IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 4, Ver. I (July – Aug. 2015), PP 82-87
www.iosrjournals.org

Cooperative Navigation in Sparsely Populated Swarms

Thyagaraj T[1], Venkat Chinta[2], Karun S[3], Sumukh U[4]
[1234]*(Department of ECE, BMS Institute of Technology and Management, India)*

Abstract: *The current state of the art techniques for robot swarm navigation use communication assisted local interactions to achieve a cooperative solution to the problem. While this solution provides optimum, near-shortest paths in swarms of considerable population, it fails to scale well with reduction in population. We propose a solution to this shortcoming by incorporating cardinality in the swarm and utilizing it to impose constraints on the robots' possible movements. We have tested our solution using time and population as parameters of performance. We compare the performance with existing communication assisted algorithm to accentuate the improvement in performance and outline possible future work in the area.*

Keywords: *Cardinality, Communication assisted navigation, Cooperative control, Robot swarm navigation, Sparsely populated swarms*

I. Introduction

Swarm robotics is an emerging technology that is being researched extensively [1]. The basic premise of swarm robotics is to use a large number of mostly simple robots that can coordinate amongst themselves to accomplish complex tasks [2]. Solving complex tasks requires demonstration of collective behaviour by the swarm wherein the robots are able to work cooperatively through local interactions between individual members of the swarm [3]. As the robots are designed to be relatively simple, they do not have global positioning capabilities. Thus navigation within the swarm remains an important question to be solved. Navigation in a swarm can be defined as the ability of a robot within the swarm to be able to move from its present location to another, usually a location of interest. To achieve navigation, a robot requires spatial context i.e., it needs to know where itself is with respect to the swarm and where the location of interest is with respect to the swarm.

One of the best examples of swarms in Nature is ant swarms [4]. Most of the current work in this area is inspired by the social behaviour of ants, which rely on stigmergic communication through pheromone trails due to their. All ants moving between the colony's nest and a food source leave pheromone in thier wake, which guides others ants to the food source [5]. Current navigation algorithms use stigmergy through what is termed as a *virtual pheromone* approach. In this approach electronic communication means are used to lay pheromone trails i.e., the robots interact with each other to relay navigational information throughout the swarm. Two of the more popular techniques are described below.

1.1 Previous Work

One of the earlier solutions to swarm navigation using the assistance of communication is detailed in [6]. In this paper, two algorithms are outlined: *cardinality algorithm* and *virtual pheromone algorithm*. In both the algorithms the robots can choose to be stationary or mobile. In the virtual pheromone algorithm the stationary robots act as beacons on which a virtual pheromone distribution could be stored. The value stored is inversely proportional to the proximity to the path. A mobile robot then receives the transmitted pheromone distribution by the beacon robots based on which their motion path can be decided. In the cardinality algorithm, instead of transmitting real values the beacons transmit an integer termed cardinality which translates to *count*. Beacons standing nearest to the nest or target location transmit a 1, beacons that can hear the 1, transmit a 2 and so on. It was shown that both the algorithms enable swarm to navigate better than in the case where navigation is achieved through random motion by the robots. However, the limitation of this approach is that the robots that act as beacons remain stationary and do not perform any tasks of their own. While this may not impact theperformance of swarms of large populations, this would tie up a valuable resource in sparsely populated swarms.

This limitation is overcome in the solution proposed in [7]. Here navigation information is relayed throughout the swarm using network communication. All the robots maintain a table of navigation information about known target robots in the environment, where a target robot is assumed to have found an object of interest. This navigation information contains the effective distance to the target robot and the relative freshness of the information. Each robot periodically broadcasts its navigation table in all directions while performing its own task. Any robot that receives this information in turn updates its own table to reflect its distance to the target, based on its relative distance from the sender. This relative distance is calculated with the help of an infra-red range and bearing system(IrRB). This system provides spatial context to the robots from the messages received i.e., relative position of the sender in terms of distance and angle. A robot that needs to navigate to the target robot moves to a neighbour that is transmitting the least effective distance to the target robot. If and when it finds another robot transmitting lesser distance to target, it changes its path and moves to the new robot, thereby achieving an optimum path to the target. The limitation of this solution is that it is dependent upon the population of the swarm to be able to provide navigation to the target. Given the relatively small communication range of the robots, robots in sparsely populated swarms may not be able to form a continuous network.

Our solution aims to integrate the advantages of both the above mentioned algorithms to realize navigation in sparsely populated swarms. Specifically, we use the network communication model to relay navigation information throughout the swarm and infra-red range and bearing system to provide spatial context to the robots. We utilize the concept of cardinality to ensure that the communication network formed by the robots in the swarm does not disintegrate once formed. We achieve this by assigning cardinality to the robots akin to the cardinality algorithm. We then use this cardinality as a hierarchy in the swarm: motion of robots with lower cardinality is constrained in a manner in which they cannot move out of range of robots with higher cardinality. Our algorithm is discussed in further detail in Section II.

1.2 Robot model and Simulator

Robot model **-** our robot model is a two-wheeled moving platform that includes electronics for communication and sensors for obstacle detection. The robots communicate using an infra-red range and bearing system. The design of such a system is detailed in [8].

Simulator **-** To test the algorithms, we have utilized the Webots EDU 8.0.5, a mobile robot simulation software [9]. It enables the user to create 3D worlds in which robots and other passive or active objects exist, and all objects have physical properties such as mass, friction coefficients etc. It also provides the user with the ability to add sensors and electronics such as motors, communication modules, obstacle detectors etc. The behaviour of all robots can be programmed individually. Webots contains a large repository of models of various robot platforms. We have chosen to use the e-puck robot model for its simplicity and suitability to our application.

II. Algorithm Description

As explained earlier, our algorithm modifies the communication assisted navigation algorithm by incorporating cardinality in the swarm. For the purposes of explanation, we shall define three types of robots.

A T-Robot is said to have found an object of interest that needs to be serviced. A D-Robot is a robot that is assumed to be able to provide the required service and hence is the robot that must navigate to the T-Robot. All other robots in the swarm are termed F-Robots and are tasked to forage for other objects of interest. They do not stop performing this task to help D-Robot navigate to T-Robot. Every message transmitted by any robot contains robot type, cardinality, freshness of information and effective distance to T-Robot.

T-Robot remains stationary and continuously transmits a message with a cardinality of 0, freshness and a distance of 0 units.

F-Robots perform random motion in search of other objects of interest. At the same time they also check for communication from other F-Robots and T-Robot. At the start, until F-Robots receive any communication, they do not transmit any information. Any F-Robot that comes into communication range with T-Robot, receives T-Robot's message and calculates its distance to T-Robot based on the received signal strength, with the help of the range and bearing system. The freshness is copied as received and the distance to T-Robot is transmitted along with a cardinality of 1. F-Robots that receive messages from other F-Robots and F-Robots only, compare the received messages for cardinality, freshness and distance to T-Robot. F-Robot ignores messages with comparatively lower freshness. If a fresh message is received from robots with lower cardinality, its own cardinality is adjusted to one more than the received cardinality. From the messages with comparable freshness, its probable distance to T-Robot is determined by adding the distance received in the messages with its distance to the senders. The least value among them is then fixed as its effective distance to T-Robot. This distance, along with copied freshness and adjusted cardinality is then transmitted by the F-Robot. If F-Robots do not receive any communication or any fresh messages after having received some before, they simply re-transmit their previously transmitted message.

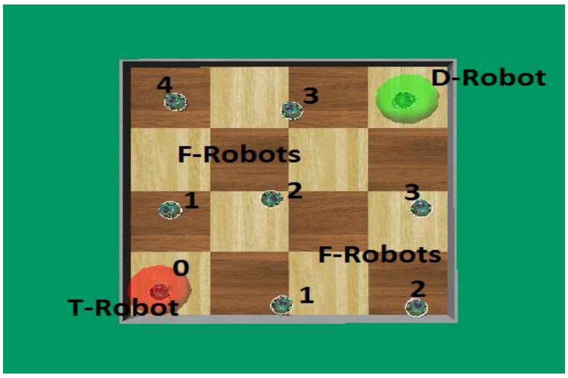

Figure 1: Illustration shows F-Robots with their respective cardinalities.

Every time an F-Robot checks for received communication, it also checks for the lowest cardinality received and stores the angle from which the message with the lowest cardinality was received. It compares this cardinality to the lowest cardinality received in the previous messages. If the same or better cardinality is not present in the current message it is inferred that the robot has moved in a manner that is undesirable. Thus it performs a corrective motion to get back in range with a robot with better cardinality, by moving in the direction of the stored angle.

D-Robot, similar to F-Robot, compares the received messages for freshness, cardinality and distance to T-Robot. Using these comparisons D-Robot decides its path with highest freshness, least cardinality and least effective distance to T-Robot, in that order.

tRobot	dRobot	fRobot
#1: set cardinality to 0 **#2:** set distance to 0 **#3:** increase freshness **#4:** transmit message **#5:** wait for a fixed time **#6:** go to #3	**#1:** check for communication **#2:** **if** T-Robot found - stop; **else** determine messages with lowest cardinality and highest freshness **#3:** add received distance with distance from sender and determine the sender with least effective distance to T-Robot **#4:** move in the direction of that sender **#5:** go to #1	**#1:** check for communication **#2:** determine messages with lowest cardinality and highest freshness **#3:** **if** lower cardinality neighbour lost - perform corrective motion, go to #1; **else** set own cardinality and copy freshness **#4:** add received distance with distance from sender and determine its least effective distance to T-Robot **#5:** transmit message **#6:** move randomly **#7:** go to #1

Figure 2: Concise algorithm for all the robots.

III. Performance Parameters

In this section, we discuss the experimental setup and performance parameters used to compare the algorithms.

3.1 Experimental Setup

The experimental setup depends upon many factors such as arena size, population i.e., number of robots comprising the swarm, communication range, robot distribution, origin and target location.

Arena size - an optimum arena size can be calculated using robot radius and communication range, after determining the maximum swarm population that we would like to study. For our study, we have chosen twelve robots as the maximum population for simplicity. This number would also keep the expenditure low for a potential comparison study using implementation of the algorithm on hardware, in the future. We have chosen a square arena for ease of robot distribution.

Population - to highlight the performance of navigation algorithms in sparsely populated swarms we keep the arena size fixed and reduce the population of the swarm by one in each case until we reach a case for which a solution is not practically feasible.

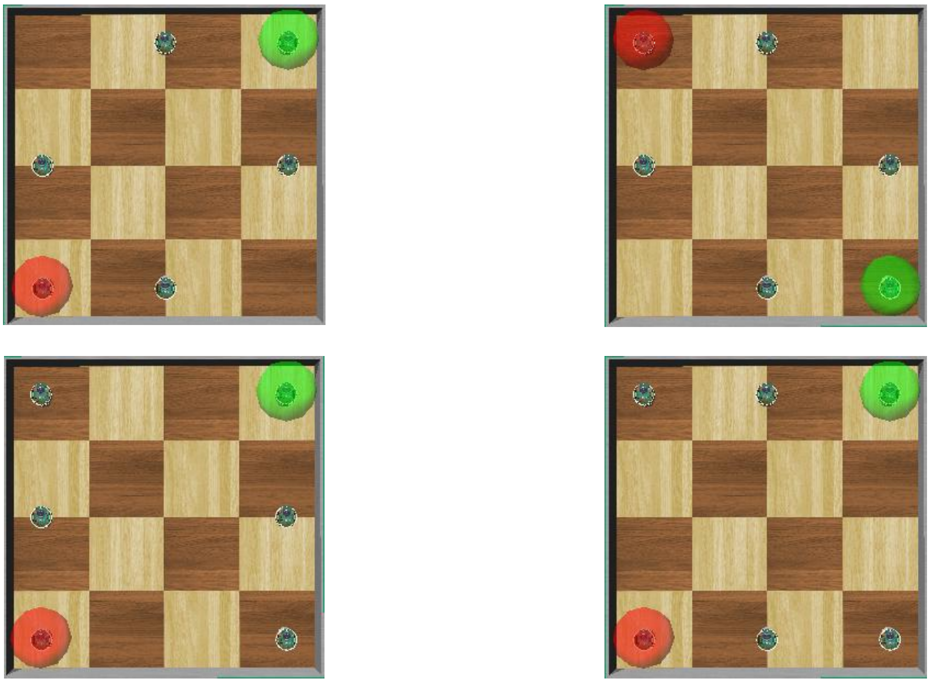

Figure 3: Possible arrangements of robots for a swarm of population of six robots.

Robot distribution - most studies in swarm navigation use uniform distribution of robots within the arena. As the focus of our work is to study the performance of navigation algorithms in sparsely populated swarms, it does not make sense to utilize uniform distribution as it would reduce the effective distance between origin and target as the population decreases, for a fixed arena size. Instead, we have fixed the location of the D-Robot and T-Robot and distributed the F-Robots uniformly throughout the arena.

Origin and Target location - the performance of the algorithms in terms of time taken for the robot to reach target location largely depends upon the distance between its starting location, i.e., origin, and the target location. To reduce its impact on our analysis, we have chosen the worst possible origin and target locations for all swarm populations. We have also iterated through all the possible arrangements of F-Robots in each case to reduce probabilistic error. Fig. 3 shows the possible arrangements for a swarm of population of six robots.

Communication range - we have set a communication range of 40 cm for the robot as commercially available IR emitters that we intend to use in our hardware implementation operate up to distances of 40 cm with usable signal to noise ratio.

3.2 Performance Parameters

We measure and compare the performance of the algorithms using time and population as the parameters.

Time - the total amount of time taken for the D-Robot to find a T-Robot is measured for all robot arrangements in each case, where a case refers to a swarm of a certain population.

Population - the population of the swarm is initially set to nine robots and reduced by one for each subsequent case.

Performance comparison - both the algorithms are run for every robot arrangement in each case and the time to completion is recorded. The time to completion is averaged over all the arrangements for each case and plotted against the population of the swarm. As an example, Table I shows time recorded for all iterations for a swarm of population of six robots and the average time to completion.

Table I: Time to completion in a swarm of population of six robots

Configuration	Time Run A	Time Run B
A	198.27	198.46
B	197.84	198.05
C	198.13	198.57
D	198.28	198.61

IV. Results

4.1 Performance of Algorithms over Time

The communication assisted algorithm and our algorithm were both run for swarms of various populations: starting from nine robots and gradually reduced until a solution was not feasible any further. In each case, the algorithms were run for all possible robot distributions and the time taken was recorded and averaged over all the iterations as explained in Section III. The results are displayed in Fig. 4.

As seen in Fig. 4, the performance of both algorithms is comparable when the swarm population is sufficiently high (9 to 6)and the robots are able to cover a majority of the arena area with their communication range. As the population reduces, our algorithm starts to perform considerably better with a gain in performance of over 4.8per cent.

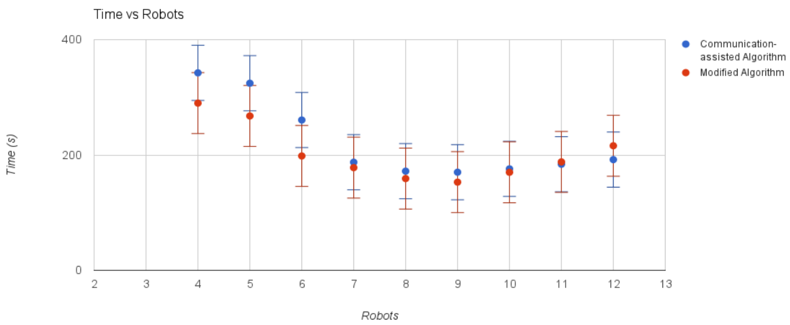

Figure 4: Comparison of performance of algorithms over time for different populations

4.2 Congestion

All navigation algorithms mentioned in this paper are affected by congestion when the robots in the swarm are packed to a high density [6][7]. This is due to the fact that the robots tend to run into each other and cause *jams*. The results for populations higher than nine show that congestion affects our algorithm much more adversely than the communication assisted algorithm. This is due to the constraints set in our algorithm that force the robots to remain closer to each other to maintain a continuous network. We can infer from this that while the modified algorithm puts up a strong performance in sparsely populated swarms, it is ill-suited for swarms of large sizes.

4.3 Structure

Some of the main features of the communication assisted navigation algorithm are the self-organization and emergence properties that it induces into the swarm [7]. As defined by the authors of the work, self-organization means that the organization of the swarm occurs from within the system and emergence refers to the fact that the organization comes about in a decentralized way. We have strived to maintain these properties in the modified algorithm. Although the use of cardinality to form a hierarchy in the swarm may appear as imposing structure, this hierarchy and apparent structure still comes about as a result of interactions between individual robots i.e., in a decentralized, self-organizing manner.

V. Conclusion

Through this paper we have presented a modification to the communication assisted navigation algorithm borrowing a feature from the cardinality algorithm. We have shown that our modification enables the algorithm to perform better in a sparsely populated swarm, using population and time as the performance parameters.

Our modification shows that while it is ideal for a swarm to be self-emergent in nature, it is beneficial, in the case of navigation in sparsely populated swarms, to impose certain constraints on the robots' behaviour to improve performance. We have also studied that the same constraints that are beneficial for sparsely populated swarms, also lead to an early onset of congestion compared to the unmodified algorithm.

In the future, we aim to eliminate this undesirable side-effect of imposing constraints on the swarm by employing dynamic switching between the two algorithms: modified algorithm for sparsely populated swarms and unmodified algorithm for sufficiently populated swarms. We also plan to parallelly develop an indigenous robot platform suitable for swarming applications.

References

[1]. Xing P Guo, Editor, Robotics Research Trends, Nova, 2008.

[2]. M. Dorigo and E. Sahin, Guest editorial: Swarm robotics, Autonomous Robotics, vol. 17, no. 2-3, 2004.

[3]. T. De Wolf and T. Holvoet, Emergence vs. self-organisation: Different concepts but promising when combined, Engineering Self-Organising Systems, pages 1-15, Springer Berlin Heidelberg, 2005.

[4]. S. Camazine, J.-L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz, E. Bonabeau, Self-Organization in Biological Systems, PrincetonStudies in Complexity. Princeton University Press, Princeton, NJ, 2001.

[5]. R. Sharpe, B. Webb, Simulated and situated models of chemical trail following in ants, Proceedings of the International Conference on Simulation of Adaptive Behavior, pages 195–204, 1998.

[6]. Hoff, Nicholas R., et al. Two foraging algorithms for robot swarms using only local communication, Robotics and Biomimetics (ROBIO), 2010 IEEE International Conference on. IEEE, 2010.

[7]. F. Ducatelle, G. A. D. Caro, C. Pinciroli, F. Mondada, L. Gambardella, Communication Assisted Navigation in Robotic Swarms: Self-organization and Cooperation, IEEE International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, pages 4981-4988, September 25-30, 2011.

[8]. J. Roberts, T. Stirling, J. Zufferey, and D. Floreano, 2.5d infrared range and bearing system for collective robotics, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2009.

[9]. Michel, Olvier, Webots: a powerful realistic mobile robots simulator, Proceedings of the Second International Workshop on RoboCup, 1998.