2014 5th IEEE RAS & EMBS International Conference on
Biomedical Robotics and Biomechatronics (BioRob)
August 12-15, 2014. São Paulo, Brazil

Swarm Ant Algorithm Incorporation for Navigation of Resource Collecting Robots

C. L. Garzon, *Student Member, IEEE*, H. R. Chamorro, *Student Member, IEEE,* M. M. Diaz, *Student Member, IEEE*, E. Sequeira, *Student Member, IEEE,* and L. Leottau, *Student Member*

Abstract – **Swarm robotics requires the development of new strategies and algorithm integration, which allow for the improvement of the design and the applications for harvesting or collecting resources. This paper describes the programming and design of Finite State Machines (FSM) bio-inspired algorithms for seeker and resource gathering Pherobots systems, like Anthill Known Location (AKL) aggressiveness and sense of panic. FSM designing allows for the use of control architectures for behaviour-based agents and for measuring the change in system performance. Simulations demonstrate the capability of the algorithms under different environments and scenarios.**

Keywords – **Pherobots Based-behaviour Architecture, Bio-inspired Algorithms, Multi-Agent Systems, Swarm Intelligence.**

I. Introduction

Distributed systems is a branch of science that has aroused the interest of researchers, and is currently booming. Distributed artificial intelligence, Multi-Agent Systems (MAS), distributed robotics, are topics included in this cluster. Swarm robotics is an applied field related to these topics, where modelled biological paradigms are transferred to artificial swarms in order to understand the means of solving problems that real swarms have to face in their environment.

Pherobots are a particular case of swarmbots based on species colonies that communicate and organise through chemical messages. Through emergent behaviour the overall result is known giving another view of portraying the set of agents as an autonomous system that meets specific tasks. Local interactions of information make swarm behaviour emerge [1]. The bio-inspired algorithms are improved continuously to understand this behaviour by means of surveys and tests in order to resemble those found in nature.

The swarm behaviour of ants is the basis of defining how the swarm robot system will operate [2]. Ants communicate among themselves using pheromones; pheromones are classified into primer pheromones which are related to the queen's substance, and releaser pheromones which are associated to sexual pheromones, alarm pheromones, trail pheromones and aggregation pheromones [3]. Using a pheromones trail ants attract other ants to food, when they are in the process of food gathering.

Cesar L. Garzon is an Engineer at Automatizacion Avanzada, Bogotá, Colombia (e-mail: clgarzons@gmail.com)
Harold R. Chamorro is a PhD Student at KTH, Stockholm, Sweden (e-mail: hr.chamo@ieee.org)
Margarita M. Diaz is Leader Project in NDT Innovations Inc , (e-mail: margarita.diaz@ieee.org)
Emmanuel Sequeira is a Master of Statistics Student at UTEP, Texas.
Leonardo Leottau is a PhD student at Advanced Mining Technology Center, Department of Electrical Engineering, Universidad de Chile, (e-mail: dleottau@ing.uchile.cl)

The properties of the pheromone trail are adjusted according to changes in the environment. Virtual Pheromone is related to the connection of the agents through the software [4]. Implementing the swarm behavioural mechanism to the robotics is feasible; the proposed method is the adjustment of the trail duration based on the food quantity. In the real environment the food volume is limited, so when the food is finished, the pheromone trail attracting the agents is interrupted in order to refocus the search to another place [5]. Considering the simulated and experimental results obtained, the mentioned method is effective.

On the other hand, the methods of evaporation and diffusion are demonstrated through the increase of the pheromone field and strengthen the information when the nest is found. However, the swarm is a distributed autonomous system, and the results of simulations and experiments have variance due to external condition changes like humidity and temperature. Colonies of social insects are constructed by enormous numbers of the individuals, so in order to decrease the variance the number of agents must be increased [6].

Data acquisition systems ensure that the whole system will be operational, and an important characteristic, which simplifies future implementations, is modularity. The embedded system proposed in [7] was successfully integrated into a low-cost commercial mobile robot for creating a modular robot platform with advanced sensor capabilities.

Some of the advantages of MAS over single agent systems is the failure to be tolerant, because the task could be completed by other agents even if some of them fail; the scalability of the number of agents operating increase without affecting the system design; the simplicity of design is not a performance obstacle during the task, as it does not require advanced programming. This means a reduction of costs and construction time; some tasks require a group solution, so a collaborative MAS is suitable for developing the task; greater efficiency let us solve a task using less time.

An alternative to approach a better system implementation is the design in [8]; an experimental device to study bio-inspired self-organization paradigms as stigmergy. An interactive surface is proposed which provides an active environment to implement and display dynamic information which allows the robots to interact with each other. The challenge of the swarm ant algorithms resides in defining cooperation rules that would help to the global function of the system [9].

According to the above, this paper proposes to measure the performance improvement of an emergent behaviour such as seeking, and gathering resources by a pherobots group. Three algorithms for simulation are used to determine results.

The tool Pyro (Python Robotics) is used to control a set of robots based on swarm intelligence programming [10]. In practice, Pyro normally operates in addition to a simulator such as Player/Stage. The main objective is to enable programming swarm robots to search for target objects in maze. As a result, the swarm robots are capable to move the target objects in a maze to the home location, the starting point. The success of the implementation is that using the control logic (the brain) of swarm robots, a swarm intelligence for solving a maze problem is implemented, moving a set of target objects in a maze.

The Player/Stage is an open source/free software for multirobot simulation that enable the testing of swarm systems and will be the tool used in this trial [11].

This paper is structured as follows; Section II briefly describes the characteristics of the algorithms implemented. Section III contains the description of the FSM and how it works. Section IV shows the algorithms implemented on the FSM. Section V presents the tests and the results based on implementation. Section VI explains the results of the simulation using Pyro. Section VII is reserved for conclusions and future possible research work.

II. Bioinspired Algorithms Applied

The main objective of this paper is to use three algorithms based on animal behaviours on swarm pherobots. Each algorithm has specific functions and a specific goal; however these should not be in opposition to the overall aim.

A. Anthill Known Location Algorithm (AKL)

In [12], is defined tha AKL Algorithm main feature is the Anthill Known Location, hence the name. AKL implies a deep knowledge of agents that facilitate the return of ants to the origin place. This knowledge is centred on the observation of ants' colonies, which have complex organs and systems that permit them to count their steps with the purpose of estimating the direction of where the anthill could be, or to guide them in determining the sun's position. Therefore, erratic and random movements when seeking food results taking them more time, than when they are simply returning to the anthill.

Also in [12], is studied the dependence between simulated swarms that use AKL and others that use Sectorial Division of Labour (SDL). The latter one attempts to eliminate the knowledge and replace the orientation system by using other sensors. It is observed that when using the AKL method convergence exists, which in this case achieves 100% convergence. The convergence means that the agents are able to build and strengthen a pheromone's path between the anthill and the food area. That study concluded that an orientation system allows for a correct performance of the swarm while the SDL algorithm depends on the environment and on experimental conditions.

the effects and consequences of each one in the final re

B. Feeling of Panic and Self-preservation Algorithm

Panic is an individual, inherent signal in animals; pherobots artificially spread panic to affect the behaviour of each agent with the intention of moves from one activity to another while does not compromising their integrity and avoiding exposing them to damage. The term "feeling" refers to the way used by an agent to calculate and decide that it might be at risk, either by external variables such as a predator or inner variables such as low energy level.

This technique allows for keeping most of the agents in operation and contributing to the task. Panic is defined as a consequence of an event where an agent reaches the end of the map without having found food, so the agent is lost [12]. Reference [13], has another point of view, it defines panic as an state of the agent caused by an obstacle that has not been avoided, resulting in the agent getting stuck, so it sends out a panic signal to move backward to try and get out of the state and continue navigating.

Using this type of conservation strategy is essential for the proper performance of MAS, however its impact has not been measured on overall performance which may be of future research interest.

C. Spatial Interference Reduction

The aim of this strategy is to prevent collisions between agents that attempt to occupy the same space due to competition among homogenous agents that can cause stagnation. In order to solve this difficulty, in [13] is proposed a bio-inspired technique based on the aggressive behaviour among animals competing for a resource; this is a passive fight that uses a non-contact ritual dance and is denominated rational aggression.

There are three simple ways to stimulate aggressive behaviour of agents, apart from rational aggression, which depends on the way of estimating the level of aggressiveness: random, pre-established hierarchy and the amount of free space behind them. The greater the area the higher the level of aggressiveness.

D. Gripper Algorithm

The gripper algorithm simulates a two arm claw with sensors, which are activated when objects are inside. This element is suitable for the collecting objects task. The specifications of the prototype must define the arm claw size and the initial state (open/close). The use of Gripper is inspired by the robotic device Pioneer Gripper, used on the research robot Pioneer. The Gripper interaction with the simulation environment is adequate for the gathering objects task in the map. In addition, the Gripper could be programmed using bio-inspired algorithms in order to increase the efficiency during the task. The performance of sensors' prototype is shown on Fig. 1.

Fig. 2, Fig. 3 and Fig. 4 show how a map with the AKL and Spatial Interference Reduction algorithms applied looks and the path followed by an agent that has identified the resource.

III. State Machines Based on Behaviours

The state machines enable the development of a method to include new algorithms in the agent's programming, and thus measure the impact of each one on the overall system performance.

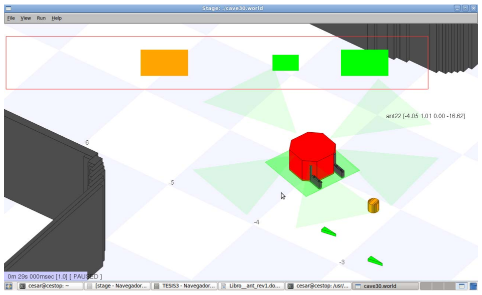
Fig. 1. The "Ant.inc" prototype and some of its sensors operating.

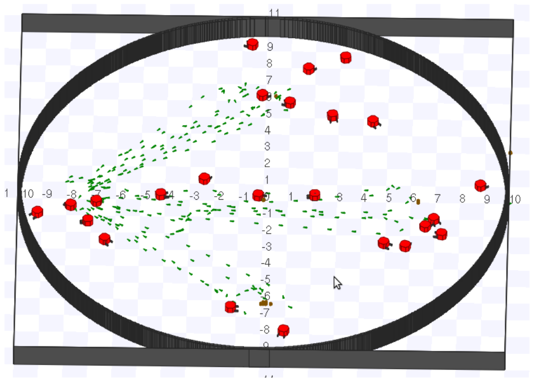
Fig. 2. RING map simulation shows the path followed and the trail strengthen.

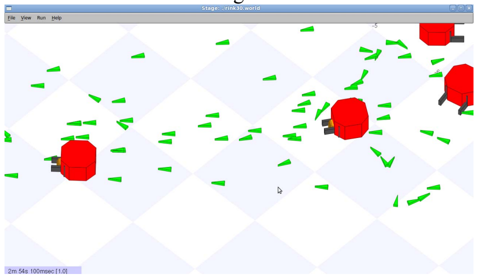
Fig. 3, Some agents are spreading and following the pheromones path to find the food resource

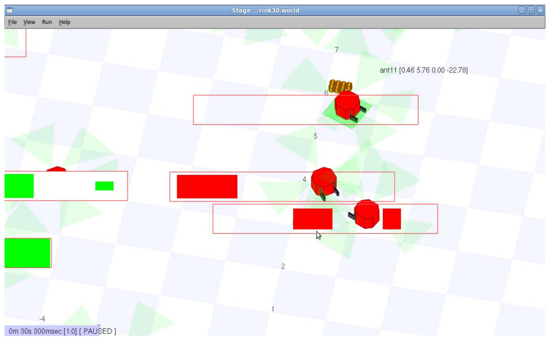
Fig. 4, Two agents figthing for space, using Spatial Interference Reduction Algorithm.

A control architecture based on behaviours is used; it is compound, with layers that correspond to a single behaviour or state, a concept is introduced in [14]. The behaviours are not linked, but sorted by relevance; advanced behaviours are at the top while survival behaviours are at bottom, some behaviour may inhibit another based on external changes of the environment. One of the most outstanding benefits of behavioural control is that systems are capable of overcoming failures, if any of the behaviour from above has failed, the survival behaviours can keep the robot operating.

Through the implementation of FSM, the agents are simulated with a behavioural architecture. One of the main features of using FSM is that the change of logic signals through the states can be represented easily using graphics, while other tools like truth tables need to show all the combinations that stimulate the change [14]. Thereby, it is just required to consider the output signal and the possible future state. Additionally, each state has some actions associated that activate specific signals. For this reason all the states' change are known and defined by transition paths. Thus, a present state will only change to a known future state. The main advantage in using FSM is the high independence of the states and the low linkage between them. This implies a direct association of the control architecture based on behaviour with the states of a FSM.

The state machines have transitions defined that are activated by a logic function. For implementing state machines using code, it is enough to use control words from a programming language like C++, for example *if, while, for or switch.*

Fig. 5 shows a FSM of a basic gathering agent capable to explore, gather, go back and avoid obstacles. The initial state of this machine is the behaviour "navigate". When the agent navigates, it can find an obstacle or an object to collect. In the case that the agent encounters an obstacle, he avoids it and continues navigating. If the object is to be gathered, the agent approaches, takes it, and then carries it back to the starting point. If the agent finds an obstacle when he is going back, he has to avoid it and continue on his way. After the agent arrives at the anthill, he goes out and starts to explore again to continue the cycle.

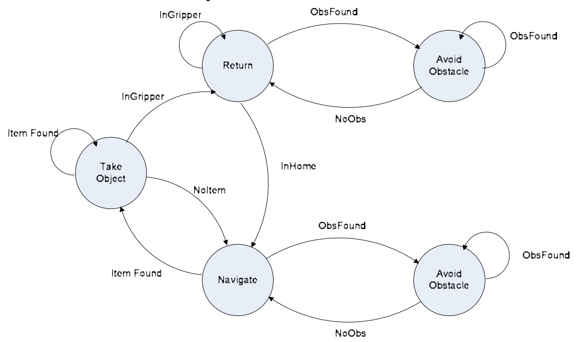

Fig. 5 State Machine for a Collector Agent

The logic signals used in both basic FSM as improved versions are consequences of the application of bio-inspired algorithms are explained in Table I.

When the agent is performing an established behaviour and any of the signals described above turn on then a state change is produced and the pherobot changes its behaviour.

In this study case, the agent actuators are the wheels to move and the claw to take the objects. The behaviours can be defined like all the actions that agent does through those actuators.

TABLE I. LOCAL SIGNALS FOR STATE TRANSITIONS

NAME	DESCRIPTION
OBSFOUND	TURNS ON WHEN THE DISTANCE SENSORS DETECT OBJECTS WITHIN ITS OPERATING RANGE
NOOBS	TURNS ON WHEN DISTANCE SENSORS DO NOT DETECT OBJECTS WITHIN ITS OPERATING RANGE. IN THE CODE THIS SIGNAL AND OTHERS ARE NEGATED FUNCTIONS, I.E. NOOBS = NOT (OBSFOUND)
ITEMFOUND	TURNS ON WHEN IN FRONT OF THE CAMERA OF BLOBFINDER SENSOR THERE IS AN ORANGE OBJECT, REGARDLESS OTHER COLOURS PRESENT.
NOITEM	NOT(ITEMFOUND)
INGRIPPER	TURNS ON WHEN THERE IS AN OBJECT BETWEEN THE HANDLES OF THE CLAW.
INHOME	TURNS ON WHEN THE AGENT IS LESS THAN ONE METER OF THE START POINT.
PHEROMFOUND	TURNS ON WHEN BLOBFINDER SEES A GREEN OBJECT AND NO PRESENCE OF ORANGE OBJECTS.
NOPHEROM	NOT(PHEROMFOUND)
ROBOTFOUND	BLOBFINDER IS IN FRONT OF A RED OBJECT AND THIS OBJECT IS IN THE RANGE OF FORNTAL RANGERS, THERE IS NO PRESENCE OF ORANGE OBJECTS.
NOROBOT	NOT(ROBOTFOUND)
PANIC	SOME TIME WITHOUT FINDING GREEN OR ORANGE OBJECTS.

However, there is a particular case related to "avoid obstacles" because the agent reacts automatically applying an arithmetic function instead of an algorithm. This is similar to reactive control architectures, so the response of the system is faster. Another special case is when an object is detected near the origin; since it would be an already gathered object the agent must reject it.

IV. SWARM ANTS ALGORITHMS INCORPORATION

A. *AKL Algorithm Incorporating*

The AKL algorithm must include a modification in the behaviour code "return" because it is expected that the agent leave a trail of pheromone when it has collected an object; this implies that it is possible to automatically generate a trace between the place where the resources are and the anthill, and thus any agent can follow the trace and save on search time.

The second part of the algorithm is "follow trace" in order to take advantage of the path marked by other agent. This behaviour is not taken into account in the initial state machine because follow trace was not defined as a behaviour; now using a behavioural architecture it has to be defined and coupled in the FSM.

Accordingly, the direction used is opposite to the AKL algorithm direction, so the agent decides to take the path that move away from the start point. The result of using this technique is as expected; due to fact that the resources are located in the far places of the map. These results can be matched in the FSM of Fig. 6.

B. *Incorporating Aggressiveness*

The behaviour "fight" introduces the aggressiveness to the algorithm; this behaviour is focused on creating a hostile and defensive agent, when another agent is blocking his way, i.e. an agent is returning to the anthill carrying the resource, where it meets with another agent that is following the pheromones way. So, this behaviour is aimed on reducing the spatial interference between agents, based on animal behaviour. In the Pherobot systems, the agents are competing for the space in front of them, when the winner takes the space and the fight is finished. When a Pherobot detects another Pherobot in front of frontal sensors, a random work variable is calculated between 0 and 10, where 0 is the aggressiveness level and 10 is the fear threshold. This level means the distance that the Pherobot will advance or move back. The "fight" is implemented using random values so the agent calculates a random value of aggressiveness and using a math relation the fear threshold is calculated. Aggressiveness Level and Fear Threshold are inversely proportional.

Having defined this behaviour, it is included in the code, adding the logical control signals, Fig.6.

C. *Panic Incorporation*

Panic is a Pherobot's logic signal which activates by internal counters and not by sensors. When a limit, which was defined by the programmer, is overcome the Pherobot leaves their search and returns to the starting point; the counter avoids the event that the Pherobot can be lost after a non-successful search time, because he returns to the initial point to start the search again. These design details allow for a real Pherobot improving and increasing its chances to be successful and also avoids the risk of getting far and without enough energy to return to the initial point. If this happens to various robots, the system would be reduced in a significant amount.

The difference between the states are: the state "return" ends when the Pherobot arrives at the starting point, while the state "Panic" could end when it founds in its return path a pheromone's object.

This state is activated by the panic signal turning on or because the pherobot has an object in its claw.

Finally, a FSM is defined as a result of a design development where are included the three bio-inspired behaviours. This version is named Type 3 and it is showed in Fig. 6.

V. SCENARIOS

With the FSM codes for the agents is pretended to measure the performance of searching and gathering for a large population. Three different maps are used, low, middle and high level difficulty.

Including different scenarios are necessary, with the purpose of involving the features of multi-agent Pherobots systems because the environment around them is unknown, and they depend on the environmental information to do their task. Additionally, modifying the difficulty levels of the maps allows testing the bio-inspired behaviours for such purposes. Fig. 7 shows the maps for the tests.

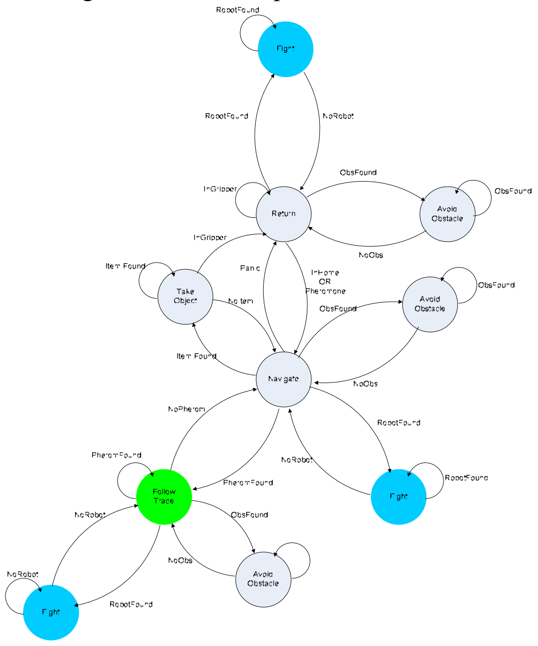

Fig. 6 State Machine with BioInspired Algorithms

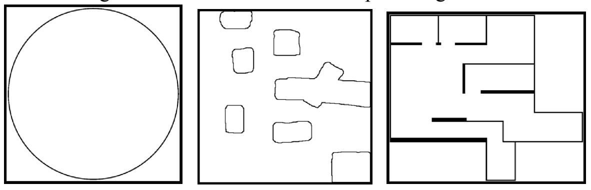
a)Ring b) Cave c) Autolab

Fig. 7 Experiment Maps

Each map has to be accomplished with features defined that describe the simulation variables. In Table II the features of the test are shown.

For data collection, the main feature measured is the time used by each system to complete the task of searching and gathering the objects repeatedly. The average is then calculated. As verification tool for results were used standard deviation and standard error.

Fig. 9 shows the result of the experimentation for each testing environment, like average time for each type of system in connection with the difficulty increase. When difficulty increases, the systems take more time to get successful.

Apart from simulated tests to measure averages, the data dispersion effect and the error of measurement are also observed.

Data dispersion is calculated from the standard deviation, which is what measures the dispersion power of a statistical function, like those used in this study case. The error that is obtained is called the standard error and is calculated from the standard deviation and the number of samples. The acceptance criterion is 10%, and in this case it is not overcome.

Due to the pheromones' effect of creating direct paths between the resource place and the storage place, the percentage of time for solving the problem described in Table IV has been improved.

TABLE II. DESCRIPTIVE SHEET OF VARIABLES SIMULATION

MAP	RING	CAVE	AUTOLAB
AREA	314 M2	284 M2	
DIFFICULTY	WITHOUT OBSTACLES	FEW OBSTACLES	OBSTACLES AND NARROW PATHS
AMOUNT OF RESOURCES		30	
AMOUNT OF RESOURCES PER GROUP		10	
EVAPORATION TIME		24 S	
AMOUNT OF AGENTS		28	
FSM TYPE		BASIC, TYPE 1, TYPE 2, TYPE 3	
TIMEOUT		1000 SEC	
AMOUNT OF TESTS PER EACH TYPE	16	15	15
TOTAL TESTS	64	60	60

TABLE III. DECREASE PERCENTAGE ON AVERAGE TIME FOR PHEROBOTS SYSTEMS ON DIFFERENT MAPS COMPARED TO THE BASIC AGENTS SYSTEM

MAP	TYPE I	TYPE II	TYPE III
RING	19%	15%	18%
CAVE	35%	32%	33%
AUTOLAB	43%	34%	40%

Another variable highly influential in the success of this simulation was the evaporation time, which according to the programming criterion and based on observations from each one of the maps, is 24 sec.

TABLE IV. STANDARD ERROR PERCENTAGE ON THE DIFFERENT MAPS

MAP	BASIC	TYPE I	TYPE II	TYPE III
RING	5%	3%	3%	3%
CAVE	5%	3%	5%	3%
AUTOLAB	9%	7%	6%	5%

VI. SIMULATION RESULTS

Type 3 Pherobots are used to measure the impact of the number of agents on performance. During the test, the quantity of agents are changed. As a result, is found that the simulated system has simulation deficiencies and contains mistakes as well as conflicts in performance, when more than 28 agents are used. However, a lower limit was defined according to the emergent behaviour of the agents. The lower limit is 8 agents. During the tests samples using 8, 12, 16, 20, 24, and 28 agents are taken, each group of agents are tested 10 times and it is applied on the three maps.

According to the complexity of the system, the number of agents required is higher. The survey showed that the task is not successful, or is partially fulfilled when 8 agents are used, so it is necessary to establish 1000 seconds as a time limit. In contrast, the groups with higher population of agents completed 100% of the task despite the time it took.

In Fig. 8 performance curves according to the number of agents are shown, whose analysis allows for a criterion of the number of robots for experimentation.

While analyzing the figure, it is noticeable that the effectiveness of the system decreases if the agent's population is lower. The performance of the system in the range of 20 up to 28 agents could be considered as stuck, due to the increase of agents population not increasing the effectiveness of the system significantly; the CAVE curve shows that the time increases in spite of the population of the agents also increasing, so the higher population of the agents does not mean more efficiency.

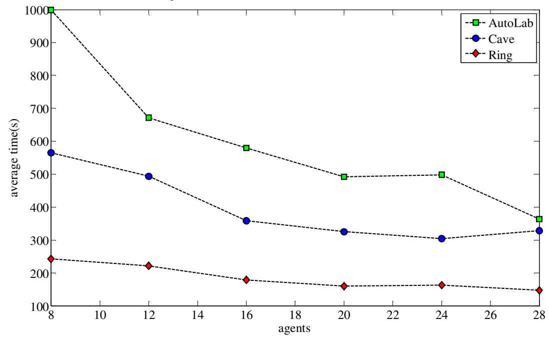

Fig. 8 Average Time Obtained for Each Type of Pherobot System

According to the performance results, the similar behaviour for each type of system, and the maximum population of agents for a stable system, it has been decided that the test should use 28 pherobots.

The bars in Fig. 4 show a large decrease in time needed for fulfilling the task of searching and gathering for the systems based on state machines Type 1, Type 2 and Type 3. This behaviour is a tendency that repeats in all simulated maps, ensuring that using bio-inspired algorithms increase the performance of the systems.

The average time regarding the use by a system based on AKL algorithms, aggressiveness and panic feeling to a system based on basic algorithms reflects percentage decreases. This information is shown in Table III.

The systems Type 1, Type 2 and Type 3 have a better performance for complex maps and higher difficulty levels. This may be due to the bio-inspired mechanisms being more useful in hard conditions because they get feedback from the environment, while the success of basic agents is achieved randomly.

Apart from simulated tests to measure the average, the data dispersion effect and the error of measurement are also observed.

VII. Conclusions

The bio-inspired algorithms improved the pherobots systems efficiency simulating their searching and gathering task, achieving a decrease of up to 18% in RING map, up to 35% in CAVE map and up to 43% in AUTOLAB map.

The Anthill Known Localization AKL algorithm and the Pheromones Follow Path are the ones that have more impact on the system efficiency.

The implementation of aggressiveness on the simulated pherobots system does not increase the efficiency of the system directly, however, it is a key development for the system and its future implementation, because it allows for locating and reacting to the presence of another pherobot, avoiding jams, and decreasing the quantity of inactive ones. In a real implementation it would help to fulfil the entire task.

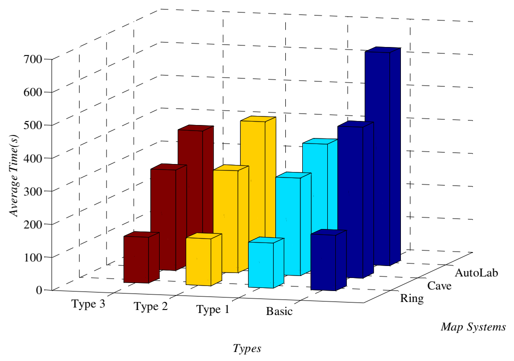

Fig. 9 Average Time Obtained for Each Type of Pherobot System

The feelings of panic and self-preservation are two important features of bio-inspired behaviour, even though they does not pose a big impact on decreasing the gathering time, and do not affect the performance. However, they are values added to the system that reduce extra cost as a result of lost agents.

References

[1] E. O. Wilson. Sociobiology. Sinsisakusya, 1999. in Japanese

[2] R. Fujisawa, Y. Shimizu, F. Matsuno, "Effectiveness of Tuning of Pheromone Trail Lifetime in Attraction of Robot Swarm", System Integration (SII), 2011 IEEE/SICE International Symposium on, On page(s): 702 – 707

[3] M. Matsuka, H. Kitano, T. Matsumoto, M. Oono, and N. Gokan. Biology of Insect. Tamagawa university publishing, 1992. in Japanese

[4] L. Szumel, J. D. Owens "The Virtual Pheromone Communication Primitive", in Distributed Computing in Sensor Systems, Vol 4026/2006, pp: 135-149

[5] H. Hashimoto, S. Aso, S. Yokota, A. Sasaki, Y. Ohyama,; Kobayashi, H., "Stability of Swarm robot based on Local forces of Local swarms," SICE Annual Conference, 2008 , vol., no., pp.1254,1257, 20-22 Aug. 2008

[6] R. Fujisawa, H. Imamura, T. Hashimoto, F. Matsuno "Communication using pheromone field for multiple robots", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, IROS 2008. On page(s): 1391 - 1396

[7] D.M. Martinez, J. Haverinen, J. Roning, "Sensor and connectivity board (SCB) for mobile robots", Electronics Conference, 2008. BEC 2008. 11th International Biennial Baltic, On page(s): 175 – 178

[8] O. Simonin, T. Huraux, F. Charpillet, "Interactive Surface for Bio-inspired Robotics, Re-examining Foraging Models", *23rd IEEE International Conference on Tools with Artificial Intelligence* (ICTAI), 2011, On page(s): 361 – 368

[9] G. Picard, "Cooperative Agent Model Instantiation to Collective Robotics", Engineering Societies in the Agents World V, ESAW 2004, On Page(s): 209 – 221.

[10] L. Byung-Jun; J. Jeong-Hoon, W.Gyun "A Case Study on Programming Intelligent Swarm Robots Using Pyro Environment and Player/Stage Simulator", *International Conference on Convergence and Hybrid Information Technology*, ICHIT '08. On page(s): 3 – 6

[11] Richard Vaughan. "Massively Multiple Robot Simulations in Stage", Swarm Intelligence 2(2-4):189-208, 2008. Springer.

[12] Álvaro Gutiérrez Martín, Felix Monasterio-Huelin Maciá. Algoritmos de rastreo inspirados en colonias de hormigas. Universidad Politécnica de Madrid (ETSIT).

[13] P.J. Yinan (Carl) Zang b. Spatial interference reduction for multi-robot systems using rational and team-based aggression. Sc., Ohio State University, 2003.

[14] R. A. Brooks. A robust layered control system for a mobile robot. IEEE Journal of Robotics and Automation, 2(1):14- 23, 1986.