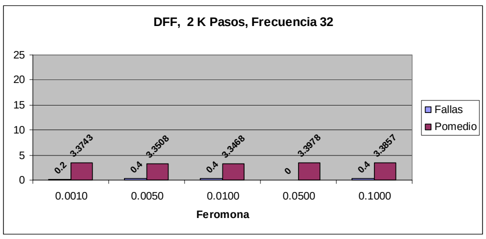



*Figure 9 DFP, 2K steps, frequency 32*

## Pheromone

The quantity of *pheromone* assigned to each ant plays an important role in the performance of the algorithm. If each ant have enough energy to deposit a more intense *pheromone* trail, then less ants would be necessary to conform good routing tables.

We observe in the experimentation that algorithms PFP and PVP, the amount of *pheromone* has an important impact over *failures* and *average* for any variation of other controllable parameters. This is shown in figure 10.

For DFP, we can see in general, that the amount of *pheromone* does not change the results. This algorithm seems to reach convergence since the initial stages of execution of the program. See figure 11.

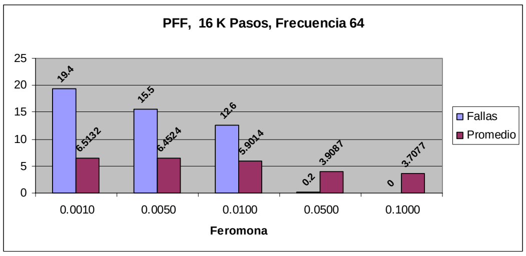



*Figure 10 PFP, 16K steps, frequency 64*

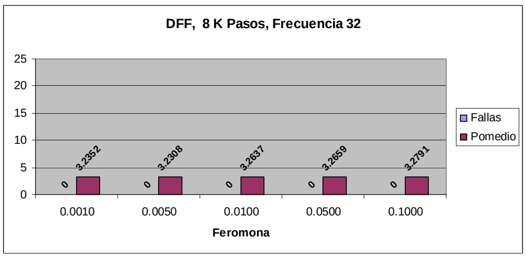



*Figure 11 DFP, 8K steps, frequency 32*

To illustrate the importance of the asseveration of Millonas (Millonas, 1994), *"Order is generated on the edge of chaos"* , it is presented figure 12, where we can see that using fixed values for *pheromone* and *frequency*, passing from 4K *steps* to 8K *steps*, the variable *global* descends noticeably.

In figure 13, we observe that with a fixed amount of *steps* and *frequency*, a little increment of *pheromone* is enough to take the value of *failures* from 4.7 to 0, while *average* goes down from 5.21 to 3.58.

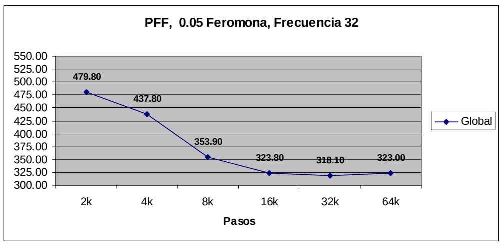



*Figure 12 Passing from 4K to 8K steps, global descends importantly*

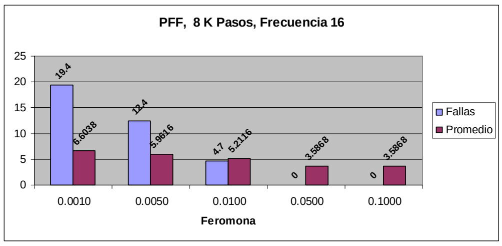



*Figure 13 Passing from 0.01 to 0.05 pheromone, failures and average descends importantly*

Combination of the best values for the controllable parameters, depend fundamentally in the size of the network. For a network with a few nodes, usually the values which can output good results are:

- Frequency 32 or 16
- 8K steps and greater
- 0.005 pheromone and greater

For a middle size network, recommended values are:

- Frequency 32 or 16
- 32K steps and greater
- 0.01 pheromone and greater

For a bigger size network, recommended values are:

- Frequency 16
- 64K steps and greater
- 0.05 pheromone and greater

It is important to say that once the results get near to optimal values, increasing values for the different variables, does not lead us to even better results. We can think that the system reaches stability in a point where the parameters have a convenient value, and any variation to this combination does not contribute to a better performance of the algorithm.

## Results on Simplenet and NTTNet

As mentioned earlier, most of the tests where performed over *NSFNet*. We made other experiments over *Simplenet* an *NTTNet* and the comments in the matter are the following:

### *Simplenet*

As this is a small network, it also is very simple in matter of complexity. The values for *average* and *global* manually calculated for this network are 2,92 and 82 respectively. In figure 14 is shown that no matter the quantity of *steps* it is possible to reach system's convergence with zero *failures*.

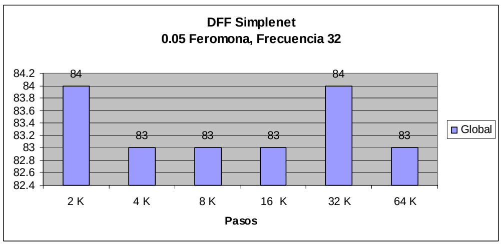



*Figure 14 DFP Simplenet, 0.05 pheromone, frequency 32*

### *NTTNet*

In the case of this network, we did not calculate manually the optimal values for *global* and *average*, because of the high complexity the problem presents. However, we believe it is important to include the results we got for its comment.

In general, we got values for *average* of 8.5 when there are no *failures*. We assume this value is good, when we are talking about a network with 55 nodes. Variable *global* finds a light negative variation as *steps* are incremented, what lead us to make an important conclusion: The quantity of *steps* the system requires, is larger as the network nodes number grows.

Execution of these programs could take approximately 3 hours, when the number of *steps* where 128K and *frequency* 32. Decreasing these parameters, descend proportionally the time of execution.

The size of the problem, would turn unmanageable the use of conventional methods, if we would try to find best routes in a non distributed way.

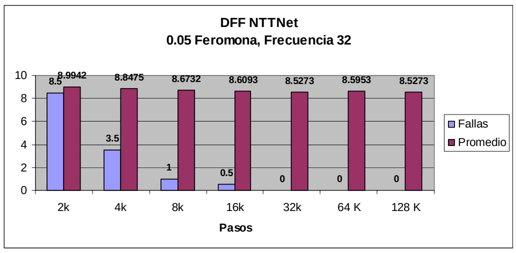



*Figure 15 DFP NTTNet, 0.05 pheromone, frequency 32*

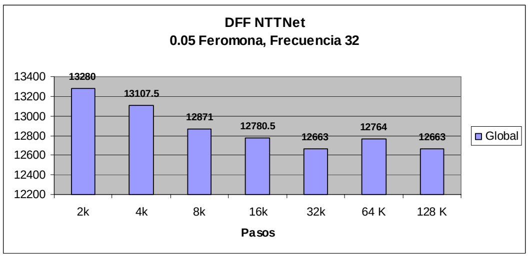



*Figure 16 DFP NTTNet, 0.05 pheromone, frequency 32*

## 5. Conclusions and Future Work

The use of algorithms inspired in social insect behavior, have reported promising results in the search for solutions for complex problems. Routing in networks with distributed ant algorithms, comprises a new and exciting source of research, because there are good reasons to affirm that knowledge on *swarm intelligence* [7], until now, have been acquired in a fragmented manner, and it is time to get all the pieces together.

We conclude that the simplicity of the implementation of the algorithm presented, plays an important role in the achievement of the objectives. Our work in comparison with the work of Dorigo, offers an advantage: we could remark that the implementation is more simple, but it captures the essence of the problem and detects the variables that participate in the global system behavior, without the need of complicated algorithm implementation.

As these are emergent technologies, it is important to say that the methodology used where totally empirical.

We have found a convenient value rank for the variables of the system, and we believe, they can be used in most of the network topologies that could arise, however, it would be necessary to make more tests to declare the last with reliability.

We confirmed with experimentation that *"The order is generated on the edge of chaos"*, as it is affirmed by Mark M. Millonas. When it seems that the global system acts erratically, suddenly it stabilizes, which let us think that there is a very thin line that divides good and bad results, and in certain occasions, changes may be very abrupt, and that is why correct selection of optimal parameters are even more important.

Although these distributed routing techniques could be seen as modern and promising, they have not been implemented yet in a real network environment, because some important issues have to be address before. Network models used until now have been only reality simplifications and they are required to be proven in more realistic conditions:

- Considering traffic control and congestion.
- Flow control.
- Quality of service
- Packet loss.
- Bandwidth requirements.
- Routing tables in function of costs or other metrics.

Results obtained in this work, could be used as a starting point for future work.

In networks with a great number of nodes, execution time would take a few hours, given the number of independent agents that could be active in every stage of the execution, and it is recommended to adapt the algorithm to work over several processors, in other words, implement parallelism.

It is recommended to use Linux as operating system for the execution of these programs. Acquired experience in this work, showed that it is much faster to execute the programs on this platform.

## 6. References


[1] Grassé, P., *"La Reconstruction du nid et les Coordinations Inter-Individuelles chez Bellicositermes Natalensis et Cubitermes sp. La Théorie de la Stimergie: Essai d´interprétation du Comportament des Termites Constructeurs"* Insect. Soc. 6 1959.

[2] Hagit Attiya, Jennifer Welch, *Distributed Computing*, Mc. Graw Hill, 1998

[3] Gerard Tel, *Introduction to Distributed Algorithms*, Chapter 4, Cambridge Edition, 2000

[4] Deneubourg, J., Aron, S., Gross, S., Pasteels, J.M.,*"The Self-Organizing Exploratory pattern of the Argentine Ant"* J. Insect Behavior. 1990.

[5] Dorigo, M., Di Caro, G., Gambardella, L., *Ant Algorithms for Discrete Optimization*. Artificial Life, The MIT Press. Volume 5, Number 5, 1999.

[6] Millonas, M., *Swarms, Phase Transitions, and Collective Intelligence.* Artificial Life III, The Santa Fe Institute Studies in the Sciences of Complexity. 1994.

[7] Dorigo, M., Bonabeau, E. Theraulaz, G., *Swarm Intelligence, From Natural to Artificial Systems.* The Santa Fe Institute Studies in the Sciences of Complexity. 1999.