Swarm Intelligence and its Applications

Shebin S
Dept. of Computer Science & Engineering
S.J.B Institute of Technology
Bangalore, India
shebins298@gmail.com

Mallikarjunaswamy S
Dept. of Computer Science & Engineering,
S.J.B Institute of Technology,
Bangalore, India
Pruthvi.malli@gmail.com

***Abstract*—Swarm intelligence is the collective behaviour of decentralized, self-organized systems, natural or artificial. The concept is employed in work on artificial intelligence. The expression was introduced by Gerardo Beni and Jing Wang in 1989, in the context of cellular robotic systems.**

Swarm Intelligence systems are typically made up of a population of simple agents or boids interacting locally with one another and with their environment. The inspiration often comes from nature, especially biological systems. The agents follow very simple rules, and although there is no centralized control structure dictating how individual agents should behave, local, and to a certain degree random, interactions between such agents lead to the emergence of "intelligent" global behaviour, unknown to the individual agents. Natural examples of Swarms include ant colonies, bird flocking, animal herding, bacterial growth, and fish schooling. The definition of swarm intelligence is still not quite clear. In principle, it should be a multi-agent system that has self-organized behaviour that shows some intelligent behaviour. In this paper, the various application of Swarm Intelligence is analysed. It includes mainly Bio Tracking, Ant Colony Optimization.

***Index Terms*—Swarm Intelligence, artificial intelligence, bio tracking, ant colony optimization.**

I. Introduction

We are all familiar with swarms in nature. The word swarm conjures up images of large groups of small insects in which each member performs a simple role, but the action produces complex behaviour as a whole. The emergence of such complex behaviour extends beyond swarms. Similar complex social structures also occur in higher-order animals and insects that don't swarm: colonies of ants, flocks of birds, or packs of wolves. These groups behave like swarms in many ways. Wolves, for example, accept the alpha male and female as leaders that communicate with the pack via body language and facial expressions. The alpha male marks his pack's territory and excludes wolves that are not members. Several areas of computer science have adopted the idea that swarms can solve complex problems. For our purposes, the term swarm refers to a large group of simple components working together to achieve a goal and produce significant results. Swarms may operate on or under the Earth's surface, under water, or on other planets.

Swarms consist of many simple entities that have local interactions, including interacting with the environment. The emergence of complex, or macroscopic, behaviours and the ability to achieve significant results as a team result from combining simple, or microscopic, behaviours. Intelligent swarm technology is based on aggregates of individual swarm members that also exhibit independent intelligence. Members of the intelligent swarm can be heterogeneous or homogeneous. Due to their differing environments, members can become a heterogeneous swarm as they learn different tasks and develop different goals, even if they begin as homogeneous. Intelligent swarms can also comprise heterogeneous elements from the outset, reflecting different capabilities as well as a possible social structure. Researchers have used agent swarms as a computer modelling technique and as a tool to study complex systems. Simulation examples include bird swarms and business, economics, and ecological systems. In swarm simulations, each agent tries to maximize its given parameters. In terms of bird swarms, each bird tries to find another to fly with, and then flies slightly higher to one side to reduce drag, with the birds eventually forming a flock. Other types of swarm simulations exhibit unlikely emergent behaviours, which are sums of simple individual behaviours that form complex and often unexpected behaviours when aggregated.

Swarm intelligence techniques are population-based stochastic methods used in combinatorial optimization problems in which the collective behaviour of relatively simple individuals arises from their local interactions with their environment to produce functional global patterns. Swarm intelligence represents a metaheuristic approach to solving a variety of problems. Swarm robotics refers to the application of swarm intelligence techniques to the analysis of activities in which the agents are physical robotic devices that can effect changes in their environments based on intelligent decision-making from various input. The robots can walk, move on wheels, or operate under water or on other planets. Practitioners in fields such as telephone switching, network routing, data categorizing, and shortest-path optimizations are investigating swarm behaviour for potential use in applications.

II. BASIC CONCEPT

A. Bio tracking

To expedite the understanding of how large-scale robust behaviour emerges from the simple behavior of individuals, the project videotaped ants' behavior over time, using a computer vision system to analyze data on the insects' sequential movements to encode the location of food supplies. The intention was to use ants' behaviour models to improve simple robot teams capable of complex operations.

The objective here is to show how robotics research in general and multi-robot systems research in particular can accelerate the rate and quality of research in the behavior of social animals. Many of the intellectual problems we face in multi-robot systems research are mirrored in social animal research. And many of the solutions we have devised can be applied directly to the problems encountered in social animal behavior research.

One of the key factors limiting progress in all forms of animal behavior research is the rate at which data can be gathered. As one example, Deborah Gordon reports in her book that two observers are required to track and log the activities of one ant: One person observes and calls out what the ant is doing, while the other logs the data in a notebook . In the case of social animal studies, the problem is compounded by the multiplicity of animals interacting with one another.

One way robotics researchers can help is by applying existing technologies to enhance traditional behavioural research methodologies. For instance computer vision-based tracking and gesture recognition can automate much of the tedious work in recording behavior. We use three approaches to solve the challenges and accomplish bio tracking successfully; they are Tracking Multiple Interacting Targets, Automatic Recognition of Social Behaviour and Leaning Executable Models Of behaviour.

B. Ant colony optimization

Ant colony optimization (ACO) takes inspiration from the foraging behavior of some ant species. These ants deposit pheromone on the ground in order to mark some favorable path that should be followed by other members of the colony. Ant colony optimization exploits a similar mechanism for solving optimization problems.

From the early nineties, when the first ant colony optimization algorithm was proposed, ACO attracted the attention of increasing numbers of researchers and many successful applications are now available. Moreover, a substantial corpus of theoretical results is becoming available that provides useful guidelines to researchers and practitioners in further applications of ACO.

In Ant Colony Optimization, a number of artificial ants build solutions to the considered optimization problem at hand and exchange information on the quality of these solutions via a communication scheme that is reminiscent of the one adopted by real ants Different ant colony optimization algorithms have been proposed. The original ant colony optimization algorithm is known as Ant System and was proposed in the early nineties. Since then, a number of other ACO algorithms were introduced. All ant colony optimization algorithms share the same idea.

Ant Colony Optimisation technique is used to solve Travelling Salesman problem. In the traveling salesman problem, a set of cities is given and the distance between each of them is known. The goal is to find the shortest tour that allows each city to be visited once and only once. In more formal terms, the goal is to find a Hamiltonian tour of minimal length on a fully connected graph.

In ant colony optimization, the problem is tackled by simulating a number of artificial ants moving on a graph that encodes the problem itself: each vertex represents a city and each edge represents a connection between two cities. A variable called pheromone is associated with each edge and can be read and modified by ants.

Times is specified, Times Roman or Times New Roman may be used. If neither is available on your word processor, please use the font closest in appearance to Times. Avoid using bit-mapped fonts. True Type 1 or Open Type fonts are required. Please embed all fonts, in particular symbol fonts, as well, for math, etc.

III. DETAILED DESCRIPTION

A. Bio tracking

The objective here is to show how robotics research in general and multi-robot systems research in particular can accelerate the rate and quality of research in the behavior of social animals. Many of the intellectual problems faced in multi-robot systems research are mirrored in social animal research. And many of the solutions we have devised can be applied directly to the problems encountered in social animal behavior research.

One of the key factors limiting progress in all forms of animal behavior research is the rate at which data can be gathered. As one example, Deborah Gordon reports in her book that two observers are required to track and log the activities of one ant: One person observes and calls out what the ant is doing, while the other logs the data in a notebook . In the case of social animal studies, the problem is compounded by the multiplicity of animals interacting with one another.

One way robotics researchers can help is by applying existing technologies to enhance traditional behavioral research methodologies. For instance computer vision-based tracking and gesture recognition can automate much of the tedious work in recording behavior. Three approaches are used to solve the challenges and accomplish bio tracking successfully; they are Tracking Multiple Interacting Targets, Automatic Recognition of Social Behaviour and Leaning Executable Models Of behaviour.

1) Tracking multiple interacting targets

Traditional multi-target tracking algorithms approach this problem by performing a target detection step followed by a track association step in each video frame. The track association step solves the problem of converting the detected positions of animals in each image into multiple individual trajectories. A joint particle filter tracker is proposed to perform tracking with certain modifications. The general operation of the tracker is illustrated in Figure 1. Each particle represents one hypothesis regarding a target's location and orientation. For ant tracking in video, the hypothesis is a rectangular region approximately the same size as the ant targets. In the example, each target is tracked by 5 particles. In actual experiments hundreds of particles per target are used.

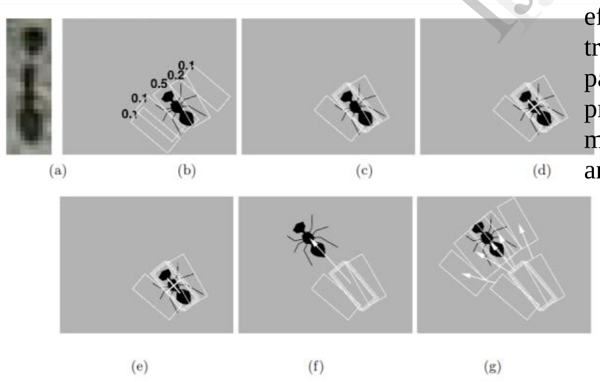


Fig 1. Ant tracking model

In Fig 1(a) the appearance model used in tracking ants is shown. This is an actual image drawn from the video data. In fig (b) a set of particles (white rectangles), are scored according to how well the underlying pixels match an appearance model. In fig (c) particles are resampled according to the normalized weights determined in the previous step. In fig (d) The estimated location of the target is computed as the mean of the resampled particles. In fig (e) the previous image and particles is shown. In fig (f) a new image frame is loaded. In (g) each particle is advanced according to a stochastic motion model. The samples are now ready to be scored and resampled as above.

Particle Filter Tracking is suitable for tracking an individual ant, but it fails in the presence of many identical targets. Hence Joint Particle Filter Tracking technique is used which applies thee additional extensions to the former method. First, each particle is extended to include the poses of all the targets (i.e. They are joint particles). Second, in the "scoring" phase of the algorithm, particles are penalized if they represent hypotheses that are unlikely because they violate known constraints on animal movement (e.g. Ants seldomly walk on top of each other).. Finally, exponential complexity of joint particle tracking must be addressed.

The computational efficiency of joint particle tracking is hindered by the fact that the number of particles and the amount of computation required for tracking is exponential in the number of targets. For instance, 10 targets tracked by 200 hypotheses each would require 20010 = 1023 particles in total. Such an approach is clearly intractable. Independent trackers are much more efficient, but as mentioned above, they are subject to tracking failures when the targets are close to one another. The advantages of joint particle filter tracking must be preserved, but avoid an exponential growth in complexity. In order to reduce the complexity of joint particle tracking targets that is in difficult to follow situations (e.g. Interacting with one another) should be tracked with more hypotheses, while those that are isolated should be tracked with fewer hypotheses. The approach is rather complex, but the effective result is that it can achieve the same tracking quality using orders of magnitude fewer particles than would be required otherwise. Figure 2 provides an example of how the approach can focus more hypotheses on challenging situations while using an overall smaller number of particles.

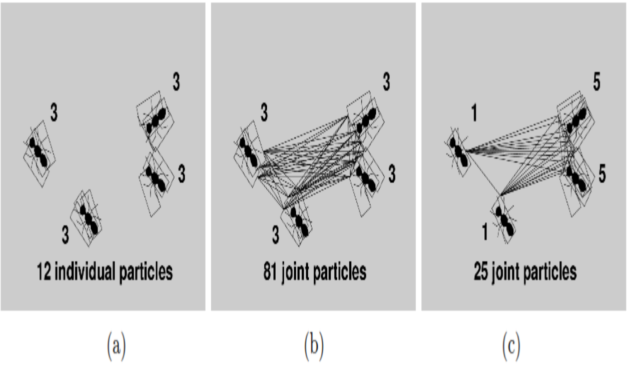


Fig 2. Ant tracking model with particles

In Fig 2: (a) Each ant is tracked with 3 particles, or hypotheses. With independent trackers this requires only 12 particles, but failures are likely. (b) With a joint tracker, to represent 3 hypotheses for each ant would require 81 particles altogether. However, MCMC sampling (c) enables more selective application of hypotheses, where more hypotheses are used for the two interacting ants on the right side, while the lone ants on the left are tracked with only one hypothesis each.

2) Automatic recognition of social behaviour

Behaviour recognition, plan recognition and opponent modeling are techniques whereby one agent estimates the internal state or intent of another agent by observing its actions. These approaches have been used in the context of robotics and multi-agent systems research to make multi-robot teams more efficient and robust. Recognition helps agents reduce communications bandwidth requirements because they can infer the intent of their team members rather than having to communicate their intent solutions because of the accompanying speed improvement.

A sensory model is used for the purpose of recognition of social behaviour. When ants approach one another, they often meet and tap one another with their antennae. The interest is in detecting and differentiating between different types of these interactions. Stephen Pratt has identified three types of encounters as the most significant for quorum detection in Temnothorax curvispinosus . The interaction types, perceived from the point of view of a particular ant (the focus ant) are: Head-to-Head (HH), Head-to-Body (HB), and Body-to-Head (BH). It may also be the case that Body-to-Body (BB) interactions are important as well, so we include them for completeness (see Figure 3).

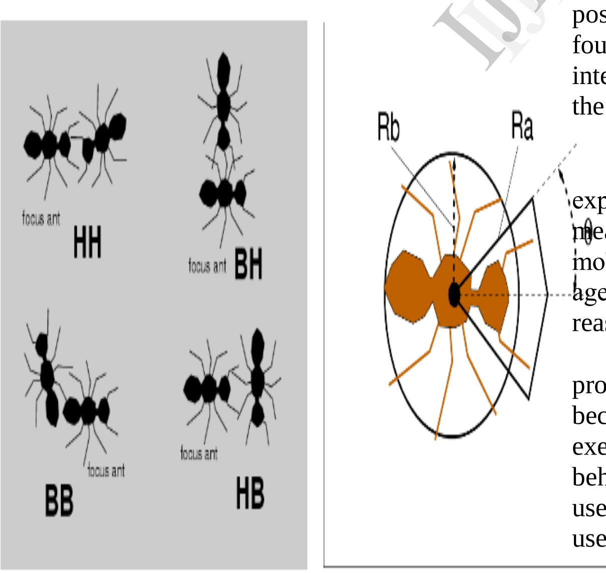


Fig 3: Left: The problem of detecting different types of encounters between ants: Head-to- Head (HH), Head-to-Body (HB), Body-to-Head (BH), and Body-to-Body (BB). Right: In one approach, a geometric model of an ant's sensory system is used. Parameters of the model include Rb the radius of the animal's "body" sensor, Ra the range of antenna sensing, and θ the angular field of view of the antennae

It is approximated that an ant's antennal and body sensory fields with a polygon and circle, respectively (Figure 3). An encounter is inferred when one of these regions for one ant overlaps a sensory region of another ant. The model is adapted from the model introduced for army ant simulation studies by Couzin and Franks

The following are details of the sensory model for Aphaenogaster cockerelli. Recall that the tracking software reports the location of the center of the ant and its orientation. It is estimated that the front of the head to be a point half a body length away from the center point along the centerline. From the head, the antennae project to the left and right at 45 degrees. A polygon is constructed around the head and antennae as illustrated, with one point at the center of the ant's thorax, two at the outer edges of the antennae and an additional point, one antenna length away, directly along the ant's centerline in front of the head. The inferred "antennal field of view" is the resulting polygon defined by these points. It is assumed a body length of 1cm and antenna length of 0.6cm. It is estimated that an ant's "body" sensory field to include a circle centred on the ant with a radius of 0.5cm.

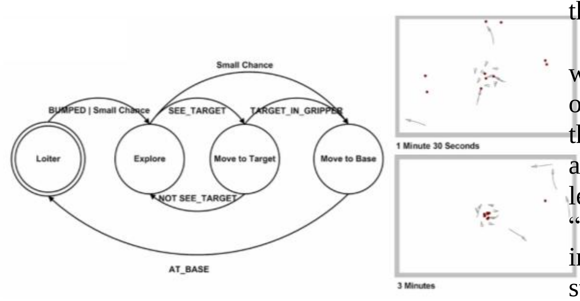
It is assumed that any object within the head sensory field will be detected and touched by the animal's antennae, and any object within the range of the body can be detected by sense organs on the legs or body of the ant. By considering separate sensory fields for the body and the antennae, it is possible to classify each encounter into one of the four types described above. To determine when an interaction occurs, its looked for overlaps between the polygonal and circular regions described above.

3) Learning executable models of behaviour

An executable model provides the most complete explanation of an agent's behavior. "Executable" means that the model can run in simulation or on a mobile robot. "Complete" means that all aspects of an agent's behaviour are described: from sensing, to reasoning, to action.

Robotics researchers are well positioned to provide formalisms for expressing executable models because the programs used to control robots are in fact executable models. Furthermore, researchers who use behaviour based approaches to program their robots use representations that are closely related to those used by biologists to describe animal behavior.

An executable model of foraging inspired by the behavior of social insects is created, using the TeamBots simulation platform and motor schema-based control. The simulation includes 12 agents and 10 objects to collect. The model is ran in simulation for 5 minutes with 12 agents at 33 frames per second, waiting for all of the targets to be carried to base, recording at each frame, the position and orientation of all agents as well as the position of all targets. (Note that the output of the simulation log file is equivalent in format to the output of the vision-based tracking software.) The objective is to learn a duplicate of the original model by examining the activities of the simulated agents recorded in the log file.



(a) (b)

Fig 4: (a) The original foraging model. (b): Two frames from the running simulation that show the agents gathering food items and dropping them at homebase in the center of the arena

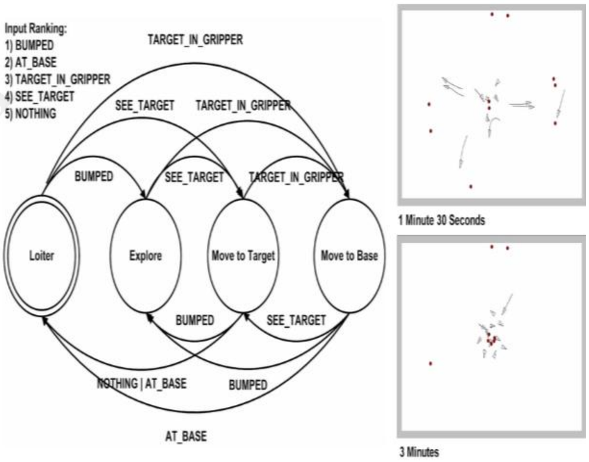
Figure 4 shows the state diagram for the original model and simulation screenshots. The agents are provided four behavioral assemblages to accomplish: loitering at the base (the center), exploring for targets, moving toward the closest target, and moving back to base. There are four binary features that trigger transitions: bumping into something, seeing a target, being near the base, and holding an object. If an agent is exploring and bumps into a loitering agent, that agent also begins to explore. Some transitions are also triggered randomly (e.g., agents eventually return to the base if they do not find a target).

A 100 trials was done using Expectation Maximization (EM) to learn 100 executable models (for both ranked and unranked input). Some of the learned models correspond well with the original models, but some do not. In order to distinguish poor models from good ones, likelihood scores for all the models were calculated using the standard forward probability recursion values. 14 ranked and 20 unranked input trials yielded models with likelihood scores higher than the original model's. It was noted that all models that successfully learned appropriate mixing weights had high likelihood scores. The highest scores for the ranked and unranked input models were essentially identical to the original model. The models with the lowest likelihood scores that were still greater than the likelihood score for the original model (in some sense the worst of the successful trials) was executed, and found these models were able to recreate the behavior of the original model. As seen in Figure 5, these models also recovered the structure of the original model.

There were flaws. In some models, bumping into an object while exploring would sometimes cause the agent to transition prematurely into the move-to-base state, slowing foraging. This appears to be a result of detection inaccuracies from bumping. Similarly, agents would drop a target early if they bumped into something while returning to base. Also if an agent was moving towards a target but another agent picked it up first, the first agent would loiter in place. One reason for these apparent failures is due to the fact that some of these situations did not occur in the training run.

Models with low likelihood scores did not perform well. Their behavior was not at all similar to the original model. The most common failure mode was the absence of an important behavioral assemblage as a state in the model (this is more or less equivalent to learning the wrong mixing weights). One example "bad" model is illustrated in Figure 4. Note that it includes two loiter states and it lacks a move-to-target state.

So far this method is applied only to data from simulation (versus live ant data). One important reason for exploring simulation results first is that by using a simulation with agents that is programmed, it is possible to directly compare the learned models with the ground-truth original. It is not possible to do this with ant data because it is impossible to know for sure how ants are really "programmed."



(a) (b)

Fig 5:(a)A learned foraging model. Note that there are more transitions between the behavioral states than in the original model. These transitions, in concert with the ranked inputs result in functional equivalence between this learned model and the original. (b): Two frames from the running simulation that shows the agents gathering food items and dropping them at homebase in the center of the arena. The behavior is nearly identical to the original.

B. Ant Colony Optimization

In Ant Colony Optimization, a number of artificial ants build solutions to the considered optimization problem at hand and exchange information on the quality of these solutions via a communication scheme that is reminiscent of the one adopted by real ants Different ant colony optimization algorithms have been proposed. The original ant colony optimization algorithm is known as Ant

System and was proposed in the early nineties. Since then, a number of other ACO algorithms were introduced. All ant colony optimization algorithms share the same idea.

Ant Colony Optimisation technique is used to solve Travelling Salesman problem. In the traveling salesman problem, a set of cities is given and the distance between each of them is known. The goal is to find the shortest tour that allows each city to be visited once and only once. In more formal terms, the goal is to find a Hamiltonian tour of minimal length on a fully connected graph.

In ant colony optimization, the problem is tackled by simulating a number of artificial ants moving on a graph that encodes the problem itself: each vertex represents a city and each edge represents a connection between two cities. A variable called pheromone is associated with each edge and can be read and modified by ants.

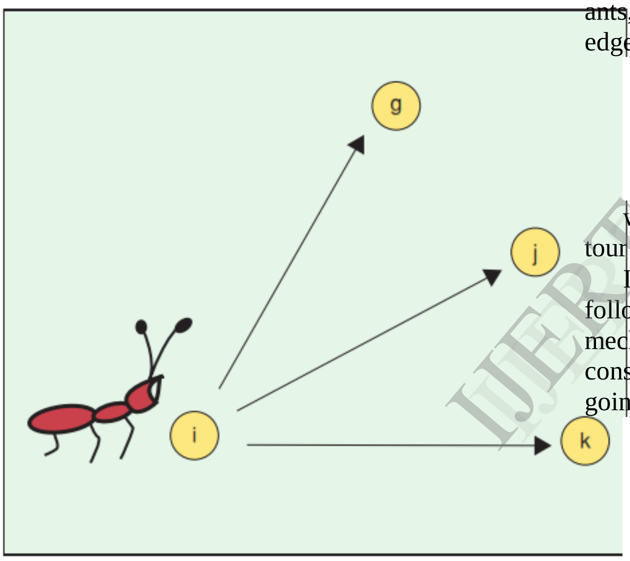


Fig 6: An ant in city *i* chooses the next city to visit via a stochastic mechanism: if *j* has not been previously visited, it can be selected with a probability that is proportional to the pheromone associated with edge *(i, j)*.

Ant colony optimization is an iterative algorithm. At each iteration, a number of artificial ants are considered. Each of them builds a solution by walking from vertex to vertex on the graph with the constraint of not visiting any vertex that she has already visited in her walk. At each step of the solution construction, an ant selects the following vertex to be visited according to a stochastic mechanism that is biased by the pheromone: when in vertex *i*, the following vertex is selected stochastically among the previously unvisited ones (see Figure 6). In particular, if *j* has not been previously visited, it can be selected with a probability that is proportional to the pheromone associated with edge *(i, j)*.

At the end of an iteration, on the basis of the quality of the solutions constructed by the ants, the pheromone values are modified in order to bias ants in future iterations to construct solutions similar to the best ones previously constructed.

Ant colony optimization has been formalized into a meta heuristic for combinatorial optimization problems. A metaheuristic is a set of algorithmic concepts that can be used to define heuristic methods applicable to a wide set of different problems. In other words, a metaheuristic is a general-purpose algorithmic framework that can be applied to different optimization problems with relatively few modifications.

The first algorithm used to implement Ant Colony Optimization is Ant System. Its main characteristic is that, at each iteration, the pheromone values are updated by *all* the *m* ants that have built a solution in the iteration itself. The pheromone *τij* , associated with the edge joining cities *i* and *j*, is updated as follows:

$$\tau_{ij} \leftarrow (1-\rho) \bullet \tau_{ij} + \qquad \tau^k_{ij} \,, \qquad (2)$$

where ρ is the evaporation rate, *m* is the number of ants, and τ^k_{ij} is the quantity of pheromone laid on edge *(i, j)* by ant *k*:

$$\Delta\tau^k_{ij} = \begin{cases} Q/L_k & \text{if ant } k \text{ used edge } (i,j) \text{ in its tour,} \\ 0 & \text{otherwise,} \end{cases} \qquad (3)$$

where *Q* is a constant, and *L k* is the length of the tour constructed by ant *k*.

In the construction of a solution, ants select the following city to be visited through a stochastic mechanism. When ant *k* is in city *i* and has so far constructed the partial solution *s p* , the probability of going to city *j* is given by:

$$p^k_{ij} = \begin{cases} \dfrac{\tau^\alpha_{ij}\cdot\eta^\beta_{ij}}{\sum_{c_{il}\in\mathbf{N}(s^p)} \tau^\alpha_{il}\cdot\eta^\beta_{il}} & \text{if } c_{ij} \in \mathbf{N}(s^p), \\ 0 & \text{otherwise,} \end{cases} \qquad (4)$$

where **N**(*s p*) is the set of feasible components; that is, edges *(i, l)* where *l* is a city not yet visited by the ant *k*. The parameters *α* and *β* control the relative importance of the pheromone versus the heuristic information *ηij* , which is given by:

$$\eta_{ij} = \frac{1}{d_{ij}} \,, \qquad (5)$$

where *d ij* is the distance between cities *i* and *j*.

1) Application of Ant Colony Optimization

In recent years, the interest of the scientific community in ACO has risen sharply. In fact, several successful applications of ACO to a wide range of different discrete optimization problems are now available. The large majority of these applications are to *NP* -hard problems; that is, to problems for which the best known algorithms that guarantee to identify an optimal solution have exponential time worst case complexity. The use of such algorithms is often infeasible in practice, and ACO algorithms can be useful for quickly finding high-quality solutions. Other popular applications are to dynamic shortest path problems arising in telecommunication networks problems. The number of successful applications to academic problems has motivated people to adopt ACO for the solution of industrial problems, proving that this computational intelligence technique is also useful in real-world applications.

a) Applications to NP -Hard Problems

The usual approach to show the usefulness of a new meta- heuristic technique is to apply it to a number of different problems and to compare its performance with that of already available techniques. In the case of ACO, this type of research initially consisted of testing the algorithms on the TSP. Subsequently, other *NP* -hard problems were also considered. So far, ACO has been tested on probably more than one hundred different *NP* -hard problems. Many of the tackled problems can be considered as falling into one of the following categories: routing problems as they arise, for example, in the distribution of goods; assignment problems, where a set of items (objects, activities, etc.) has to be assigned to a given number of resources (locations, agents, etc.) subject to some constraints; scheduling problems, which–in the widest sense–are concerned with the allocation of scarce resources to tasks over time; and subset problems, where a solution to a problem is considered to be a selection of a subset of available items. In addition, ACO has been successfully applied to other problems emerging in fields such as machine learning and bioinformatics.

Common to many of these applications is that the best performing ACO algorithms make intensive use of the optional local search phase of the ACO meta heuristic. This is typically very effective since, on the one hand, the solutions constructed by the ants can often be improved by an adequate local search algorithm; on the other hand, generating proper initial solutions for local search algorithms is a difficult task and many experimental results show that the probabilistic, adaptive solution generation process of ant colony optimization is particularly suited to this task. ACO algorithms produce results that are very close to those of the best-performing algorithms, while on some problems they are the state-of-the-art. These latter problems include the sequential ordering problem, open-shop scheduling problems, some variants of vehicle routing problems, classification problems, and protein-ligand docking

b) Applications to Telecommunication Networks

ACO algorithms have shown to be a very effective approach for routing problems in telecommunication networks where the properties of the system, such as the cost of using links or the availability of nodes, vary over time. ACO algorithms were first applied to routing problems in circuit switched networks (such as telephone networks) and then in packet-switched networks (such as local area networks or the Internet). Following the proof of concept provided by Schoonderwoerd et al., ant-inspired routing algorithms for telecommunication networks improved to the point of being state-of-the-art in wired networks. A well-known example is AntNet. AntNet has been extensively tested, in simulation, on different networks and under different traffic patterns, proving to be highly adaptive and robust. A comparison with state-of-the-art routing algorithms has shown that, in most of the considered situations, AntNet outperforms its competitors.

Ant-based algorithms have given rise to several other routing algorithms, enhancing performance in a variety of wired network scenarios. More recently, an ACO algorithm designed for the challenging class of mobile ad hoc networks was shown to be competitive with state-of- the-art routing algorithms, while at the same time offering better scalability.

IV. EVALUATION

The impressive performance of Swarm Intelligence in discrete and continuous optimization problems has increased the attention of many researchers with different backgrounds to apply Swarm Intelligence approach into their own research areas. As a result, there has been an almost exponential increase in the number of research papers reporting the successful application of Swarm Intelligence based algorithms in a wide range of domains, including combinatorial optimization problems, function optimization, finding optimal routes, scheduling, structural optimization, image analysis, data mining, machine learning, bioinformatics, medical informatics, dynamical systems, industrial problems, operations research, and even finance and business.

The potential of Swarm Intelligence is yet far from being exhausted with many interesting applications still to be explored, especially in bioinformatics. In the past few years, there has been a slow, yet steady increase in the number of research papers that have successfully applied SI algorithms in bioinformatics. This is because several tasks in bioinformatics involve optimization of different criteria (such as, energy, alignment score, overlap strength, etc.), and the various applications of SI algorithms proved them to be efficient, robust and computationally inexpensive optimization techniques , which made their applications in bioinformatics more obvious and appropriate .

It is worth noting, however, that Swarm Intelligence -based algorithms do not fully show their competitive edge over other optimization techniques on static problems whose characteristics and conditions do not change over time. Nevertheless, they are often more competitive to deterministic approaches3 in dealing with uncertainty, as well as general- purpose heuristics (e.g., hill climbing and simulated annealing approaches) in dealing.

V. CONCLUSION

Swarm Intelligence has got many advantages. It is now being considered as one of the most promising Artificial Intelligence techniques with steady growing scientific attention. This can be supported by the increasing produced number of successful Swarm Intelligence research output, as well as the rapidly expanded conferences and journals dedicated for Swarm Intelligence. In this paper three important Swarm Intelligence techniques is discussed, namely Bio Tracking, Ant Colony Optimization and Particle Swarm Optimization. Each has its advantages and disadvantages.

Swarm Intelligence systems are highly scalable; their impressive abilities are generally maintained when using groups ranging from just sufficiently few individuals up to millions of individuals. In other words, the control mechanisms used in Swarm Intelligence systems are not too dependent on swarm size, as long as it is not too small. SI Systems respond well to rapidly changing environments, making use of their inherit auto-configuration and self-organization capabilities. This allows them to autonomously adapt their individuals 'behaviour to the external environment dynamically on the run-time, with substantial flexibility.

The potential of swarm intelligence is indeed fast-growing and far-reaching. It offers an alternative, untraditional way of designing complex systems that neither requires centralized control nor extensive pre-programming. That being said, Swarm Intelligence systems still have some limitations. Because the pathways to solutions in SI systems are neither predefined nor pre-programmed, but rather emergent, Swarm Intelligence systems are not suitable for time-critical applications that require (i) on-line control of systems, (ii) time critical decisions, and (iii) satisfactory solutions within very restrictive time frames, such as the elevator controller and the nuclear reactor temperature controller. It remains to be useful, however, for non-time critical applications that involve numerous repetitions of the same activity

However, the advantages of Swarm Intelligence make it suitable to be applied successfully in various fields. Nature is a rich inspirational source and there is still much to learn from. With the help of Swarm Intelligence can take advantage of the social collective behaviour of swarms to solve our real-life problems, by observing how these swarms have survived and solved their own challenges in nature. Different Swarm Intelligence based computational models are fast-growing, as they are generally computationally inexpensive, robust, and simple.SI-based optimization techniques are far-reaching in many domains, and have a wide-range of successful applications on different areas.

BIBLIOGRAPHY

[1] "Swarms and Swarm Intelligence", Michael G. Hinchey, Loyola College in Maryland Roy Sterritt, University of Ulster Chris Rouff, Lockheed Martin Advanced Technology Laboratories, April, 2007

[2] T. Balch et al., "How A.I. and Multi-Robot Systems Research Will Accelerate Our Understanding of Social Animal Behavior," Proc. IEEE, July 2006, pp. 14451463) Symantec Corporation, Understanding Heuristics: Symantec's Bloodhound Technology," Symantec White Paper Series.

[3] "Flocks, Herds, and Schools: A Distributed Behavioral Model" (Proc. 14th Ann. Conf. Computer Graphics and Interactive Techniques, ACM Press, 1987, pp. 25-34

[4] "Ant colony optimization", Dorigo, Marco Univ. Libre de Bruxelle, Brussels, Computational Intelligence Magazine, IEEE, November, 2006

[5] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," Proceedings of IEEE International Conference on Neural Netwroks, Vol. IV, pp. 1942-1948, Perth, Australia, 1995.

[6] D. Martens, M. De Backer, R. Haesen, M. Snoeck, J. Vanthienen and B. Baesens, "Classification with Ant Colony Optimization," IEEE Trans. Evolutionary Computation, vol. 11, no. 5, pp. 651-665, 2007.