Blinkered LOST: Restricting Sensor Field of View Can Improve Scalability in Emergent Multi-Robot Trail Following

Seyed Abbas Sadat, Richard T. Vaughan
Autonomy Lab, Simon Fraser University
{sas21, vaughan}@sfu.ca

Abstract— **We consider the classical task of transporting resources from source to home by a group of autonomous robots. The robots use ant-like trail following to navigate between home and source. This paper studies the effect on global performance of changing the field of view of each robot's trail-following sensor. It is shown that, under certain conditions, a narrow field of view can improve system performance. We argue that the benefit is obtained by selectively degrading the individuals' trail-following ability so that more work space is exploited in parallel, thus decreasing mutual spatial interference. This "worse-is-better" idea may be applicable to other large-scale multi-robot systems.**

I. Introduction

We consider the classical *resource transportation* task, in which a team of robots works to transport resources in an initially unmapped environment. Robots start from a home position and search for a supply of resources. On reaching the source, they receive a unit of resource and must return home with it, then return to fetch more resource repeatedly for the length of a trial. Achieving this task reliably with robots will meet a real-world need. It is a canonical multi-robot task since the work is inherently parallelizable. The critical factor limiting scalability is mutual spatial interference between robots.

Our earlier work [1], [2] examined an implementation of ant-inspired trail following that is suitable for imperfectly-localized mobile robots. In our "localization-space trails" (LOST) algorithm, robots generate and share trail data structures composed of waypoints specified by reference to task-level features that are shared by all robots. The trails are continuously refined online, and maintain the ant-algorithm property [3] of converging to near-optimal paths from source to home.

An attractive feature of LOST and other ant-algorithm methods is that it is simple and natural to use travel-time as the distance function to be optimized, so that the system can discover paths that may be longer in space but shorter in time since they spread robots out to minimize mutual spatial interference between robots. As the population size increases, such interference eventually dominates the travel cost. However, since ant algorithms (including LOST) tend to converge to a single "best" trail, in large populations this trail can become badly congested and performance reduced. To address this, we seek to perturb the ant algorithm such that it does not converge to a single trail when the interference is high, and instead to maintain multiple trails that spread the robots in space and time.

The contribution of this paper is to examine the effect of modulating the field of view (FOV) of the robots' trail-detecting sensor. We show that global throughput is a function of robot FOV, where narrower FOVs perform better in large populations. Experimental evidence suggests that the narrow FOV causes multiple trails to be maintained, so that the system can support larger population sizes before saturating due to interference. This simple means of controlling congestion does not require any change in the original trail following algorithm, or any additional sensing.

II. Related Work

Various different robot implementations of ant-like trail following have been presented. Real chemical marks were first used to produce true stigmergic trail-following in [4]. Also recently, Fujisawa et al. [5] carried on a study out of communication in a swarm of robots using pheromone and proposed a behavior algorithm for robots to search for prey and attract other robots. They used ethanol as pheromone in their real robot experiments. The challenge of chemical and sensor engineering makes these methods often impractical. A more parsimonious method was invented by Payton [6] where virtual pheromone trails are implemented by directional infra-red messages transmitted from robot to robot. Robots echo received messages, incrementing a contained hop-count which is used to estimate the distance to the message source. In both chemical and IR-mediated methods, the local "gradient" is sensed directly from the environment. If robots are mutually localized, virtual trails can be created from global waypoints, which are communicated by wireless network. We showed that this scheme can be robust to large zero-mean localization error [1], and admits a relaxed and practical definition of mutual localization [2].

The diminishing-to-negative-returns effect of increasing the number of robots on performance has been studied in related contexts. In a mathematical model of robot foraging [7], it was shown that adding more robots to the system improved the group performance while decreasing individual robot's performance. Based on that model, an optimal group size was found that maximizes the group performance. Explicit anti-interference strategies are studied in real robots in [8], to increase performance in the transportation task. Congestion control in a dense multi-robot system is studied in [9], where asymmetries that resolve conflicts are introduced by modifying either the environment or the robot controllers. In contrast, the method presented here is symmetric and can be considered complementary.

III. Localization-Space Trails (LOST) review

This section briefly reviews the generalized trail-following method formulated in [2].

LOST generates trails between the locations of *Events*. An Event is defined as a task-relevant occurrence that may happen to any member of the team, and is locally but reliably perceived. For example, in our transportation task the relevant Events would be ‘pick-up-resource’ and ‘drop-resource’. A robot must be able to recognize these events in order to switch between resource-seeking behavior and home-seeking behavior. When an Event occurs to a robot, its current pose in localization space is recorded to create an [Event,Pose] tuple called a *Place*. A robot can then express information about the world relative to the Places it has seen. Other robots that have position estimates for the same Events can interpret the coordinates in their own local frame of reference. Thus robots are mutually localized by the shared experience of the common task, rather than conventional global localization in some arbitrary coordinate system.

The purpose of LOST is to guide the robot to a Place currently of interest: the goal. The algorithm provides the robot controller with two pieces of information; (i) the *heading-hint* that is the local direction in which to travel to reach the goal; (ii) the *distance-hint* that is the estimated cost (usually in time) to reach the goal. These hints are extracted by examining a set of waypoints called *Crumbs* which are poses specified relative to a Place. The current set of Crumbs specified relative to a particular Place is a *Trail* to that place. A Crumb is a tuple $C = [P_c, L_c, d_c, t_c]$ containing the name of the Place P_c to which it refers, a localization space pose L_c, an estimate d_c of the distance (in some distance function) from L_c to P_c, and the time t_c when the Crumb was created.

Each robot maintains an initially empty temporary trail. Every S seconds, a robot inserts a new crumb to the temporary trail. The crumb contains the current location of the robot, the name of the most recent Event experienced by that robot, the distance from the last event, and the current time. When another event occurs to the robot (e.g. when a robot drops off its cargo), the temporary trail is broadcast to all robots, including itself, then deleted. A new temporary trail is then created for the recent Event.

Besides the temporary trail, each robot maintains a trail for each different Event it has learned about from the network. When a broadcast trail is received, the crumb poses are transformed into the local frame of reference by the rigid body transform defined by comparing the local and received poses of the trail’s Place. The transformed crumbs are added to the local trail for this Place. All trails are periodically scanned and any Crumb with timestamp older than age threshold a seconds is discarded. Thus the trail is updated dynamically, and out-of-date information is expired. The dynamic response of the trail to changing environments is a function of a.

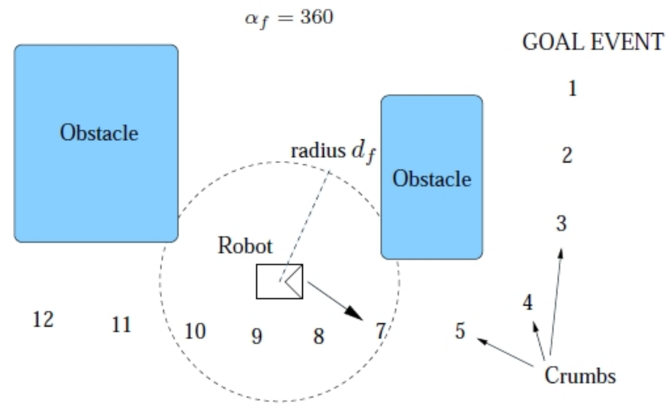


Fig. 1. Sketch of the LOST algorithm, showing a trail of Crumbs with decreasing distance values leading to a goal Event. The robot moves towards the crumb within radius d_f that has the lowest distance estimate.

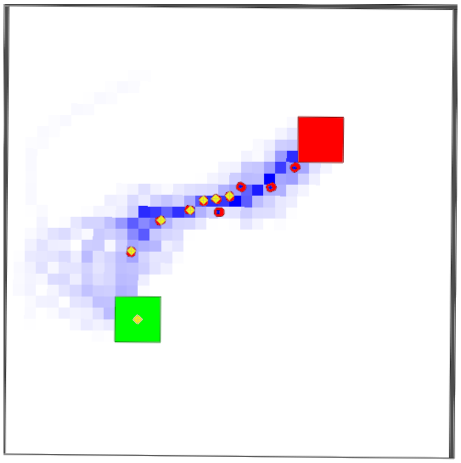
Suppose a robot at pose L_r has Place P_g as its goal, such as $Event(P_g)$ = ‘drop-resource-at-home’. The robot searches the set of Crumbs with Place = P_g to find the set of crumbs that lie within its *field of view*, i.e. within radius d_f of L_r. From this set it finds the crumb C_L with the smallest distance-to-goal d_c. This distance is returned as the distance-hint. The heading-hint is the angle from the robot’s pose L_r to $L_c = Pose(C_L)$. Figure 1 shows the robot’s field of view which is a circle about the robots current location with radius d_f.

If the robot moves in the direction of the heading hint and repeats this process, it will encounter crumbs with decreasing distance to goal values, and eventually arrive at P_g.

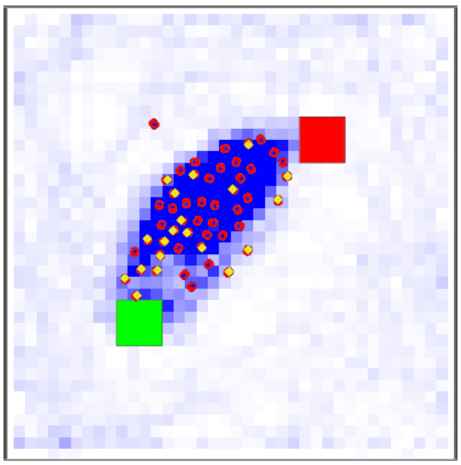
The robot will take the shortest route so far discovered from that location. By following the Crumbs dropped by the whole population, each robot benefits from the others’ exploration; robots will find a reasonable route much more quickly than they would alone. The larger the population size, the greater the probability of finding a good route and the more quickly a good route is found.

IV. Trail congestion

We seek to increase the throughput of our transportation system by adding more robots. But every added robot increases the probability of spatial interference, which can reduce global performance. Eventually the marginal value of adding another robot goes below zero. Figure 2 shows this phenomenon in our trail-following robot system implemented in the well known simulator Stage [10]. In Figure 2(a), 10 robots are successfully following a trail between source and sink (large squares). The small/blue squares are a normalized histogram of space occupancy over time: the robots can be seen to be frequenting the same places as they follow a reasonably direct route. Robots are able to navigate past each other using local obstacle avoidance and the throughput is 232 round-trips per hour, which is close to



(a) 10 robots



(b) 50 robots

Fig. 2. Effect of increasing the population in trail-following robots

the ideal of ten times the single robot work rate (18 round-trips per hour). In Figure 2(b) 50 robots work in the same space. The histogram shows that the robots are still mostly working on a single direct trail. The space is now so crowded that local obstacle avoidance breaks down and robots make little progress. They also lose the trail frequently and explore to recover it. The throughput is 368 round-trips per hour, which is much less than 50 times the single robot workrate. The method described below aims to ameliorate this problem.

V. Changing the Field Of View

Using LOST, each robot moves toward the visible crumb with the minimum distance to goal. The more that two robots' fields of view overlap, the higher the probability that they will select the same crumb and thus follow the same trail. Our approach to congestion reduction is to modify the robots' field of view so that trail-following is still achieved, but to increase the probability of different best crumbs being detected. The hypothesis is that this can cause different trails to be followed and thus reinforced and maintained, spreading robots out in the environment to reduce interference while still making progress on the transportation task.

An analogous mechanism may be used in biological systems. For example in the recruitment behavior of honey bee colonies, unemployed foragers will select at random a single bee displaying food source information, while ignoring all the others. Thus individual bees are not fully informed, and may choose to forage sources other than the best available. It has been argued that this has advantages for the colony overall by preventing overconvergence, for example maintaining exploration of the environment as it changes [11].

To selectively hide crumbs from the LOST forager, we simply vary the radius d_f and reception angle α_f of their virtual crumb-detecting sensor. The FOV of the robot always points forward. In all previous work, α_f was effectively $360°$.

VI. Experiment 1

A. Simulation Setup

To test our hypothesis, we ran Stage simulations with the task environment shown in Figure 2. The arena is 20x20m, with robot length 0.45m, and free of obstacles. Robots are Stage's Pioneer 3DX and SICK LMS200 laser rangefinder models. The bottom left (green) square is the source; top right (red) square the sink of resources. In the screenshots, robots (red polygons) are shown with yellow diamonds to indicate they are carrying a unit of resource. Robots start every trial at the same randomly-chosen uniformly distributed positions, do not know the initial location of source and sink locations, and must find them by exploration at the start of the trial. Each trial runs for 30 minutes, and the total number of resources delivered at the end of the trial is our performance metric. 10 trials are performed for each of a range of settings of radius d_f, reception angle α_f, and population size. LOST is deterministic but the local obstacle avoidance and searching is stochastic (for robustness), hence the need for repeated trials.

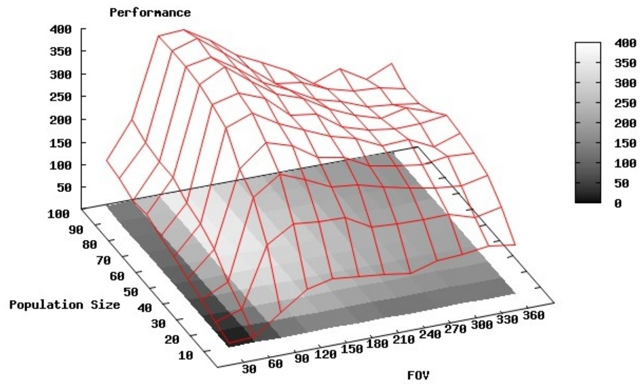
Experiment 1 examined all permutations of $d_f = [1.5, 2.0, 2.5]$ meters, $\alpha_f = [10, 20, \ldots 360]$ degrees, population $P = [10, 20, \ldots 100]$ robots.

B. Results

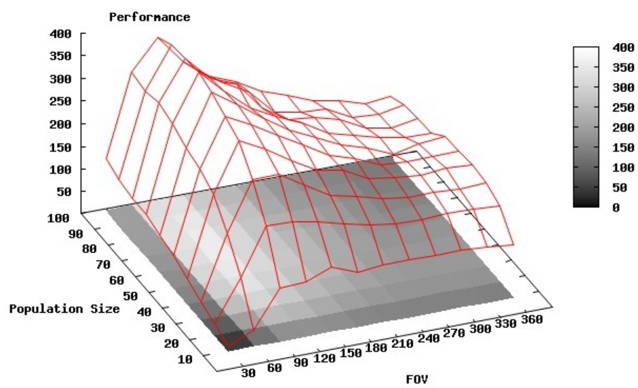
The results of the first experiment are summarized in Figure 3, with mean performance over 10 repeated trials plotted for each $[d_f, \alpha_f, P]$ configuration. Error bars are ommitted for clarity: the variance is $< 12\%$ in 80% of experiments.

The FOV range parameter d_f appears to have relatively little effect on the performance, but the FOV angle α_f appears to have an important effect. The results show that, with a constant d_f, a small team of 10 robots has about the same performance for any α_f above 90 degrees. As the population size increases, the performance is better for smaller α_f, until a lower bound is reached. Performance falls off quickly with α_f below 90 degrees in all cases.

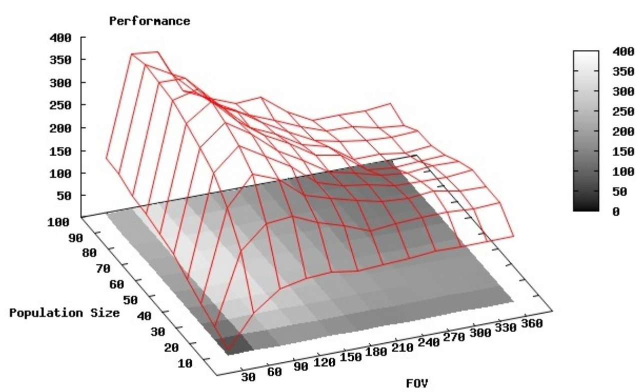
To verify that the performance results are significantly different for different values of α_f, we performed hypothesis testing using a T-test. The P values for the hypothesis that the performance values for α_f=90 and α_f=180 are from



(a) With lookahead distance $d_f = 1.5m$



(b) With lookahead distance $d_f = 2m$



(c) With lookahead distance $d_f = 2.5m$

Fig. 3. Results of Experiment 1, in a world with no obstacles, showing the mean number of resources transported by robots with different populations and field of view configuration. Variance is $< 10\%$ for each point.

the same distribution are given Table I. For all population sizes above 10, the test suggests that the distributions are significantly different, and combined with the higher mean scores for $\alpha_f = 90$, we conclude that $\alpha_f = 90$ performs better than $\alpha_f = 180$ for all populations above 10.

These results show improved performance at large population sizes, suggesting we have achieved a reduction in interference.

To explain how this is happening, see Figure 5, which shows normalized histograms of the last 5 minutes of robot positions at 10, 20 and 30 minutes during one of the trials of Experiment 1 $[d_f = 2, \alpha_f = 90, P = 70]$. Multiple trails between source and sink can be perceived, and the robots spread out in the environment, using these trails. Comparing Figure 6, with a wider FOV $[d_f = 2, \alpha_f = 180, P = 70]$, we see fewer, wider trails and robots closer together.

VII. Experiment 2

To test larger populations and a more challenging search task, we performed a similar experiment in a 4 times larger world (40x40m) containing obstacles. The experimental procedure is identical, testing all permutations of $d_f = 2.0$ meters, $\alpha_f = [10, 20, \ldots 360]$ degrees, population $P = [10, 20, \ldots 100, 120, 160, 200]$ robots.

The results are plotted in Figure 4, showing a similar trend to Experiment 1. Performance is not improved by employing more than 100 robots, but performance is better for smaller values of $\alpha_f > 60$ once the population rises above 50 robots. Hypothesis testing supports this interpretation (Table I right side).

Occupancy histograms are shown in Figures 7 $[d_f = 2, \alpha_f = 90, P = 120]$ and 8 $[d_f = 2, \alpha_f = 180, P = 120]$. The smaller FOV angle produces more trails that are spread around the space. The larger FOV angle produces fewer trails, the shortest of which are heavily used, creating congestion.

A second important feature (we believe) is the lack of congestion at the source and sink locations when the smaller FOV angle is used. Congestion at these areas is very significant since all robots must access them eventually.

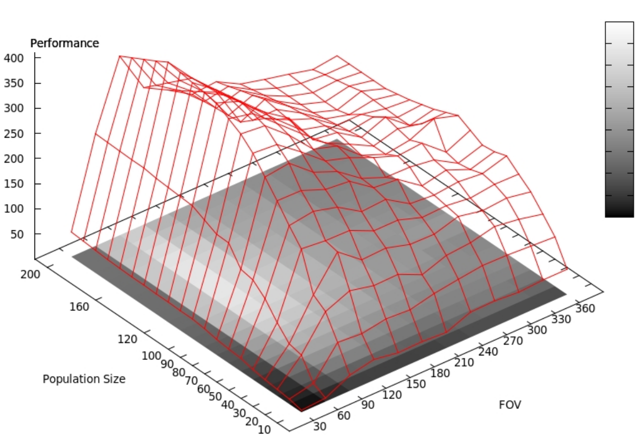


Fig. 4. Results of second experiment

TABLE I

RESULTS OF HYPOTHESIS TESTING, SHOWING THE RESULT OF A T-TEST BETWEEN THE DATASETS GATHERED USING $\alpha_1 = 90, \alpha_2 = 180$ FOR DIFFERENT POPULATION SIZES IN BOTH EXPERIMENTS.

Population Size	Parameter p Experiment 1	Parameter p Experiment 2
10	0.25	0.69
20	< 0.005	0.93
30	< 0.005	0.25
40	< 0.005	0.12
50	< 0.005	< 0.015
60	< 0.005	< 0.03
70	< 0.005	< 0.024
80	< 0.005	< 0.001
90	< 0.005	< 0.0001
100	< 0.005	< 0.0001
120	-	< 0.0001
160	-	< 0.0001
200	-	< 0.0001

VIII. DISCUSSION

The results above support our hypothesis that LOST robots with a narrow ($60 < \alpha < 120$ degree) field of view perform better than the original 360 degree FOV. The occupancy histograms show that when the FOV is restricted, robots form a number of less-direct paths, each with a reduced probability of interference. But when the robots have a larger field of view, they converge to a smaller number of more-direct paths, with increased probability of interference.

The dispersal of narrow FOV robots over multiple trails can be seen as a spontaneous load balancing on the routes. Consequently, the robots enter/exit home and source from different sides which leads to another mechanism to distribute robots into trails. This also reduces the congestion near home/source.

With small population sizes where interference is not significant, the narrow field of view performs no differently than the original 360 degrees.

IX. CONCLUSION AND FUTURE WORK

In this paper we showed how changing the field of view of a sensor can influence the overall system performance of an ant-inspired foraging-and-trail-following robot system. It was shown through simulation experiments that if the receptive angle of the trail-marker sensor is narrow, but not too narrow, (about 90 degrees) the overall performance of our swarm was maximized. To the best of our knowledge, this is the first study that shows hiding some information from the agents can reduce the congestion and improve the overall performance. The "worse-is-better" data-hiding idea used here may be applicable to other large-scale multi-robot systems.

In future work we will investigate navigation strategies for trail-following multi-robot systems. To date, our LOST robots use a very simple controller for trail-following and local obstacle avoidance. We expect that performance can be improved by adding more data into crumbs, and exploiting the trails more intelligently. Also, local coordination strategies such as flocking or formations could increase navigation efficiency and possibly throughput while maintaining the attractive features of the LOST method.

ACKNOWLEDGEMENTS

This work was supported by NSERC and DRDC in Canada. Thanks to Greg Broten at DRDC, and to members of the Autonomy Lab for their constructive input.

CODE PUBLICATION

All source code, scripts, etc. used to produce the results reported in this paper are available online:
URI: http://autonomy.cs.sfu.ca/source/abbas_blinkered.tar.gz
SHA1: b0f49743f408701a4ce0411f2cff296fbb8b215e

REFERENCES

[1] R. T. Vaughan, K. Støy, G. S. Sukhatme, and M. J. Matarić, "Whistling in the dark: cooperative trail following in uncertain localization space," in *Proceedings of the Fourth International Conference on Autonomous Agents*, (Barcelona, Spain), pp. 187–194, June 2000.

[2] R. T. Vaughan, K. Støy, A. Howard, G. Sukhatme, and M. J. Matarić, "Lost: Localization-space trails for robot teams," *IEEE Transactions on Robotics and Autonomous Systems*, vol. 18, no. 5, pp. 796–812, 2002.

[3] M. Dorigo, *Optimization, Learning and Natural Algorithms*. PhD thesis, Politecnico di Milano, Italy, 1992.

[4] A. Russell, D. Thiel, R. Deveza, and A. Mackay-Sim, "Sensing odour trails for mobile robot navigation," in *Proc. Int. Conf Robotics and Automation*, pp. 2672–2677, IEEE, 1994.

[5] R. Fujisawa, H. Imamura, T. Hashimoto, and F. Matsuno, "Communication using pheromone field for multiple robots," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1391–1396, Sept. 2008.

[6] D. Payton, M. Daily, R. Estowski, M. Howard, and C. Lee, "Pheromone robotics," *Auton. Robots*, vol. 11, no. 3, pp. 319–324, 2001.

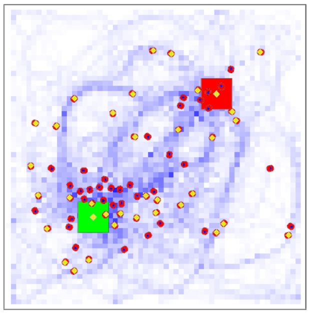
[7] K. Lerman and A. Galstyan, "Mathematical model of foraging in a group of robots: Effect of interference," *Auton. Robots*, vol. 13, no. 2, pp. 127–141, 2002.

[8] M. Zuluaga and R. Vaughan, "Reducing spatial interference in robot teams by local-investment aggression," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (Edmonton, Alberta), August 2005.

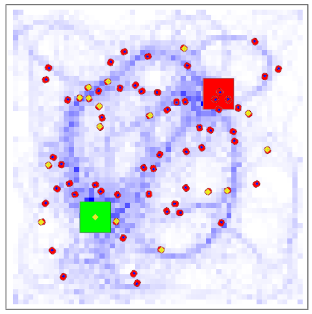
[9] M. M. Alexander Scheidler, Daniel Merkle, "Congestion control in ant like moving agent systems," *Biologically-Inspired Collaborative Computing*, vol. 268, pp. 33–43, 2008.

[10] R. T. Vaughan, "Massively multi-robot simulations in stage," *Swarm Intelligence*, vol. 2, no. 2-4, pp. 189–208, 2008.

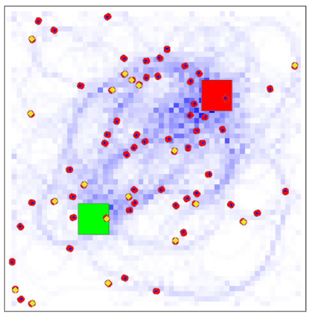
[11] S. Camazine, N. R. Franks, J. Sneyd, E. Bonabeau, J.-L. Deneubourg, and G. Theraula, *Self-Organization in Biological Systems*. Princeton, NJ, USA: Princeton University Press, 2001.



(a) $t_1 = 10min$

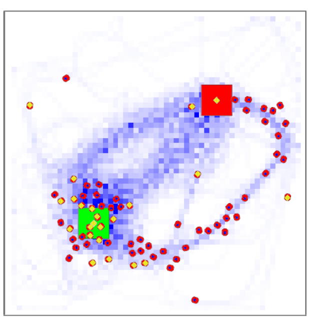


(b) $t_2 = 20min$

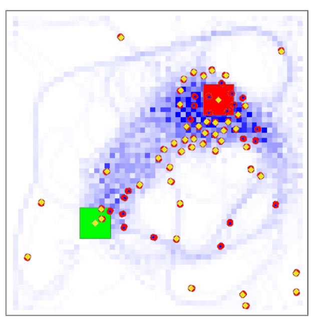


(c) $t_3 = 30min$

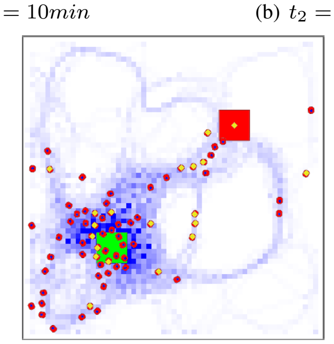
Fig. 5. Histograms of robots' locations in the last 5 $mins$ with $\alpha = 90, d_f = 2m$



(a) $t_1 = 10min$

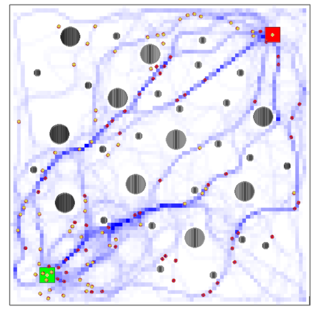


(b) $t_2 = 20min$

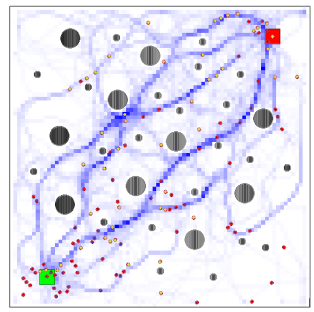


(c) $t_3 = 30min$

Fig. 6. Histograms of robots' locations in the last 5 $mins$ with $\alpha = 180, d_f = 2m$

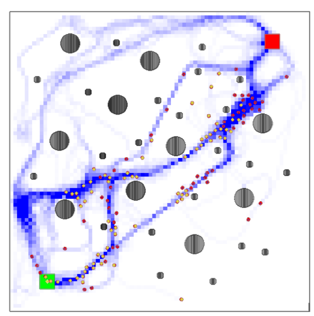


(a) $t_2 = 20min$

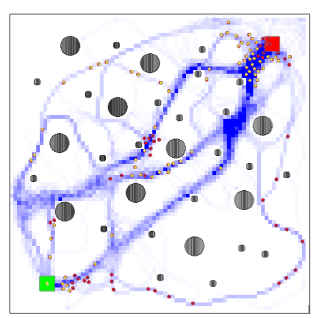


(b) $t_3 = 30min$

Fig. 7. Histograms of robots' locations in the last 5 $mins$ with $\alpha = 90, d_f = 2m$



(a) $t_2 = 20min$



(b) $t_3 = 30min$

Fig. 8. Histograms of robots' locations in the last 5 $mins$ with $\alpha = 180, d_f = 2m$