Model Of Ant-Based Robot Team For Exploratory Missions

Ninad N. Pradhan
ninad.pradhan@rediffmail.com

Abstract

Ants exhibit a variety of behaviours which synergise to result in discovery and exploitation of food sources around the nest. These behaviours can be adapted to robot teams which are performing exploratory missions on uncharted landscapes. Also, aspects of mobile robot coordination and control can be adapted to enhance the functionality of the team. This paper proposes a structure for a team of ant-emulating robots, a strategy for field operation, and behaviours which can be incorporated into the mobile agents on a foraging or exploratory mission.

1. Introduction

Ants use the phenomenon of pheromone deposition to mark their trail from the nest to food and back. This pheromone has a time variant strength after deposition. In certain ant species, the strength of the pheromone is proportional to the quality of the food source discovered by them. Once food has been discovered, mass recruitment is used in some species as a tool for gathering foragers to exploit a particular food source. Mass recruitment is a direct recruitment mechanism. Another mechanism for recruitment is indirect, known as allelomimetic communication. Here, the pheromone trails deposited by ants are followed by other foragers who come across them in their random forays through the search area. They leave their own trails and strengthen the pheromone trail such that the probability of the trails being followed by other unsuccessful foragers, and a shortest path to food being traced, is high. Ants retire to their nest if their foraging trip is unsuccessful after some time.

In this paper, I have proposed amalgamating the ant behaviours listed above to make a usable model of a team of robots. Ant colonies are distributed control systems which use a combination of individual behaviours to create a complex and intricate team structure. However this paper does not propose a fully distributed model. The concept of a team leader or a central overseer is brought into play. This larger robot can fulfill the requirements of housing the mobile robots, triangulating position, creating and relaying a global map of the territory to the mobile 'ants' from local information. It can also make macro decisions about the strategy to be followed in a sector of the search area.

It is proposed (*see* **Figure 1**) that the team leader will itself be mobile, and divide a large search area into smaller areas, each of which will constitute an exploratory mission. The ants will be of two types: mappers and foragers, only one of which are usually active at a time during the mission. They will exhibit eight behaviours: Waiting, Searching, Exploiting, Recruiting, Following, Mapping, Calling and Retiring. They should have a finite capability of performing useful tasks even in the breakdown of communication from the team leader. The end result should be exploration of an area, exploitation of food sources inside it, and optimal use of the agents available for this effort.

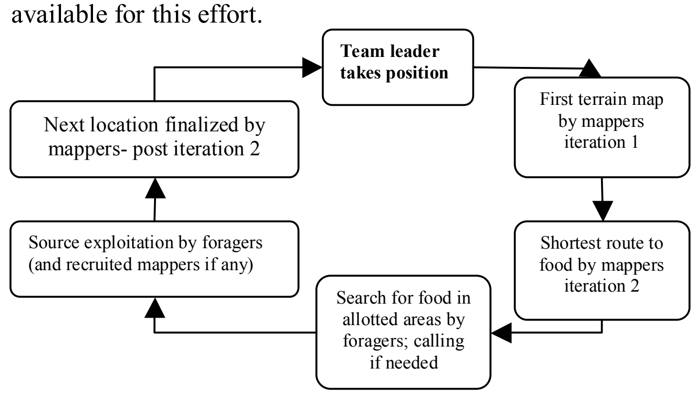


Figure 1: Broad strategy

2. Team composition

2.1. The team leader or nest

The words 'team leader' and 'nest' are used interchangeably in the paper. They can be taken in the context of functions which are direct analogies to ant behaviour and those which are modifications made for better teamwork in robots.

Ant robots reside in the nest and return to it at the end of their foraging journeys. All foraged food is collected and stored at the nest. It is also the origin of the coordinate system for purposes of localisation and path tracing for each individual robot. Hence, the

word 'nest' is often used where the foragers, pheromones or the instinct to return to base is referred to.
The nest is assumed to be a mobile vehicle, unlike those in real ants. This property is essential for the. The team leader is a central overseer of activity, something which does not exist in real ant colonies, which can be considered to be fully distributed control systems. It contains localisation pingers situated a fixed distance apart for the purpose of triangulating robot position. It also performs the computationally complex task of mapping the environment based on local information provided by different agents. It has the capability of supplying team robots with orientation and strategy information. The team leader may or may not be fully autonomous. It depends on how many or few activities it is supposed to coordinate on the field. If certain decisions that it makes are to be highly intelligent, and beyond the scope of existing architectures for control, then the team leader may well be a manned vehicle with a human making the difficult choices (not necessarily in terms of risk, but in terms of complexity) before or during a mission.

2.2. The field robots or 'ants'

The proposed robot team is to be a amalgam of heterogeneous and homogeneous; as well as that of central and distributed control. The element of central control is introduced through the presence of a mobile team leader or 'nest'. There is heterogeneity in the fact that there are two distinct functional agents at work, to distribute labour [8].
The first category of functional agents is that of the mappers, whose objective is that of speedy coverage of an area whose topology is yet unknown to the mission. Mappers can provide information about major or impassable obstacles or valleys which lie inside the search area as well as provide preliminary knowledge about the topology of the terrain itself. Mapper-provided information can also decide the location of the nest for actual excavation activity.
The second category of agents is that of the foragers, who perform the task of confirming a food source, forage at source as well as incorporate different behavioural attributes for the teamwork by itself to be closely knit. Some behavioural attributes [4] have already been proposed. Foragers are proposed to be the more intelligent of the two robots in that they take more independent decisions on the field.
The entire team as such is heterogeneous since it contains two distinct types of agents. However, it is homogeneous in the sense that at a given time, only one type of agent is working on the field. This agent can perform the foraging functions independent of a specialist [12]. The recruitment of waiting robots, (*section 7*) introduces an element of heterogeneity, but it has to be appreciated that the recruited team member is wholly dependent on a 'leader' and hence is only an extension of the recruiter for its role in the mission.

2.3. The possibility of modular architecture in the team

The example of Millibots [12] demonstrates the effectiveness of a modular architecture in robots. Modularity is possible on two levels in this team. The first is the modularity on the hardware level. Different sensing and computing systems can be added and taken out as desired. The I^2C bus architecture has been discussed with reference to Millibots in [12]. The second is that on the hardware level, the physical construction of the agent. Mappers and foragers have easily interchangeable faculties, which lend to ease in modularity in architecture.

3. Behavioural states in the ant team model

The five behavioural states in [4] are waiting (W), searching (S), exploiting (E), recruiting (R), and following (F). We have made use of the mapping (M), calling (C) and retirement (B) state as per the features of the team. The significance of these states is:
W: only in mappers. They wait to be recruited after completing the second iteration. (*section 7*)
S: only in foragers. Foragers search their area and try to identify food sources within it.
E: both foragers and mappers. Samples are taken from the food source as per mission requirements.
R: only in foragers. They return to nest to recruit a mapper if absence of response to call (*section 7*)
F: only in mappers. They follow the recruiter to the food source.
M: only in mappers. It is active during first and second iteration when they are mapping the area.
C: only in foragers. They call for assistance if they discover a food source larger than their foraging capability.
B: only in foragers. They start to go back to nest if they fail to locate a food source in a certain time (*section 5*), and make themselves available for assistance to call.
Recruiting (**R**) is not group recruiting [4] but instead is single recruiting in which each forager returns with only one idle member to the site. The only two states whose status needs to be actively known to other team members during an exploration are the calling (**C**) and recruiting (**R**) states.

4. Role of Mappers

4.1. The need for mappers

The way the teams are organised, mappers have the ability to forage, when they are recruited; they can contribute to the mapping activity. The division of the search area for mappers is also similar, in the second iteration (*sections 3.2, 9*) to that of the foragers. What is the need for another type of robot, when the mappers by themselves can perform foraging operations and reduce both the time and the resources required for an operation?
Mappers do not have the ability to identify food sources. They also do not have the hardware required to communicate between robots and respond to calling robots. The absence of such hardware implies they may be risked in unstable or uncertain environments with a comparatively smaller risk. Mappers can have shortest path detection algorithm built in. Food carrying foragers can benefit from this algorithm and return to the nest with food expending lesser energy than they would have to if they had to find the shortest path on their own, laden with extra weight. Also, foragers are not wasted in forays to search areas which are completely inaccessible. The resulting distribution of foragers and division of search areas by the team leader is more streamlined.

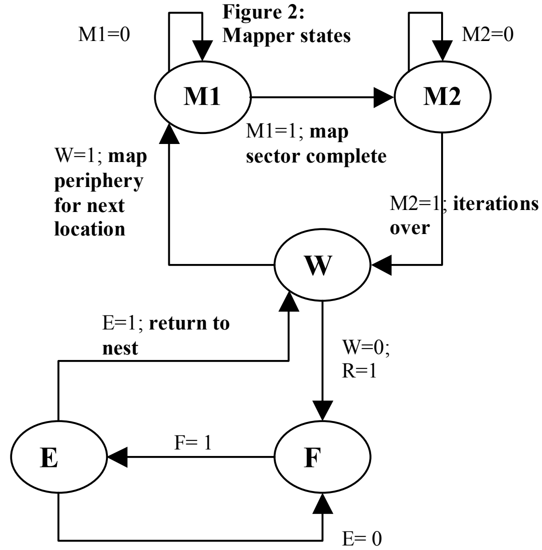
4.2. Two iterations of mappers

Mappers will be sent out in two iterations.
The first iteration of the mappers aims to achieve the following three objectives:
a. Generate a global map of the entire area under observation
b. Identify completely inaccessible areas
c. Decide on a suitable location for the team leader for the next mission.
There is a case for sending a second iteration of the mappers once the search areas have been allocated. The second iteration will map the straight line to search area and optimize this route for the foragers to follow.

4.3. Generation of a global map

The combination of local information to generate a global map [12] is suitable since there are many mappers out on the field at the same time. The sensory inputs required to build an occupancy grid [14] are to be given to the team leader. These can be time division multiplexed, giving us the advantages of TDM communication: saving of power, transmission in the same bandwidth slot. Now the path for each robot is fixed. So at each time slot of transmission the team leader knows where the sensor update is coming from with respect to global coordinates. Coordinates of robot may be transmitted along with the mapping data. Over the entire mapping time of the mission, a consolidated picture of the operating environment is constructed. If the environment is unstable, then the mapping faculty can also be given to foragers, so that an updated map can be built during foraging in real time by the team leader. There are clear advantages in the map construction process for explore and progress missions, if some other missions are to take place in the same search area. A built map can be conveyed to following missions.

Figure 2: Mapper states



4.4. Preparation for next mission

The team leader keeps exploiting the terrain by breaking a massive area search into many smaller missions each working on the same map-optimise-forage routine. Each time a smaller mission has been completed, the team leader moves out of the exploited area, enters an unmapped zone, and sends mappers out to initiate the first of the set of operations for a mission. Mappers can play a part in deciding the route out of the present search area and location of the team leader in the next. Location can be fixed by having some of the mappers explore the fringes of the present search area and detect obvious obstacles which might be present at the periphery. The relocation of some mappers to fringe areas in a mission is another aspect of heterogeneity by way of which some mappers can be put to good use when foragers are at work.

5. Retirement of a searching robot

Retirement is the abandoning of a search effort. According to [4], a searching ant will travel a certain time until it decides to abandon the search and return to the nest. Only a retiring robot is eligible to respond to a call for assistance from a team mate. Hence retirement must come at an appropriate time and such that resources are diverted to successful finds ultimately.

The problem arises for agents which follow a 'retire if nothing has been found in entire area' technique. A successful forager will inevitably give a call before an unsuccessful robot has asserted that nothing of interest is visible in its area, which the unsuccessful robot will be unable to respond to since it is still not retiring. This will lead to a situation where the probability of waiting robots at the nest being recruited is high.

We do not ideally want this to happen since the functionality of the mappers is lesser than the foragers. One way of dealing with it is giving a time bound strategy where robots start to retire if they have not found anything up to a fixed time. But a purely time bound strategy may prove to be ineffective for a few reasons:

* the search area assigned to a robot may contain more obstacles as an effect of which a robot will cover less search area compared to a robot in a relatively plain landscape in the same time; this will be an inefficient use of resources

* the approach to the search area may contain obstacles

5.1. Retirement time equation

An addition can be made to the time bound strategy by introducing an allowance for the time taken to get around obstacles in or on the way to the search zone of a robot. The path to the centre of the search zone is a straight line (in ideal situations) from the nest. The path around the centre of the search zone is random during the first iteration or foray. Hence the two time allowances can be made separately. A proposed equation for this retirement time is:

$$\mathbf{R_i = R_{oi} + g[\ \sum(\sum dev\ (t))\ ] + h[\ \sum O\ (T- t)]}$$

$\mathbf{R_{oi}}$ is the retirement time in absence of obstacles which may or may not be the same for all robots.

$\mathbf{g[\ \sum(\sum dev\ (t))\ ]}$ The summations term is for approach to centre of search zone. Inner summation is deviation at 't' going from 0 to t . The deviation from straight line path is calculated and stored at discrete time s intervals, ending at time t when the robot is back on course, for one obstacle. This is done for all s obstacles, outer summation. The parameter g is used for scaling.

$\mathbf{h[\ \sum O\ (T- t)]}$ O is the number of obstacles found around search area. The factor (T-t) and the summation ensure that the obstacles in search path add only a finite value to the retirement time, since they are being considered only in this (T-t) window.

5.2. Implications of the retirement time equation

* When there is a obstacle free arena, all the robots will retire at times Roi which may be all the same or different. These can be adjusted for a scenario where we want to increase the time spacing between different robots being available for calls.

* Robots are given a better opportunity of scavenging an area before they retire or make themselves available to answer calls from others, since obstacles are considered in the equation.

* Different robots should be available for responding to calls at different times, because of the number of variables in the retirement time equation. This reduces the waiting time between a call and the response to it.

5.3. Map updating for use by retiring robots

Retiring robots need to get a map update to know if there are any calls for assistance from successful foragers. Team leader can provide such updates which give information about:

* the status of potential food sources (as confirmed or possible sources)

* the status of trails from food sources (colour coded according to strength)

Foraging and shortest path decisions can then be left to the algorithms being run on the individual machines, thus giving them greater control over their own actions.

6. Calling for assistance

'Calling', in this paper, refers to one robot inviting another to forage at a food source it has located in its area. This food source may either be too extensive for one robot to handle, or one amongst many, which is why another robot in the same area would be useful. This is an interpretation of the pheromone laying process.

6.1. Trail strength and its relation to a called robot

According to [4], the trail response is a S-shaped function which saturates with time. As noted before, the strength of a trail will depend on: the strength of a food source and time elapsed. The former is a direct relation whereas the latter is an inverse relation. A third factor comes into play; which is the number of robots already foraging at a site. A trail strength function needs to be devised for the map update on calling and food sources. Considerations:

* A called robot will add to existing pheromone strength on the trail
* Trail strength has to decrease as time elapses
* The trail strength has to be weighed against the distance from the trail to make a final foraging choice
* The strength of a source has to be somehow used to decide the saturating number, maximum requirement of foragers at a source.
* Suppose a robot is receiving calls from two robots simultaneously; it should visit the non-saturated robot.
* The robots on the field have to make choices purely based on trail strength, saturation and straight line distance from trail.

6.2. Trail strength function

The trail strength function (TSF) as calculated by an individual robot 'i' is proposed to be:

$$\mathbf{TSF(i)} = \frac{[a\,S_f + b\,d_f]Q}{d_i\,m^t}$$

where:
S_f absolute strength of food source f
d_f distance of calling forager from source f
d_i straight line distance of the called robot from pheromone trail i
Q=1 if trail not saturated; Q=0 if trail is saturated
m^t is time elapsed after calling started; m is an arbitrary constant
a,b are constants chosen through simulation results
d_f ensures that the trail does not go cold when the successful forager is returning to nest with the food; it gives the forager till the last moment before it takes on the responsibility of recruiting from the nest. Trail strength actually increases as the forager approaches the nest even though it is physically away from source.
Q inhibits the trail once saturation has been achieved. The next map update will automatically give a zero trail after saturation. A robot will receive the TSF through a map update; this does not contain the information about d_i. It will calculate straight line distance and divide this by d to get its own TSF and then make a choice. The choice will be based on the higher valued TSF out of the trails to which it has been called to assist.

6.3. Calling a robot- a beacon deposition method

The beacon deposition method is an alternative to map update method and makes the system more distributed.
The forager can forage only at 1 source at a time, leaving other sources unattended. It can deposit beacons to attract retiring ants. A potential problem is that the deposited beacons could pulse at overlapping time-slots, giving confusing signals to a called robot. To resolve this, all the beacons (their total available number known and fixed) can be synchronised to pulse so that each has a fixed pulse slot when put on the field.
Issues:
* How will a robot know if any other robot is tracking the same beacon?
* How will a robot be able to 'home in' on a particular beacon?
Homing in is done by ants by following a system of sensing the concentration gradient of pheromone deposition around them [8]. Alternately, the beacon relays to the team leader, which calculates its position and relays the position to all robots. Only the retiring robot or robot may use this information for tracking

7. Recruiting

Direct recruitment is initiated when there is no response to a call for foraging assistance. To enable a choice between two recruiting robots, a figure of merit is proposed. Moreover, since distance and quantity are dimensionally different, we can normalise them to D and Q respectively. Thus:
FOM= [αQ]/[ßD]
alpha and beta are for scaling.
The calling robots at the nest can simply relay this FOM to the waiting robot. Waiting robot will decide to follow the robot with the higher FOM.

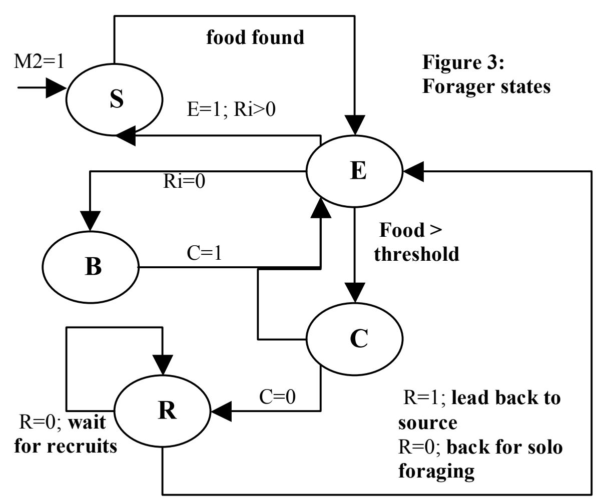


Figure 3: Forager states

8. Localisation

Since the route followed by the robot is predictable, internal dead reckoning [12] seems to be adequate for robots in this team.
Position updates on map can be event based, when the robot encounters an obstacle or a food source. This will refresh information available with the leader and eventually benefit the rest of the team during the mission. The robot can refresh its position information by recalculating it using the pingers located with the team leader [10].
A robot without external localisation will only be hindered by the non-availability of assistance at foraging and reduced accuracy, but will still be able to find its way to its target area and back. Localisation based on pingers transmit-receive is available for confirmation of position. Localisation based on collaboration [12] can be further incorporated to improve accuracy if that is desired.

9. Shortest path to food source

Real ants are observed to come up with shortest paths to food source as an emergent property of their behaviour [6]. This is possible since ants forage in large numbers. An optimal solution often results from their behaviour. The method of stimulation of worker ants by pheromone deposition is called stigmergy [11]. This cannot be adapted directly to robots since robots operate with reduced numbers.
The second iteration of mappers could be sent after the first iteration has mapped the terrain. They could follow the same path allotted to the foragers, and navigate completely around obstacles in their journey to and from the search area to zero in on the shortest possible path to the periphery of a forager's search area, a process similar to allelomimetic communication [9].

10. Failures in the team

Even if a robot loses communication link with the team leader it still should be able to perform a set of useful functions and return to the nest. The robot after losing contact can immediately estimate its position from dead reckoning which followed the last map update. It can then fix this position as the local origin around which it can perform exploratory excursions. On revival of contact it can return to the local origin, get its actual global location and continue as per mission strategy.
Robot failure can either be detected or inferred by the team leader. Detection can come from a communication provision; inference when a robot does not acknowledge some regular communication with leader. The solution to this can be the team leader generating a pheromone trail to the robot's zone having high trail strength, so that a retiring robot can substitute for the failed robot, or possibly investigate and rectify some obvious problem with the robot.

11. Conclusion

This paper has presented a strategy to coordinate the actions of a robot team such that they pick useful features from ants and add a few features typical to mobile robots. The theme can be further worked on by simulating the scenario to tweak equation parameters, and compare with existing models for exploration teams of mobile robots. Actual implementation of such a team will also present a few issues to be addressed, and suitable changes may be made in the framework of the robot team.

References

[1] Umre, Ashish (2001). Social Insect Analogies for Distributed Communication Networks, From Worker to Colony: Understanding the Organisation of Insect Societies, IUSSI Meeting, Dec'2001. Isaac Newton Institute, Cambridge, UK.
[2] B. Bullnheimer, R.F. Hartl, and C. Strauss. An Improved Ant System Algorithm for the Vehicle Routing Problem. Annals of Operations Research
[3] D. J. T. Sumpter, M. Beekman (2003). From nonlinearity to optimality: pheromone trail foraging by ants. ANIMAL BEHAVIOUR 66, 273-280
[4] D. J. T. Sumpter, S. C. Pratt (2003). A modelling framework for understanding social insect foraging. BEHAVIORAL ECOLOGY AND SOCIOBIOLOGY **53**, 131-144.
[5] Di Caro G. and M. Dorigo (1997). AntNet: A Mobile Agents Approach to Adaptive Routing. Tech. Rep. IRIDIA/97-12, Université Libre de Bruxelles, Belgium
[6] Dorigo M., G. Di Caro and L.M. Gambardella (1998): Ant Algorithms for Discrete Optimization, Technical Report IRIDIA/98-10, Universite Libre de Bruxelles, Belgium. To appear in Artificial Life.
[7] Couzin, I. D. & Franks, N. R. (2003) Self-organised lane formation and optimised traffic flow in army ants. Proceedings of the Royal Society of London, Series B. **270**, 139-146.
[8] Jean-Philippe Rennard (2003) Social insects and self-organization. http://www.rennard.org/alife -- alife@rennard.org
[9] Leerink L.R., S.R. Schultz and M.A. Jabri. (1995). A Reinforcement Learning Exploration Strategy based on Ant Foraging Mechanisms. Proceedings of the Sixth Australian Conference on Neural Networks, Sydney, Australia, 1995
[10] Lynne E. Parker, Heterogeneous Multi-Robot Cooperation, Massachusetts Institute of Technology Ph.D. Dissertation, January 1994. Available as MIT Artificial Intelligence Laboratory Technical Report 1465
[11] Holland, O. Melhuish,C., 'Stigmergy, self-organisation, and sorting in collective robotics', Artificial Life, 5:2 (1999) pp.173-202
[12] Grabowski, R., et al, "Heterogeneous Teams of Modular Robots for Mapping and Exploration," Autonomous Robots, Vol. 8, No. 3, June 2000, pp. 293-308
[13] Balch, T. The impact of diversity on performance in multirobot foraging. In Proc. Autonomous Agents 99, Seattle, WA, 1999
[14] Balch, T.Grid-Based Navigation for Mobile Robots, The Robotics Practitioner, 2(1), 1996.
[15] Balch, T., Boone, G., Collins, T., Forbes, H., MacKenzie, D., and Santamar'ia, J. "Io, Ganymede and Callisto - a multiagent robot trash-collecting team", AI Magazine, 16(2):39--51, 1995
[16] Z. Butler, R. Fitch, D. Rus and Y. Wang, "Distributed Goal Recognition Algorithms for Modular Robots", ICRA 2002