Biologically Inspired Decision Making for Collective Robotic Systems

Chris A. C. Parker
Dept. of Computing Science
University of Alberta
Edmonton, Alberta, Canada
Email: parkerca@ieee.org

Hong Zhang
Dept. of Computing Science
University of Alberta
Edmonton, Alberta, Canada
Email: zhang@cs.ualberta.ca

Abstract—Practical collective robotic systems likely will be confronted with problems which have more than one unique solution. When deciding on which of a set of candidate solutions to a problem to pursue, a collective system should ensure that its members reach a unanimous decision regarding which solution to implement so that the system itself does not split apart with different members pursuing different solutions. If such a split were to occur, much of the collective system’s functionality could be lost. In this paper, we present a unique approach to collective decision making that is based on an algorithm employed by a particular species of ant when it chooses a new nest site. We expand the ants’ algorithm into a general purpose decision making scheme and apply it to the collective relocation problem. A detailed study of the performance of our decision making algorithm was carried out in simulation using the collective relocation task as a test bed. Consistent system performance was observed across three robot populations. It was found that one particular system variable, the decision quorum threshold played a large role in determining the system’s behaviour and that system behaviour was maximized when this variable was set to 50% of the system’s population.

I. INTRODUCTION

Collective robotic systems - systems that are composed of multiple robots - need to be able to make decisions just like their solitary counterparts. How a collective robotic system should make decisions is not readily apparent. In collective robotics, one often refers to the global behaviour of a system and the individual behaviours of the robots that make up the system. The global behaviour of a system is the accumulation of the behaviours of the individual robots that make up the system. To say that a collective robotic system makes a decision implies that the individuals that make up the system will alter their behaviours such they act in a manner that will be consistent with the decision made. Thus the global behaviour of a system after a decision has been made by that system should be consistent with its decision [5]. Much of the research being done in collective robotic systems has focused on systems where there is no centralized controller. How, then, can consensus be reached amongst the robots that make up such a collective system so that a global, or system level decision can be arrived at in a reliable fashion?

Sports teams are one example of collective systems with little or no centralized control. In order to be successful, a team in a competitive sport must be able to make critical decisions quickly and coherently in an adversarial environment. If half of a team implements one strategy and half implements another, both strategies could fail because they might be in conflict. Imagine the case where two forwards of a soccer team are about to pass and shoot the ball. If both individuals decide that they will receive the pass, then the system level plan to pass and then shoot will fail. Decision making in robotic sports teams has been studied by several groups of researchers through the RoboCup robotic soccer competitions as well as in other venues. Bowling et al. describe a decision making scheme to allow the robots of an F180 (Small Size League) RoboCup robotic soccer team to quickly choose plays from a play book in an changing environment in [1]. Here, the options or plays that a team must choose from are static. In [6], another decision making mechanism for robotic soccer was presented that involved coordination graphs. In both of these works, the notions of system level decisions and consensus amongst the team members were prominent.

In this paper, we present a unique technique for a collective robotic system to approach decisions to be made that is based on a decision making strategy employed by particular species of ant. In the next section, we will discuss collective decision making in biological systems. Then, in Section III, we will expand the nest site selection algorithm of the ant Leptothorax albipennis into a general purpose problem solving/decision making algorithm. Section IV applies our algorithm to a specific multiple robot system task: the collective relocation task. In Section V we outline a series of experiments that we conducted in simulation to understand the behaviour of our system with respect to some key system variables. We will discuss the results of our experiments in Section VI and conclude with Section VII, where we will make some closing remarks and will outline some of the future work with which we will be following up the work detailed in this paper.

II. BIOLOGICAL COLLECTIVE DECISION MAKING

The natural world is rife with examples of collective systems. In particular, social insects have been the inspiration for several collective robotic tasks including collective construction [9] and collective transport [7]. A task common to most social insects is that of relocating the entire colony. For an entire colony to relocate successfully,
all of the members of the colony must agree on a new nest site. If an agreement is not reached, the colony may split amongst several nest sites, spreading itself too thin. This is no small feat considering that many social insect colonies have thousands of members or more. In [2], Britton et al. present an analysis of how honey bees search for candidate nest sites and then choose the best one. The process involves a heuristic search by the bees followed by a period of advertising the choices to the rest of the colony. Britton et al. liken the behaviour of the bees to that of political parties. Just like in a political arena, the bees campaign for the best nest site that they have seen and try to recruit other bees to it. Effectively, the bees “vote” their approval for a the various candidate nest sites and the most popular one gets utilized.

The ant species *Leptothorax albipennis* often is confronted with the relocation task. These ants inhabit small cracks in rocks and live in relatively small colonies of less than 300 ants [4]. From time to time, their nests, which consist of small, circular regions surrounded by a simple wall, will become damaged and new a nest site will have to be found.

When an *L. albipennis* nest is damaged, some of the inhabitants act as scouts and search for potential new nest sites. When a scout finds a potential nest site, it is able to evaluate its quality [8]. The ant then returns to the original nest, possibly to recruit other ants to inspect the potential nest site that it has found. Whether an ant will try to recruit other ants to inspect a site depends on the quality of the site. The higher the quality of the site, the more likely the ant will recruit another ant to inspect it. While an ant is making up its mind whether to recruit or not, it itself might be recruited to inspect a different site.

Recruitment effectively involves one ant “asking” another to follow it to a site [4]. The ant then leads its recruit to the candidate site. The ants are able to find their way back and forth between a site and their nest by recognizing visual cues and perhaps as well pheromones that they might have deposited on prior trips between the two locations [4]. Because recruit ants follow and are not carried, they are better able to learn the path between the nest and the candidate site. Upon reaching a site, the recruit evaluates it and both return to the original nest where the process repeats.

While evaluating a site, an ant will conduct a head count where it estimates the number of other ants present at its site and computes whether the number of ants at the site was above some *quorum threshold* [10]. Once an ant believes that its site has reached quorum, instead of leading ants to the site, it instead carries them to the site, which the carried ant will adopt as its new nest. Because the likelihood of an ant leading another ant to a particular site increases with the site’s perceived quality, sites of higher quality tend to gain recruiters faster than the poorer ones.

The best site of the ones discovered tends to reach quorum and induces the high speed carrying of ants to it first. Thus the colony effectively makes the decision to move to the best site known to it through this recruitment strategy.

### III. A Biologically Inspired Decision Making Algorithm for Collective Robotic Systems

In this section we present our algorithm for collective decision making. This algorithm follows from the nest site selection mechanism of the ant *L. albipennis* as presented in the previous section. Many applications of collective robotic systems are strictly cooperative, meaning that they require the effort of more than one robot to complete. If a collective robotic system is to preserve its capability, it must be able to make decisions without fracturing itself into multiple smaller systems.

Refer to the flow chart in Figure 1 for a depiction of our decision making scheme. Like the ants discussed in

---

1It is important to note that the ants do not count in the traditional sense. Rather, they likely use encounter rates at a candidate nest site which serves as an estimate of the number of other ants there[10].
the previous section, our robots first search for solutions to the problem confronting them, and then iteratively estimate the popularity of their solution and recruit others to it, who adopt the solution as their own and proceed to recruit others to it. Once a robot believes that its solution has become popular enough\(^2\), it stops recruiting other recruiters to it and instead recruits robots to implement it. Once a robot trying to recruit implementers can no longer find any robots not implementing a solution to the problem confronting the system, it begins to implement its solution, too.

Note from the figure that, before a robot searches for another robot to recruit to its solution, it might instead go idle. Whether or not a robot recruits or go idle depends on how good the robot perceives its solution to be. The better it perceives its solution, the higher the likelihood that it will try to recruit others. Thus the recruiting force behind a solution will tend to be greater the better that solution is. As a result, the best solution to the problem will tend to gain support faster than any other solutions, and thus will reach threshold first and be selected by the system as a whole. At any time, a robot might be recruited away from its solution to the problem by another robot recruiting for its own solution.

In some ways, our decision making algorithm bears resemblance to ACO, or “Ant Colony Optimization” as described by Dorigo et al. in [3]. There are important differences between their and our approaches to distributed computation. In ACO, software “ants” modify their environment by depositing “pheromone” as they wander about a problem’s solution space. As other ants respond to pheromone trails, positive feedback via stigmergy leads to the colony of ants selecting the best solution. In our system, there is no environmental (artificial or otherwise) modification, nor is there any sort of pheromone. Robots recruit each other directly and assess the popularity of a solution directly. Dispensing with pheromones makes our approach more applicable to real robots, where chemical cues would be impractical.

IV. AN APPLICATION OF OUR ALGORITHM TO THE COLLECTIVE RELOCATION TASK

The collective relocation task is a variation on the collective search problem. A system undertaking the collective relocation task attempts to find a location in its environment that best satisfies some criteria and then moves the entire collective system to that location. The L. albipennis ants of [4], [8] and [10] and the honey bees of [2] undertake this task, as do many other social insect species. In the case of the insects, the system is trying to find a new home and to move to. There are many other applications of collective relocation: moving to a position to gain a better vantage point on a target, moving to a location that is safer for the system (e.g. further from danger), etc. By changing the value function that the robots use to measure the quality of a potential site that they might find, the type of site that the system will move to can be specified.

\(^2\) A solution becomes popular enough when the number of robots that support it exceeds some preset threshold.

The system described in this section attempts to identify candidate sites in its environment that the system could move to and then moves the system to the best one. In order to keep our results as generally applicable as possible, we assign an abstract quality to the various potential sites that our system will have to decide amongst. We developed our system in simulation using the TeamBots\(^\text{T M} \) simulation environment that was developed at Georgia Tech University [11]. The robots in our system can identify potential sites to move the system to in their environment and can assess their quality by examining a location’s “colour”. Colours in TeamBots are represented by positive integers. The higher a potential site’s colour, the better the potential site’s quality is. The robots used in our system have a limited visual field of view and a limited radio communication range. The ranges of a robot’s radio and vision are the same: if a robot can see another robot, it also can communicate with it. Since a robot only will communicate once it visually has found another, it served no purpose to extend the range of the robots’ radios beyond that of their vision.

The behavioural controllers are the same for all of the robots. The only difference is that some of the robots start in a search state and some robots start in an idle state.

The behaviour of the robots in our system is quite similar to that of the L. albipennis. We will borrow from the terminology used in the papers that discuss this ant. The robots all start out in an initial home region that we will refer to as the nest. Robots that are idle stay near the nest waiting to be told about potential nest sites to inspect. Just like the ants, our robots recruit other robots and then lead them to potential new nest sites, which we will refer to simply as “sites” from here on. Unlike the ants, when our robots believe that quorum for a particular site has been reached, they do not carry their fellow robots to their site. Rather, they tell the other robots where they can find the site and that they should go to it and adopt it as their new nest - the destination for the collective relocation task.

Another difference between how our robots behave and how the ants behave is in how the variable probability of recruiting is implemented. With the ants and indeed, with our general purpose decision making algorithm, robots/ants will attempt to recruit another robot/ant to a site with a certain probability that is a function of the quality of the robot’s/ant’s site. Further, robots can be recruited at almost any time in the general algorithm. In order to make the controllers for our robots in the collective relocation system somewhat simpler, we devised an approximation of this scheme. The robots in our system only can be recruited while in the idle state or in a “delay” state. Robots returning from sites for which quorum has not been met do not enter the recruiting state with some variable probability. Rather, they enter a delay state where they are receptive to being recruited. The better the site that a robot is returning from, the less time it will spend in the delay state and thus the less time it will be available for recruitment. The end result for these two scenarios is the same: Robots championing better sites will spend more of their time recruiting and less of their time receptive to being recruited while robots
championing poorer sites will spend more time receptive to being recruited and less time recruiting. The robots championing poorer sites are more likely to be recruited to champion better sites than vice versa. Following the delay state, all robots enter a recruiting state in which they actively recruit other robots to their site. If a robot believes that quorum for its site has been reached, it will skip the delay state and enter the recruiting state directly.

V. EXPERIMENTAL SETUP

Using the system described in the last section, we conducted a series of experiments to verify the system’s performance characteristics. In particular, we were interested in how the performance of the system would change with the population of the system, the quorum threshold \( q_0 \) used by the robots and the number of robots initially in the search state. The quorum threshold \( q_0 \), the number of robots that need to be seen championing a particular site before a robot will detect that quorum for that site has been reached, is a variable that we believed would be of particular importance to our system’s performance. If this threshold was set too low, multiple sites could reach quorum and the system might split amongst them. If it was set too high, robots may become too "fussy" and never see enough other robots at a site in order to commit to it.

A screen shot of our system in action can be seen in Figure 2. The environment for our simulations consisted of an enclosed square arena with the robots’ nest in the center. Two sites were located in the environment equidistant from the central nest. One of the sites was mediocre and the other was good. A mediocre site was a site that would be selected over an even worse site, but would be rejected in favour of a good site. This setup is consistent with that of [4].

Forty-eight experimental configurations were investigated with ten trials being conducted for each. The forty-eight configurations varied the system population, the quorum threshold \( q_0 \) and the number of robots that would start in the search state. Populations of four, eight and twelve robots were studied. For each population, we ran trials with the quorum threshold \( q_0 \) equal to 0%, 25%, 50% and 75% of the total system population. Each of these configurations were run with 1, 2, 3 and 4 robots starting in the search state. Trials were allowed to run for a maximum of 200,000 simulated time steps. Robot state transitions were recorded in logs along with where they considered the nest to be and which site they currently were championing.

VI. RESULTS AND DISCUSSION

From the logs of robots’ behaviours recorded during our experiments, we calculated three performance metrics. First, we were able to determine whether the system was able to reach a consensus and unanimously choose one of the two available sites. If a trial ended with all of the robots calling the same site their nest, then the system was deemed to have successfully made a decision. The total number of successful trials for a given experimental configuration divided by the total number of trials conducted with that configuration was defined as the success rate for that configuration. Figure 3 shows how the success rate for the system varied with respect to system population and the value of the quorum threshold, \( q_0 \).

Notice how a definite peak in the success rate occurred at \( q_0 = 50\% \) of the system population for all three of the tested populations. If the quorum threshold is too low, two nests might reach quorum and thus both would be moved to by robots. This situation results in consensus amongst the system’s robots not being reached. With a quorum threshold \( \geq 50\% \), it will not be possible for more than one site to reach quorum. Increasing the quorum beyond 50% makes it less likely for a single robot to detect that quorum has been reached for its site. Most if not all of the trial failures for \( q_0 \geq 50\% \) are due to the trial reaching its time limit before all of the robots could detect that quorum had been reached.

Next, we determined which of those trials that ended in success also saw the system choosing the good site over the mediocre site. We defined the percentage of the successful trials in which a system chose the better of the two sites to be the system’s ability to choose the better site. Figure 4 plots the system’s ability to choose the better site for the three robot populations versus \( q_0 \).

For all three system populations, the system’s ability to
The true success rate of the system is the rate at which the best site is chosen. This is calculated based on the number of trials in which a unanimous decision was made to move to the best site available. Our system’s true success rate is plotted in Figure 5 for the three system populations versus the quorum threshold. This graph appears similar to Figure 3, except that the peaks in the graph are narrower.

All of the data presented thus far has been from the trials in which four robots began in the search state. In our system, robots that search for sites do not exit the search state until they have found a site. Thus the number of robots that start in the search state represents the number of site reports that will be brought to the system. In the experiments that we conducted, there were only two sites to choose from.

Since the two sites were equidistant from the initial nest, the probability that a robot would find a particular site (say, the good one…) was 50%. With only one robot, the system would at best be able to move to the only site that was found; there would be no other site to compare the found one to. With two robots, there still would be a 50% chance that only one of the sites would be found. Carrying this logic further, three robots would find both sites 87.75% of the time, and four robots would find both sites 93.75% of the time. However, with robots never giving up the search until they find a site, increasing the number of robots that conduct a site search also increases the probability that one of the searchers might not find a site before time runs out. In our experiments, increasing the number of robots that started in the search state tended to increase the amount of time that it took for the system to come to a decision, but also increased the system’s true success rate. The number of robots that should start in the search state and search for solutions to a problem that is confronting the system

\[ q_0 = 50\% \]

Choose the better site increased with \( q_0 \) up to a quorum threshold of 50%. In the case of the 12-robot system, none of the trials with \( q_0 = 75\% \) ended in success (see Figure 4). Also note that once \( q_0 \) reached 50%, there was no further improvement in the system’s ability to choose the better site. This suggests that a quorum threshold of 50% is sufficient to maximize a system’s ability to make the best choice.

Combining the data from Figures 3 and 4 gives us what we call the true success rate of the system. The true success rate of the system is the rate at which the system is able to make a unanimous decision to move to the best site available. Our system’s true success rate is plotted in Figure 5 for the three system populations versus the quorum threshold. This graph appears similar to Figure 3, except that the peaks in the graph are narrower.

\[ q_0 = 50\% \]

Since the robots search for sites using a random wander behaviour, it is possible for a robot to never find a site to report back about before a trial’s time limit is reached. Some of our experiments did fail because of searchers that never found a site, but this phenomenon was relatively rare.
likely is closely related to the type of problem and how many potential solutions are likely to be found.

VII. Conclusion and Future Work

Collective robotic systems often are used to accomplish tasks that require multiple robots to solve them. In a realistic environment, such a system likely would be confronted with problems that would have more than one possible solution. When making a decision about which solution to a problem to pursue, it is very important for the robots that make up a collective system that they unanimously choose one particular solution so that they can remain a single cohesive group instead of fracturing into multiple smaller groups. Were a collective robotic system to split into several smaller systems, much of the original system’s functionality could be lost.

In this paper, we have presented a new approach to decision making for collective robotic systems that puts a strong emphasis on preventing splits within the system. Our collective decision making algorithm was based on the collective site selection behavior observed in the ant Leptothorax albipennis in [4], [8] and [10]. Like the ants, our robots conduct a search of the solution space of the problem confronting them. Individual robots, upon finding solutions to the problem, recruit other robots to champion their particular solution. The robots use an estimate of the number of other robots who share their particular solution to determine whether enough of a consensus exists for a system level decision to be made. When the number of robots supporting a given solution reaches a preset quorum threshold, a system level decision is made to implement that solution to the problem. The mechanism by which robots recruit each other to the various solutions promotes the better solutions to a problem over the poorer ones. As a result, the best solution will tend to reach quorum first and thus be pursued by the system as a whole.

We applied this decision making scheme to the collective relocation task, a task in which a collective robotic system searches for locations to move to and then chooses the best one. Our system was able to make the best choice consistently regardless of the system’s population. Varying a particular variable, the preset quorum threshold, significantly affected our system’s performance in terms of the three metrics that we defined. All three system populations were able to make unanimous decisions to move to the better site in 80% or more of the experimental trials conducted when the quorum threshold was set to 50% of the system’s population. Our results suggest that our decision making algorithm is highly scalable with respect to system population. The number of robots that conduct the initial search for new nest sites affected the performance of our system, too. More robots searching for sites increased the likelihood that the better site would be presented to the system during the actual decision making phase. However, having too many robots take part in the search would significantly increase the time that the search would take, increasing the likelihood that the system would be unable to complete its decision before the time limit would elapse. It was concluded that the optimal number of robots to take part in the search phase likely would be related to the problem being solved by the system.

Thus far we have concentrated our efforts on understanding the relationship between the quorum threshold of a system and the ability of a system make good, system level decisions. One of the problems that we encountered was that robots conducting searches for potential nest sites might not finish their searches for potential nest sites before the time limit for a trial elapsed. Other forms of stagnation were present, too. The ants from which our algorithm was derived utilize several simple yet effective stagnation prevention mechanisms. We would like to investigate how the inclusion of such mechanisms in our system might improve its performance.

Acknowledgment

The authors would like to thank Dr. Stephen Pratt for consulting with us on the various biological aspects of this research.

References