

REDUCING SPATIAL INTERFERENCE IN ANT-LIKE TRAIL-FOLLOWING FOR MULTI-ROBOT SYSTEMS

by

Seyed Abbas Sadat Kooch Mohtasham
B.E, Iran University of Science and Technology, 2004

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
in the School
of
Computing Science

© Seyed Abbas Sadat Kooch Mohtasham 2010
SIMON FRASER UNIVERSITY
Fall 2010

All rights reserved. This work may not be
reproduced in whole or in part, by photocopy
or other means, without the permission of the author.

APPROVAL

Name: Seyed Abbas Sadat Kooch Mohtasham
Degree: Master of Science
Title of Thesis: Reducing Spatial Interference In Ant-Like Trail-Following For Multi-Robot Systems

Examining Committee: Dr. Brian Funt, Professor,
School of Computing Science
Simon Fraser University
Chair

Dr. Richard Vaughan, Professor,
School of Computing Science
Simon Fraser University
Senior Supervisor

Dr. Greg Mori, Professor,
School of Computing Science
Simon Fraser University
Supervisor

Dr. Alexandra Fedorova, Professor,
School of Computing Science
Simon Fraser University
Examiner

Date Approved:

Abstract

We consider the classical task of transporting resources between spatially separated source and sink by a group of autonomous robots. The robots use ant-like trail-following to navigate between source and sink. This study consists of two independent parts, both aiming to improve the performance of the system by reducing the spatial interference among robots. In the first part it is shown that, under certain conditions, a narrow field of view of each robot's trail following sensor 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. In the second part, we focus on reducing the interference among robots that are following different trails. We propose a navigation strategy that is effective in separating these trails. The results of simulation experiments indicate that the performance of robots is usefully increased compared to the original algorithm in constrained environments.

*To my beloved parents,
Mohsen and Mina*

“Only from the heart Can you touch the sky.”

— *Rumi*, 13TH-CENTURY PERSIAN POET AND MYSTIC

“What is the most resilient parasite? Bacteria? A virus? An idea. Resilient... highly contagious. Once an idea has taken hold of the brain it’s almost impossible to eradicate.”

— *Inception*, WARNER BROS. ENTERTAINMENT INC., 2010

Acknowledgments

Here, I would like to thank Dr. Richard Vaughan who has been a wonderful supervisor. From the very first days at Simon Fraser University, he provided encouragement, support, and lots of good ideas. Also, I would like to thank my colleagues at the Autonomy Lab who help me in my research by their comments and discussions.

Finally, I wish to thank my family, specially my parents, for their support from when I was a little kid to now, even being on the other side of the world.

Contents

Approval	ii
Abstract	iii
Dedication	iv
Quotation	v
Acknowledgments	vi
Contents	vii
List of Tables	ix
List of Figures	x
List of Algorithms	xii
1 Introduction	1
1.1 Goal	2
1.2 Thesis Outline	2
2 Related Work	4
2.1 Bio-Inspired Swarm Robotics	4
2.2 Ant-Inspired Trail Following	5
2.3 Spatial Interference	9

3	LOST: An Ant-Inspired Trail-Following Algorithm	11
3.1	Introduction	11
3.2	Task	11
3.3	Localization-Space Trails	12
3.3.1	Events	12
3.3.2	Trails, Places and Crumbs	12
3.3.3	Trail-Laying and Trail-Using Algorithms	13
3.4	Trail Congestion and Spatial Interference	15
4	Blinkered LOST	18
4.1	Changing the Field Of View	18
4.2	Experiment 1	20
4.2.1	Simulation Setup	20
4.2.2	Results	20
4.3	Experiment 2	24
4.4	Fairness Evaluation	27
4.5	Discussion	28
4.6	Conclusion	28
5	SO-LOST	30
5.1	Spread-Out LOST	30
5.2	Experiments	35
5.2.1	Simulation Setup	35
5.2.2	Results	36
5.3	Discussion	36
5.4	Conclusion	39
6	Conclusion and Future Works	40
	Bibliography	42

List of Tables

3.1	Number of round-trips made by the swarm	16
4.1	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.	22

List of Figures

2.1	Virtual pheromone gradient formed by a team of robots [22]. Printed by permission of David Payton ©.	7
2.2	Path formation [20]. Printed by permission of Shervin Nouyan ©	8
2.3	GNATs deployed in the environment [21]. Printed by permission of Keith J. OHara ©.	9
2.4	Population vs. performance trade-off [33]. Printed by permission of Richard Vaughan ©.	10
3.1	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.	14
3.2	Effect of increasing the population in trail-following robots	16
4.1	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.	21
4.2	Histograms of robots' locations in the last 5 mins with $\alpha = 90, d_f = 2m$. . .	23
4.3	Histograms of robots' locations in the last 5 mins with $\alpha = 180, d_f = 2m$. .	23
4.4	Results of second experiment	24
4.5	Histograms of robots' locations in the last 5 mins with $\alpha = 90, d_f = 2m$. . .	25
4.6	Histograms of robots' locations in the last 5 mins with $\alpha = 180, d_f = 2m$. .	26
4.7	Fairness for different number of robots with different α_f	27
5.1	Trails formed in an obstacle-free "empty" environment using LOST and SO-LOST. SO-LOST has separated the trails, achieving better throughput due to reduced interference.	31

5.2	Trails formed in the cave environment using the LOST and the new algorithm after 30 mins of simulation.	32
5.3	Trails formed in the hospital environment. using the LOST and the new algorithm.	34
5.4	Sketch of the new LOST algorithm. While the robot was following the trail (filled circles), it sees a crumb with different goal (triangle) and thus changes its direction to a new point (the empty circle).	35
5.5	The result of the experiments in the 3 environments. The mean performance over 10 trials are shown with errorbars showing the standard deviation for both the original LOST and the new algorithm. The dotted line shows the data point for which the two algorithm do not show significant difference in distribution.	37
5.6	The histograms showing the number of times the goal crumb was shifted. The time bin is 30 seconds of simulation time. 30 robots are used in the empty environment and 50 robots are used in the other environments.	38

List of Algorithms

5.1 The Spread-Out LOST Trail-Using Algorithm 33

Chapter 1

Introduction

Multi-robot systems are becoming increasingly important. Many applications of robotics have emerged such as search and rescue, exploration, waste disposal, cleaning, transportation and etc., that are potentially parallelizable and therefore can be done more quickly by multiple robots. For example, in the exploration task in which an area should be completely visited by robots, while a robot is scanning a part of the environment, other teammates can simultaneously explore other parts and the whole task will be accomplished in a shorter time than when a single robot is being used.

The task of resource transportation can also benefit from multiple robots. This comes from the fact that this task is inherently parallelizable; i.e., each robot in the team can independently transport a unit of resource between spatially separated source and sink locations. In this study we assume that no map of the environment exists and the position of the source is not known by the robots. Starting from a home position, robots search for the source and then start to transport resources to home. We use an ant-inspired path finding and sharing algorithm (called LOST) in which robots lay waypoints in the localization space to form trails between the source and home. The trails are continuously refined online, and maintain the ant-algorithm property [9] of converging to near-optimal paths from source to home. LOST was introduced by Vaughan et. al. [30] and this thesis extends that work as follows.

In order to increase the throughput of the system, we increase the population size of the robot team. However, since ant algorithms (including LOST) tend to converge to a single best trail, in large populations this trail can become badly congested. The interference among robots in the congested areas does not let robots progress in the task by triggering

the obstacle avoidance behavior in the them. This problem exists in almost every multi-robot system.

1.1 Goal

In this thesis, we seek to perturb the ant algorithm to mitigate this problem. Due to the cost of obtaining global information in a distributed system, we do not want to use extra global information such as which trails are congested or how many robots are using a trail. Also no extra sensory information is going to be used. We create interference reducing effects by introducing some simple modifications to the trail-following algorithm.

We first 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. Secondly, we describe a modification to LOST, creating trails that share parts of the environment while being far enough apart to reduce interference. The result is superior performance in most of the cases we examine. The innovation is that the robots' trail-following behaviour is subtly modified to avoid competing trails, with the emergent effect that trails are iteratively spread out until interference is largely avoided.

1.2 Thesis Outline

Chapter 2: Related Work In this Chapter, a very brief introduction to biologically inspired swarm robotics is given including the advantages of such systems. Then the existing approaches to implement the ant trail-following algorithm in robots is explored. Finally we present some related methods used as interference reduction strategies in multi-robot systems.

Chapter 3: LOST: An Ant-Inspired Trail-Following Algorithm The LOST algorithm [30] used in our transportation system is explained in this Chapter. Inspired from ants' behaviour of pheromone laying and following, LOST defines concepts such as crumb, trail and event that are explained in this Chapter.

Chapter 4: Blinkered LOST In this Chapter we show how restricting the FOV of robots' trail detecting sensor will cause the robots to form multiple trails. The uniformity of the work distribution among robots is also examined.

Chapter 5: SO-LOST This Chapter covers an interference reduction method for trail-following robots which leads to an improvement in our transportation system. After introducing a small modification in LOST algorithm, the new trail-using algorithm is presented with experiments in simulation and results.

Chapter 6: Conclusions and Future Work Here a summary of the thesis is given along with some ideas for extentions.

Chapter 2

Related Work

2.1 Bio-Inspired Swarm Robotics

Living organisms, ranging from insects to humans, have been a source of inspiration for roboticists. Researchers in biology have also used robots to better understand the organisms in some possible levels [37, 36, 1]. The product of millions of years of evolution, living organisms exhibit intelligent behaviour and very flexible and efficient control of their body movement. Some robots have been built based on mechanical designs inspired from particular animals. Studying anatomy, kinematics and body structures and transferring the discovered principles to robots has become a way of designing robots. Snake-like robots [7, 40], fish robots [15, 18], flying robots [42, 8], to name a few examples, are built based on real instances in nature. This approach is called “biomimetics”. The behavioural characteristics of some animals have also been mimicked to create more intelligent robots. For example, in [38], a simple and robust strategy for robotic homing is introduced which is inspired by the visual homing behavior of bees and ants. Another example is the use of aggressive behaviors seen in animals to break the symmetry between robots and prevent deadlocks [33].

Swarm robotics, which is a new approach to the coordination of multi-robot systems, uses models inspired by collective behaviours in social insects. As an alternative to the conventional centralised controllers used to coordinate robots’ behaviours, researchers have proposed decentralised methods in which robots exhibit a desired collective behaviour, emerging from local intractions between individuals and their environment. Studies [4] have indicated that no centralised coordination mechanism exists in the colonies of social insects and yet

millions of individuals can accomplish a task and show coherent and collective behaviours. Dynamic task allocation among workers, cooperative transportation and nest building are some example tasks that some insects perform in nature.

In swarm robotics the emphasis is on using a large number of mostly simple and low-cost robots where decentralised controllers perform better than centralised solutions. Tasks that require miniature robots (nanorobotics, microbotics) such as distributed sensing in human body, are potential applications for swarm robotics. Also, swarm robotics could be a good solution for applications that demand cheap designs such as agricultural foraging tasks.

Robustness is claimed to be the key advantage of the swarm robotics approach and is manifested in a number of ways. First, the swarm can dynamically re-organise the way that the individual robots are deployed. This self-organization is possible since the robot swarm consists of a large number of fairly simple and usually homogeneous robots that are not permanently assigned a specific role or task within the swarm. Secondly, due to the redundancy in the system, the loss or failure of an individual robot does not damage the performance of the swarm very much. Thirdly, because the coordination mechanism in the robot swarm is completely decentralised, no common point of failure or vulnerability exists in the swarm. In fact, the robustness that is apparent in robotic swarms are inherent in the swarm robotics approach. However, in conventional robotics systems, this level of fault tolerance requires high engineering costs to achieve. Recent work by Winfield [39] suggests that swarm systems are vulnerable to certain failure modes indicating that claims of swarm robustness should be considered carefully. He shows that although robot swarms are tolerant to the complete failure of individuals, they can be less tolerant of partially failed robots.

2.2 Ant-Inspired Trail Following

In many ant species, pheromone trail laying and following behaviour is observed that serves as a way of communication. This mechanism of indirect communication is used by ants for recognizing enemy species, collecting food, aggregating, mating and etc. Being highly robust as well as adaptive, chemical pheromone communication enables the ant swarm to accomplish complex tasks. This behaviour of ants is adopted and used in theoretical computing science to solve complex optimization problems [6].

The formation of chemical trails between home and the food sources has served as a useful idea for path finding and navigation in swarm robotics. Individual ants deposit small

amounts of pheromone while travelling in the environment in search of food sources. Due to the evaporation of pheromone, the old trails eventually vanish whereas the recent trails can still be sensed in the environment. Ants can smell the nearby pheromone laid by all other ants in the environment while trying to reach a food source. A forager ant follows the trail that is frequently used and has more chemical than other alternative trails. This leads to the further reinforcement of that trail. However, in some occasional situations, the unmarked or lightly marked trails are travelled by ants which helps to discover new trails in case the environment changes. At the beginning, the ants start searching for food by random movement and accordingly several trails are formed that indicate a path between the nest and the food sources. More pheromone is accumulated in shorter trails and therefore the probability that foragers follow these trails increases which consequently leads to further trail reinforcement. Eventually, the whole colony are attracted to the shortest trail between the nest and the foodsource while other trails are rarely used or completely evaporated.

This method of cooperative path finding and sharing between individuals has been implemented in robot swarms in different ways. The central component of this approach is the representation of pheromone. In any implementation of pheromone trail-laying and following approach, individuals in the robot swarms should be able to perceive and manipulate the pheromone in the environment. This pheromone can be physically embedded in the real world or be represented symbolically in the robots' memory.

Real chemical marks were first used to produce true stigmergic trail-following in [24]. Also in [23], different chemicals were used by robots to trigger different behaviors. A leader robot would release a specific chemical to signal the others to perform congregating behavior and then used another chemical to instruct others to switch to light seeking behaviour. The experiments were done in an indoor environment keeping the background airflow low. Recently, Fujisawa et al.[11] 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. In these approaches, the robots should be able to deposit and sense real chemicals in the environment. The challenge of chemical and sensor engineering makes these methods often impractical.

In other work, the individual robots play the role of pheromone and form a path between two places of interest in the area [10, 5]. Payton et al. [22] invented a method 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



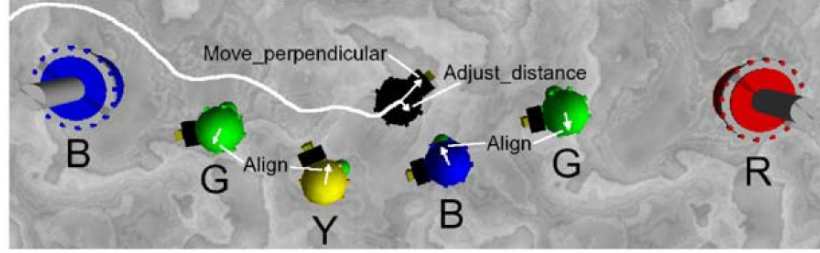
Figure 2.1: Virtual pheromone gradient formed by a team of robots [22]. Printed by permission of David Payton ©.

to estimate the distance to the message source (figure 2.1).

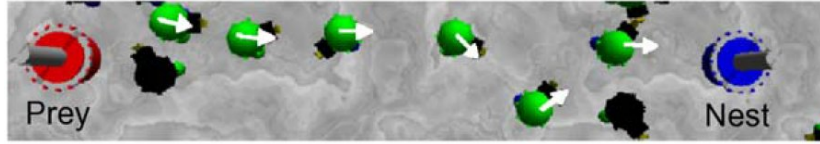
In [20], two mechanisms, chains and vectorfield, are proposed that enable robots to form a path between two target objects in the environment. When perceiving the target, a robot acts as pheromone and attracts others. The robots show different colors to direct robots in the path based on the chain or vectorfield mechanisms (see figure 2.2).

In other approaches, a network of immobile devices are deployed in the environment and robots are able to communicate with each node and get the direction to its goal. In [21], a system composed of small communication nodes are pre-deployed in the environment that serves as a navigation network that performs path planning to guide robots in their tasks (figure 2.3). This method allows complete distributed path planning in a dynamic world without mapping or localization.

Barth developed a dynamic programming method for directing the global motion of the swarm via the use of immobile relay markers[2]. The pheromone value of each marker is updated through the communication between the markers. Mamei et. al. used RFID tag



(a) In the Chain mechanism, the robot follows the direction perceived through the colors each robot shows.



(b) In the Vectorfield mechanism, each robot explicitly show a direction which leads to the target.

Figure 2.2: Path formation [20]. Printed by permission of Shervin Nouyan ©

technology to create a cheap and general-purpose pheromone representation [19]. Places of interest in the environment such as doors, or rooms can be marked by RFID tags and tracked by other robots later. Ziparo et al. have also used RFID tags to coordinate a team of robots for exploration in large areas [41]. A recent work [12] proposed a sparse representation of the pheromones using movable beacons that requires no communication between beacons. Robots can manipulate the value of beacons and move them in the environment.

In all the work above, the pheromone was represented by something that is sensed directly from the environment or exists in any form in the environment. Vaughan et al. [30, 34, 32] proposed an ant-inspired trail-following algorithm in which the gradient to the target place is built by global waypoints labeled by distance-to-goal measure. They showed that this scheme can be robust to large zero-mean localization error [34], and admits a relaxed and practical definition of mutual localization [30]. In this study we use this method (explained in Chapter 3) as the ant-inspired path finding and sharing algorithm for the robot swarm.



Figure 2.3: GNATs deployed in the environment [21]. Printed by permission of Keith J. OHara ©.

2.3 Spatial Interference

In multi-robot systems, robots may compete for resources such as communication, energy and space. Spatial interference, considered as a competition for space, can be very damaging specially when a group of autonomous robots with no centralized control are working in a constrained environment. In [33], Vaughan et al. proposed a hypothetical diagram showing the relationship between performance and population size in a transportation system (figure 2.4).

In a mathematical model of robot foraging [17], it was shown that adding more robots to the system improved the group performance while decreasing individual robot performance. Based on that model, an optimal group size was found that maximizes the group performance.

Interference reduction mechanisms have been developed to increase the area under the

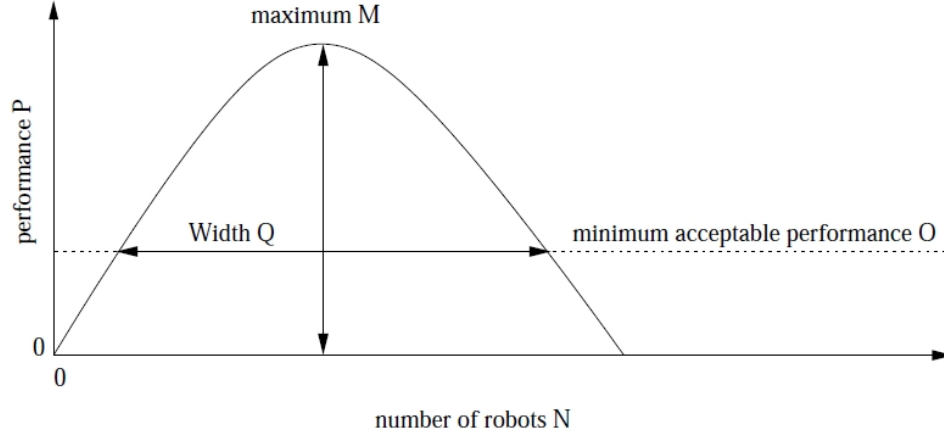


Figure 2.4: Population vs. performance trade-off [33]. Printed by permission of Richard Vaughan ©.

curve (figure 2.4) but above minimum performance. Territorial division is used to keep robots away from each other's work site and thus reduce interference [28]. In bucket brigading [45], each forager restricts itself to a specific area and relies on other workers to deliver the resource to the destination. None of the pheromone-trail following methods described in section 2.2 explicitly consider interference reduction.

Explicit anti-interference strategies are studied both in simulations and real robots to increase performance in the transportation task [43]. Aggressive display behaviors were used as a mechanism to resolve a conflict in the very constrained parts of the environment. Based on the amount of work they had invested up to that point, robots selected an aggression level and the difference in aggression levels between interfering robots was used to break the symmetry. In [27] some methods are proposed and studied to reduce and control the emergent congestion in a dense ant like moving agent system. The goal was to devise methods where the behavior of the agents is not changed or is changed only slightly. Also, no global or additional sensory information was used. For this purpose, asymmetries that resolve conflicts are introduced by modifying either the environment or the robot controllers. In addition to reducing congestion (and therefore improving the performance), the fairness of the proposed method is shown to be no lower than before.

This thesis introduces explicit anti-interference strategies to trail-following robot systems, with the goal of increasing performance.

Chapter 3

LOST: An Ant-Inspired Trail-Following Algorithm

3.1 Introduction

In this chapter we describe Localization-Space Trails (LOST) [30, 34, 32]. In Chapters 4 and 5, our novel extensions to LOST is described.

LOST enables a swarm of robots to navigate between places of interest in an initially unknown environment. It uses waypoints generated by each robot and shared with other team-mates. This method is inspired by ants' foraging behavior. Many ant species have trail-laying trail-following behavior when foraging: individual ants deposit a chemical substance called pheromone as they move from a food source to their nest, and foragers follow such pheromone trails. The process whereby an ant is influenced toward a food source by another ant or by a chemical trail is called recruitment, and recruitment based solely on chemical trails is called mass recruitment. In this section, two phenomena based on mass recruitment are described.

3.2 Task

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. It is a canonical multi-robot task since the work is inherently parallelizable. Achieving this task reliably with autonomous robots will meet a real-world need. The layout of the environment might change over time in real world and there might be robot breakdowns that block a route. The use of multiple robots for path-finding and sharing will help the team find a new route without a new up-to-date map or human intervention.

3.3 Localization-Space Trails

The LOST algorithm introduces a data structure and algorithms to use and manipulate it which is inspired from ants. In this section we briefly review this method.

3.3.1 Events

An Event is defined as a task-related occurrence which can be perceived by robots. Usually the robots are aimed to experience a series of Events. For example in the transportation task, Events can be ‘pick-up-resource’ and ‘drop-resource’. At any moment, each robot has a goal event that it seeks to experience. For instance at the beginning all the robots seek to experience the ‘pick-up-resource’ Event. After the robot recognizes its current goal Event, it switches to another goal Event, e.g. ‘drop-resource’.

The robots record an estimate of the position at which it experiences an Event and then it can express information about the environment relative to the location of these Events. This enables other robots to understand the shared data and interpret the position estimates in their own local frame of reference. However this requires all the robots to recognize and give common names to Events.

3.3.2 Trails, Places and Crumbs

We define a place as a tuple (E, L) in which E is an Event and L is a location in the localization space. A place is a global task-level landmark. In the transportation task, multiple robots are travelling between the source (A) and the sink (B). Since there is no global coordinate system, the coordinate at which an Event is experienced is different for each robot. But if a robot refers to another Event C relative to the estimate locations of A and B , all the robots can interpret a reference to C in their own local coordinates in terms of A and B .

Another data structure which is used in LOST is called *Crumb*. Crumbs are the way-points that form the trails. They describe the distance to a Place from a particular location in the local coordinate system. 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. We use time-to-goal as the distance function.

A Trail is a set of Crumbs and Places. At the beginning of the task, each robot has an empty trail which is built up over time according to the trail-laying algorithm. A very simple trail might look like this:

```
[A, (1,1)]           #Place
[B, (10,5)]          #Place
[A, (2,3), 30, 101] #Crumb
```

This could be expressed in English as: in the coordinate system in which Place A is at (1,1) and Place B is at (10,5), A was 30 distance units away from (2,3) at $t=101$ seconds.

3.3.3 Trail-Laying and Trail-Using Algorithms

In Lost, each robot maintains an initially empty trail. When a robot receives a trail on the network, the received trail is combined with the local trail. Two Trails can be combined if they have two Places in common, assuming all coordinates lie on the same plane. If Trail α contains Places at locations A_α and B_α and Trail β contains places A_β and B_β , there is a unique transformation γ that maps the vector from A_α to B_α onto the vector from A_β to B_β :

$$\gamma(A_\beta \vec{B}_\beta) = A_\alpha \vec{B}_\alpha$$

where γ is a simple linear coordinate transform consisting of a translation, scaling and rotation.

Once γ is determined using the Places common to both Trails the same transformation can be applied to map the coordinates of any location L , whether in a Crumb or a Place, from one Trail to the other. To combine the current Trail with a new Trail, every member of the incoming Trail is transformed through γ and added to the local Trail.

The main Trail is 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 . This Trail decay simulates the evaporation of pheromone in Ants trail following.

Besides the main Trail, Each robot has a temporary trail which initially consists of copies of all the Places in the robot's main Trail. As the robot moves in the environment and travel between Places, it lays Crumb i.e., at every S seconds, it inserts a new crumb to the temporary trail. The new 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. If an 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. In this way, as the robots explore the environment, experiencing Events, periodically broadcasting Trails ('laying Crumbs'), and (more frequently) receiving Trails on the network, each robot builds a Trail structure containing Places and Crumbs that describe how to get to Events.

The purpose of LOST is to guide the robot to a Place (Event) 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 (here in time) to reach the goal.

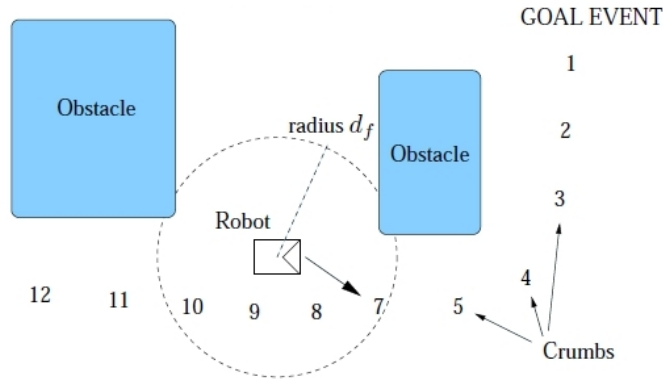


Figure 3.1: Sketch of the LOST algorithm, showing a trail of Crumbs with decreasing distance values leading to a goal Event. The robot moves towards the crumb within radius d_f that has the lowest distance estimate.

Suppose a robot at pose L_r has Place P_g as its goal, such as $Event(P_g) = \text{'drop-resource-at-home'}$. The robot searches the set of Crumbs with Place = P_g to find the set of crumbs

that lie within its *field of view* (FOV), 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 $L_c = \text{Pose}(C_L)$. Figure 3.1 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 probably 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.

3.4 Trail Congestion and Spatial Interference

Generally, a task will be completed faster using multiple robots rather than single robot. This is more obvious in tasks that are inherently parallelizable e.g., foraging and transportation. However, since the robots should work in the same environment as required by the task, they might block each other’s way during the normal navigation in the environment. If a robot senses another robot in its vicinity, it performs obstacle avoidance behaviours to prevent collision. When the population of the swarm is not very large, the robots spend most of their time working, i.e. they progress in the task. Adding a robot to the team increases the overall performance but, it also increases the probability of spatial interference among individuals. As the population grows, robots spend more time avoiding each other rather than working. In the worst case, the environment is full of robots so that they cannot move and the performance drops down to zero.

This general problem of interference in multi-robot systems is present in our trail-following robot system too. Since LOST (and maybe ant algorithms in general) tend to converge to a single “best” trail, in large populations this trail can become badly congested. Consequently, the robots experience a great amount of spatial interference and the performance is reduced.

Figure 3.2 shows this phenomenon in our trail-following robot system implemented in the well known simulator Stage [29]. In Figure 3.2(a), 10 robots are successfully following

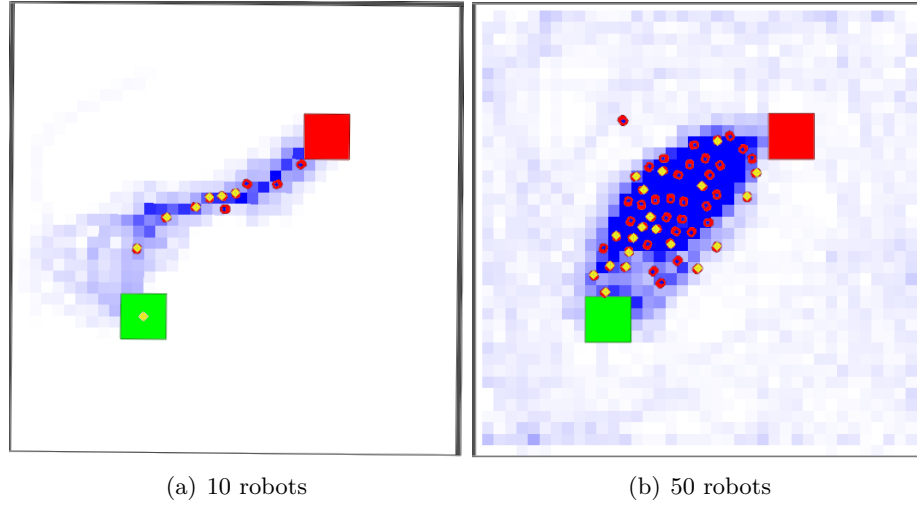


Figure 3.2: Effect of increasing the population in trail-following robots

Table 3.1: Number of round-trips made by the swarm

Population Size	Overall Performance
1	18
10	232
50	368

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 (table 3.1), which is close to the ideal of ten times the single robot work rate (18 round-trips per hour). In Figure 3.2(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.

Any robotic system that requires robots to operate in the same environment, should

somehow have some strategies to prevent this damaging effect of interference. In the next two Chapters, we describe our work that aims to ameliorate this problem.

Chapter 4

Blinkered LOST

The work presented in this Chapter was published as a refereed conference paper [26].

In this Chapter we present a simple method that increases the efficiency of trail-following in large population of robots. As it was discussed in the previous Chapter, too many robots using one single trail prevents them from progressing in the task due to high use of obstacle avoidance behaviors rather than working. 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. Usually there can be more than one trail formed and used for travelling between two places. This is more obvious when the environment is relatively open.

The contribution of this Chapter 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.

4.1 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 [4].

Another biological mechanism reported to break the symmetry in a group of individuals is *alloethism* in some species of ants, which is defined as a difference in behavior in a particular category of behavior as a function of worker size. In leaf-cutting worker ants, size polymorphism exists and the morphological trait variation restricts or enables an individual to carry out certain tasks and thus division of labour is obtained. Ants possess a well-developed olfactory system and the collective behaviour is governed using pheromone communication. Studies [13] have shown that trail pheromone is perceived differently by small and large workers which is due to the difference in the neuroanatomy of the first olfactory neuropil. This variation among the workers leads to different information processing of pheromone and a basis of differences in trail-following behaviour. The outcome will be the promotion of division of labour in the colony by allocating workers to foraging trails or colony defense [14].

The restriction of the robots' FOV seems to generate the same behaviour as *alloethism* in leaf-cutting ants, i.e. robots that are close to each other do not sense the same information as opposed to when wide FOV is used. This difference in crumb perception will lead to allocation of robots to distinct trails. This is exactly the same as in ants that workers are assigned to foraging or colony defence or brood caring as a consequence of differences in how pheromone is processed.

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° .

4.2 Experiment 1

4.2.1 Simulation Setup

To test our hypothesis, we ran Stage simulations with the task environment shown in Figure 3.2. 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 combinations of $d_f = [1.5, 2.0, 2.5]$ meters, $\alpha_f = [10, 20, \dots 360]$ degrees, population $P = [10, 20, \dots 100]$ robots.

4.2.2 Results

The results of the first experiment are summarized in Figure 4.1, with mean performance over 10 repeated trials plotted for each $[d_f, \alpha_f, P]$ configuration. Error bars are omitted 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. However it seems that with $\alpha_f \leq 60$ the robots are not able to follow the crumbs properly and tend to loose them quickly leading to a poor performance. 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.

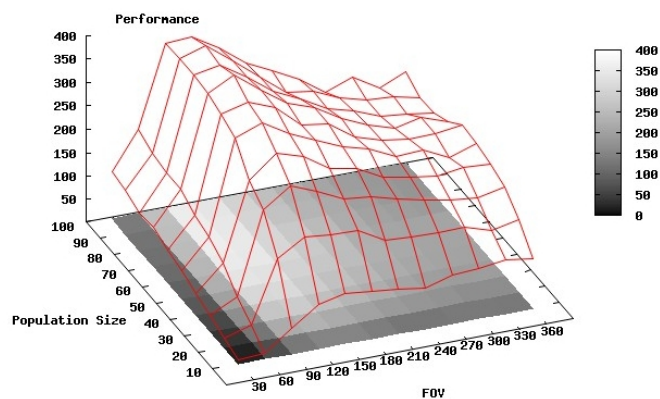
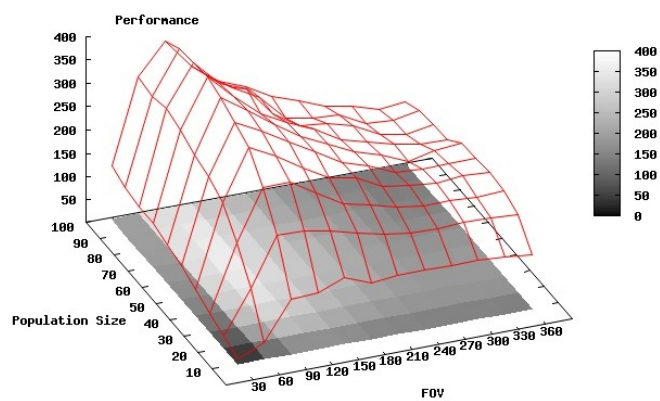
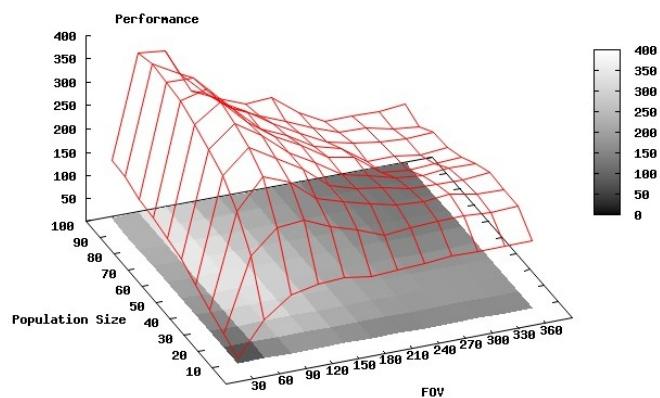
(a) With lookahead distance $d_f = 1.5m$ (b) With lookahead distance $d_f = 2m$ (c) With lookahead distance $d_f = 2.5m$

Figure 4.1: 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.

Table 4.1: 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

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 $\alpha_f=90$ and $\alpha_f=180$ are from the same distribution are given Table 4.1. 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 4.2, 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 4.3, with a wider FOV [$d_f = 2, \alpha_f = 180, P = 70$], we see fewer, wider trails and

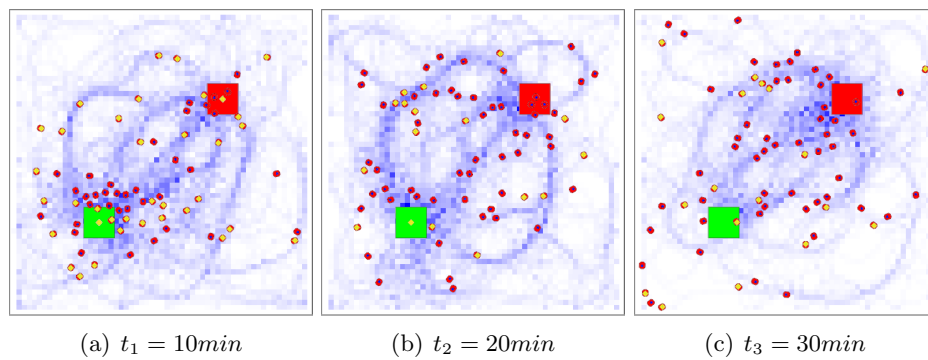


Figure 4.2: Histograms of robots' locations in the last 5 mins with $\alpha = 90, d_f = 2m$

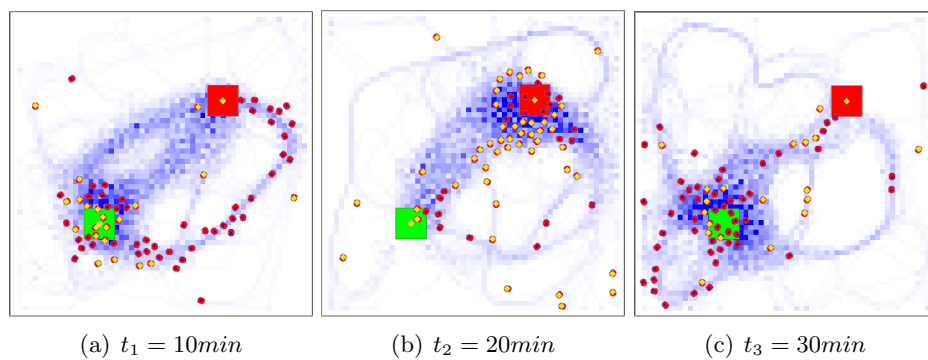


Figure 4.3: Histograms of robots' locations in the last 5 mins with $\alpha = 180, d_f = 2m$

robots closer together.

4.3 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, $\alpha_f = [10, 20, \dots 360]$ degrees, population $P = [10, 20, \dots 100, 120, 160, 200]$ robots.

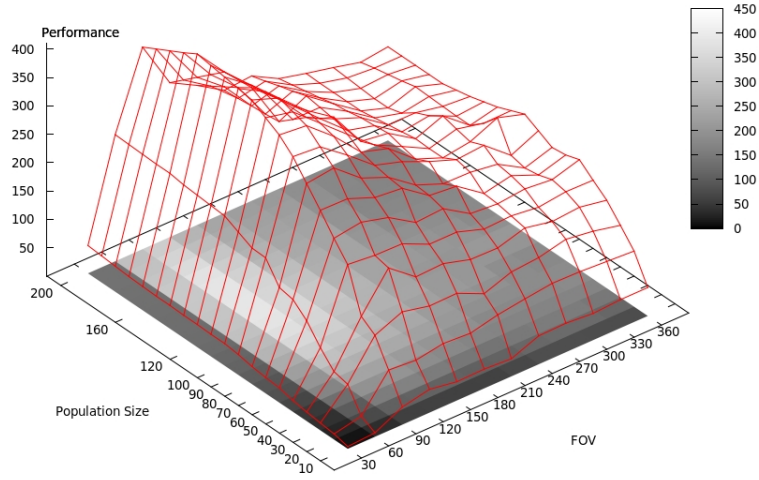
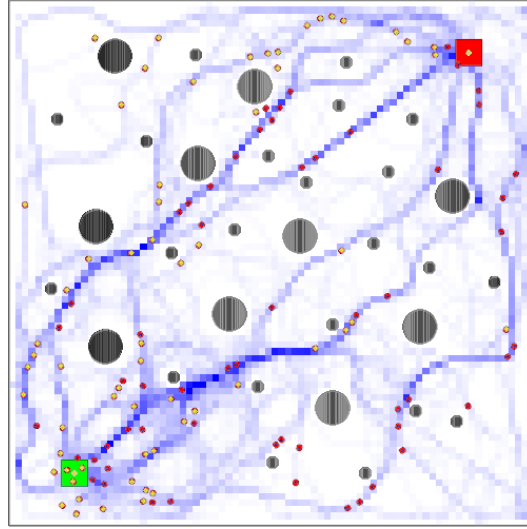
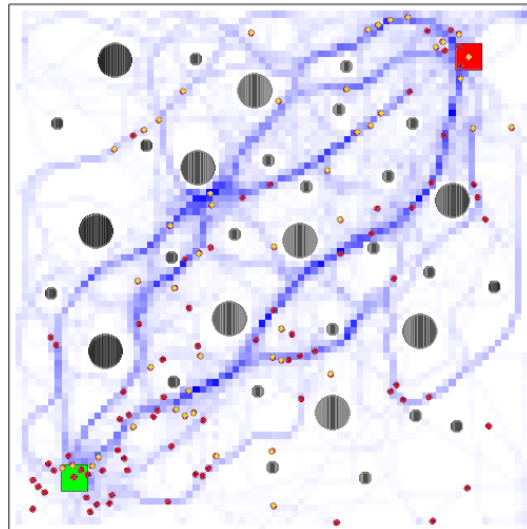


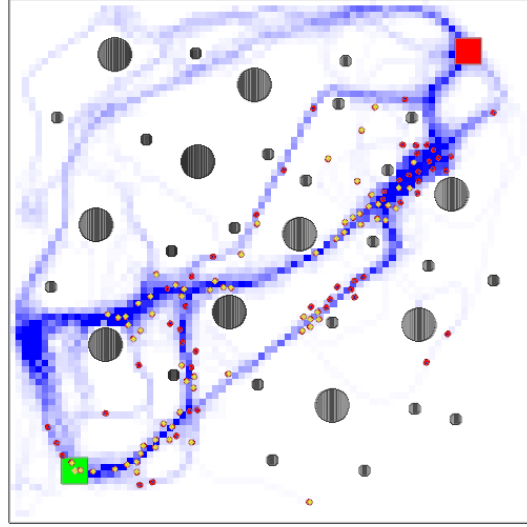
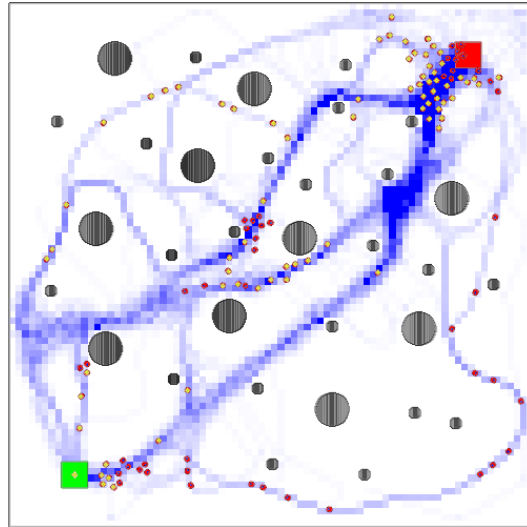
Figure 4.4: Results of second experiment

The results are plotted in Figure 4.4, 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 4.1 right side).

Occupancy histograms are shown in Figures 4.5 [$d_f = 2, \alpha_f = 90, P = 120$] and 4.6 [$d_f = 2, \alpha_f = 180, P = 120$]. 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.

(a) $t_2 = 20\text{min}$ (b) $t_3 = 30\text{min}$ Figure 4.5: Histograms of robots' locations in the last 5 mins with $\alpha = 90, d_f = 2m$

(a) $t_2 = 20\text{min}$ (b) $t_3 = 30\text{min}$ Figure 4.6: Histograms of robots' locations in the last 5 mins with $\alpha = 180, d_f = 2m$

4.4 Fairness Evaluation

The results of the experiments in the previous section shows how the FOV angle affects the trail-following behavior of the robots and thus influences the overall performance of the swarm. Besides the performance and congestion reduction, fairness is another important measure to consider for evaluating the algorithm. Generally speaking, a system is fair if the clients receive the same share of system resources. Here, the trail-following algorithm is fair if the number of times each robot visits the source is the same. Therefore, the more the work distribution among robots is similar to uniform distribution, the fairer the trail-following algorithm is. Fairness is a vital requirement for some specific tasks. For example if we consider a charging station as the *source*, then it is important that the robots visit the source about the same number of times.

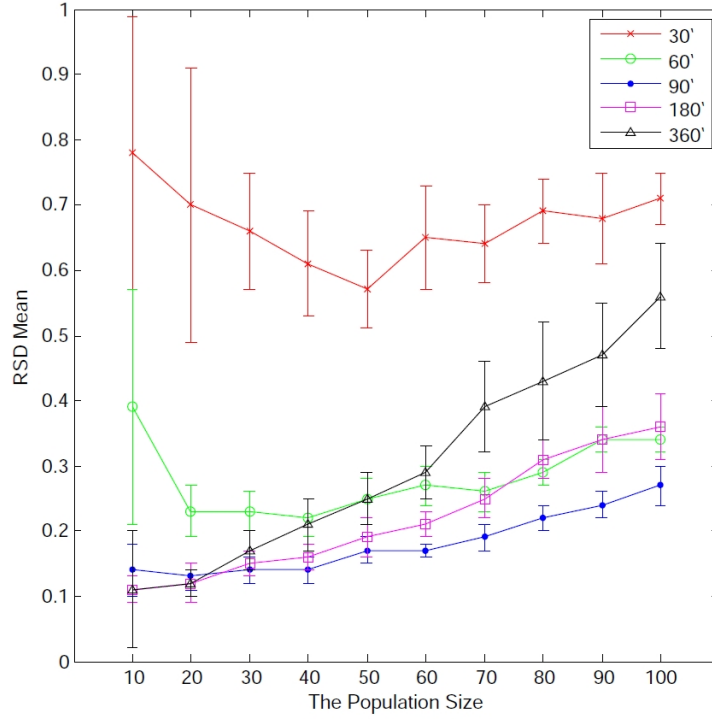


Figure 4.7: Fairness for different number of robots with different α_f

Here, we use relative standard deviation (RSD) of the number of resources transported by each robot as the fairness measure for the trail-following algorithm (the lower the RSD, the more fair the system). Figure 4.7 shows the behavior of the algorithm with different

population sizes and different FOV angle with respect to fairness. For clarity, only the plots for a selected subset of the α_f is shown.

With $\alpha_f = 30$, the system is not fair which can be due to the fact that robots tend to lose the trails when using a very narrow FOV. Increasing the FOV angle to 60 degree makes the fairness better for population sizes more than 10 robots. With $\alpha_f \geq 90$ and small populations, the robots have a good sensing of the crumbs and there is not much congestion. The system shows a fair behavior in this case.

However, when the population size increases, $\alpha_f = 360$ becomes unfair. With robot swarms of size ≥ 30 , the crumb-detecting sensor with $\alpha_f = 90$ is the most fair configuration.

4.5 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.

4.6 Conclusion

In this Chapter 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.

Chapter 5

SO-LOST

The work presented in this Chapter was published as a refereed conference paper [25].

In the transportation task, robots are repeatedly traveling between source and home. As one robot is navigating to home to deliver a resource, it should avoid robots that are going to source to fetch resource. The ant trail-following mechanism causes the two “opposite” trails to be laid very close to (if not on) each other. In other cases such as when there are multiple sources/homes or when the robots lay crumbs to form trails to charging stations as well, these trails are very prone to be formed in a way that might lead to spatial interference between robots following them. In environments that have narrow corridors or doorways, this problem is inevitable.

A simple method is proposed in this Chapter to separate trails with different goals. In trail-following algorithms, the unattractive crumbs (crumbs with different goal than the robot’s goal) are usually ignored. However, these crumbs are cues to robots that might pass by the current location while following those crumbs for some goal place. Based on this idea, after a small modification in trail-laying algorithm, we present the new trail-using strategy to avoid these crumb.

5.1 Spread-Out LOST

In the LOST algorithm, as the robots move they “lay” crumbs. The goal Place of these crumbs is the place that the robot has most recently visited. This means that in order to reinforce a trail, the robots should travel in the opposite direction that the crumbs are showing and consequently robots following a trail are very likely to interfere with robots

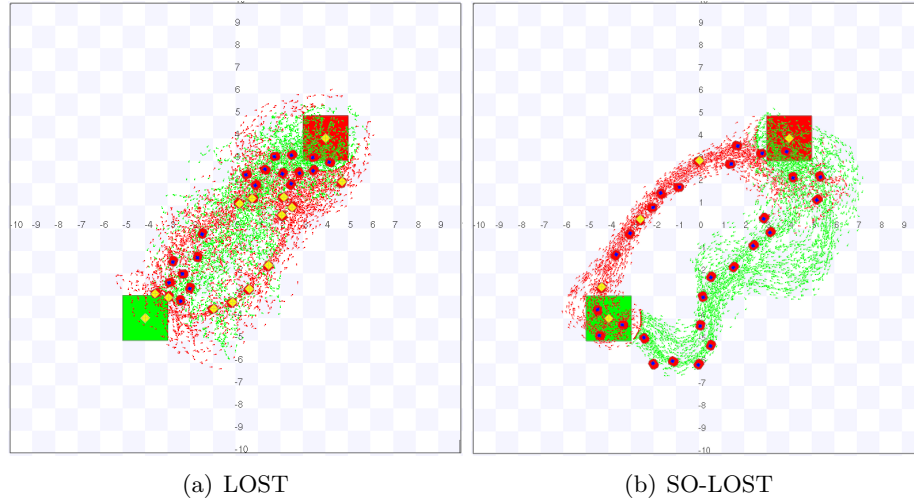


Figure 5.1: Trails formed in an obstacle-free “empty” environment using LOST and SO-LOST. SO-LOST has separated the trails, achieving better throughput due to reduced interference.

laying (reinforcing) it. With few robots, this does not have much effect on performance and the “pick-up-source” and “drop-resource” trails converge to one shortest discovered path. However as the robots’ team size increases, these interferences damage the performance of the system.

To address this, we modify the LOST algorithm so that when a crumb is created, the P_c data field will be the goal of the robot rather than the recently visited place. With this modification, the robots have to perform two searches at the beginning; one for finding a path from home to source and another one for a path from source to home. We can avoid the need for the second search by copying the first discovered trail and changing the goal and reversing the distance hint along the trail.

When the environment in which the robots are working is complicated and contains narrow corridors and doorways, or is very crowded, LOST may produce trails with different goals that are either very similar or have many parts in common. Figures 5.1(a), 5.2(a), 5.3(a) show this phenomenon in our trail-following robot system implemented in the well known simulator Stage ([29]). The trails formed between source and home are often very close to each other, leading to problematic interference between robots travelling in opposite directions. Since the crumb trail data structure does not contain any explicit information

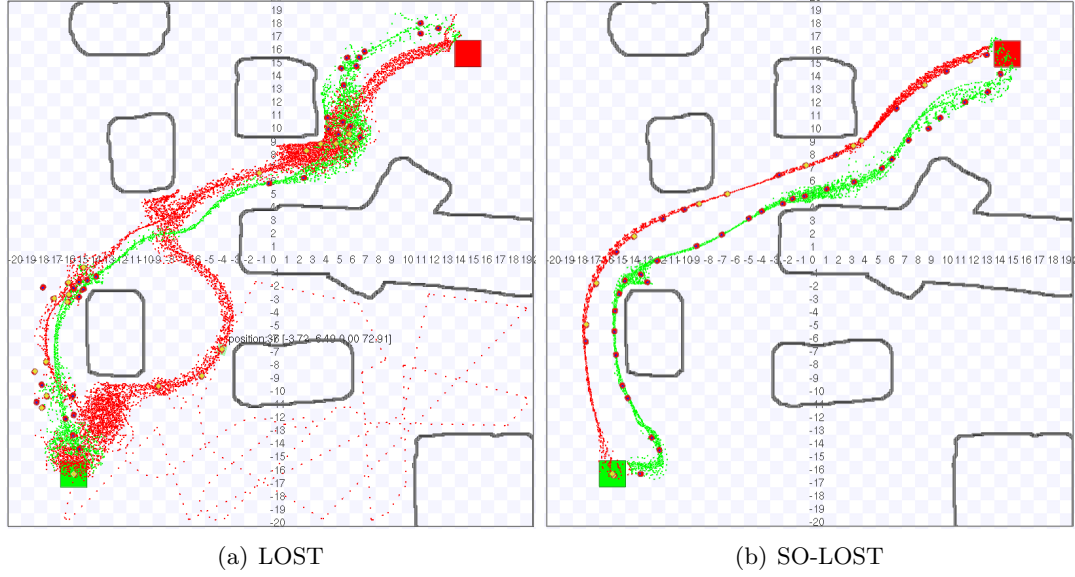


Figure 5.2: Trails formed in the cave environment using the LOST and the new algorithm after 30 mins of simulation.

about the fixed obstacles in the environment, there is no way to directly process the trail data to avoid robot-robot interference without risking directing robots into fixed obstacles. Instead, we use a small modification to the robots' trail following control strategy that results in emergent trail separation.

A robot following a trail to get to P_c , can interpret crumbs with goals other than P_c , as proxies for potentially interfering robots. If the robot follows the trail to P_c while slightly avoiding all other nearby crumbs, the new P_c crumbs it lays will tend to be slightly more distant from other crumbs than those just followed. This mechanism is essentially similar to the iterated corner-cutting that drives the ant-algorithm's ability to locally improve trail length. The resulting trails may be slightly longer but may reduce interference significantly, as suggested by the results below.

The new trail-using algorithm is presented in Algorithm 5.1. It first searches for the crumb c_{best} with minimum distance to goal that is located in the robot's FOV. Then if there exists a crumb c_{anti} with different goal than the robot's goal and it was closer to the robot than a distance threshold (*crumb_avoid*), the direction to which the robot moves will turn to the robot's left. This will change the angular velocity of the robot so that it keeps away

Algorithm 5.1 The Spread-Out LOST Trail-Using Algorithm

Require: The distance $dist_{obstacle}$ from the robot to the nearest non-robot obstacle on the left side of the robot.

return the direction Dir_{robot} to which the robot should move

Θ = all the crumbs in the robot's FOV with positions relative to the robot;

$\Sigma = \{c | c \in \Theta \wedge (c.p_c = robot.goal)\};$

$\Pi = \{c | c \in \Theta \wedge (c.p_c \neq robot.goal)\};$

$\lambda = Min(crumb_avoid, dist_{obstacle});$

$c_{best} = c$ s.t. $(c \in \Sigma) \wedge (\nexists c' \in \Sigma \text{ s.t. } c.d_c > c'.d_c);$

if $(\exists c_{anti} \in \Pi \text{ s.t. } dist(c_{anti}, robot) < crumb_avoid) \wedge (c_{best}.d_c \leq 2s)$ **then**

$$Dir_{robot} = \overrightarrow{(robot, c)} + \frac{\lambda}{2} \times \overrightarrow{(-1, 0)};$$

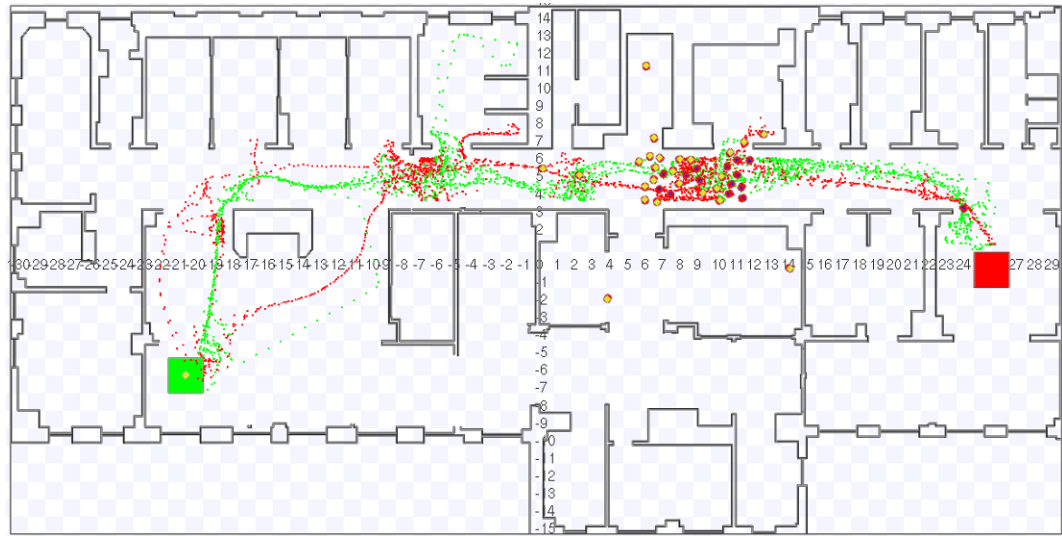
else

$$Dir_{robot} = \overrightarrow{(robot, c)};$$

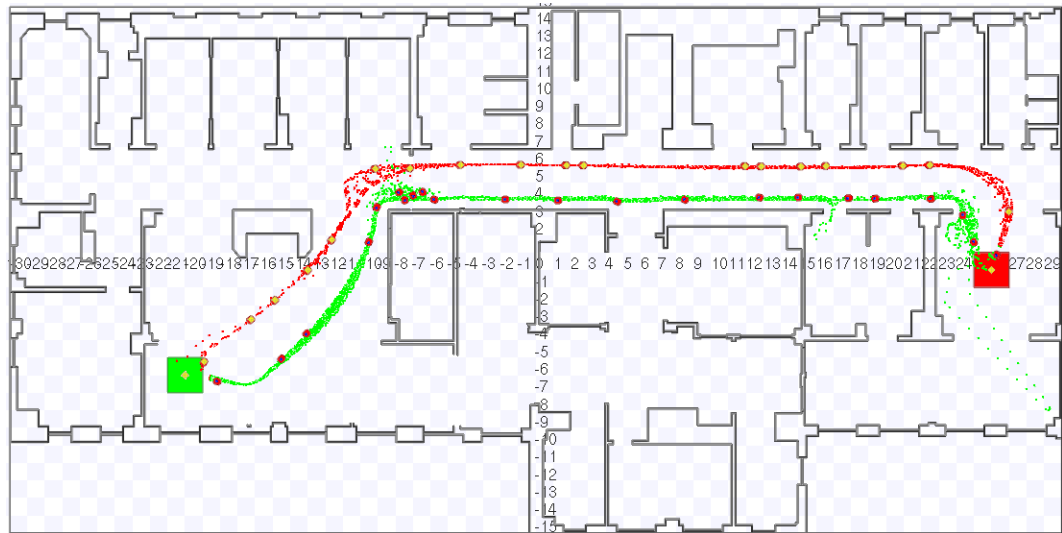
end if

from c_{anti} . The shift vector is orthogonal to the $\overrightarrow{(robot, c_{best})}$ vector. Also, the magnitude λ is calculated based on the obstacles near the robot such that the robot's target point does not lie inside an obstacle. Trails with different goals are necessarily very close to each other around source and home. Thus the shift vector is not applied when the robot is near the goals to prevent robot's circular trajectory in these areas.

Figure 5.4 illustrates how the behavior of the robot changes in presence of c_{anti} . The robot is following the small circles. On seeing the triangle crumbs, the robot's target point is changed from c_{best} to another point (the empty circle). This simple mechanism alters the robots' movement so that different trails are gradually separated from each other. The divergent movement of trails continues until they are away enough from each other, if possible.



(a) LOST



(b) SO-LOST

Figure 5.3: Trails formed in the hospital environment. using the LOST and the new algorithm.

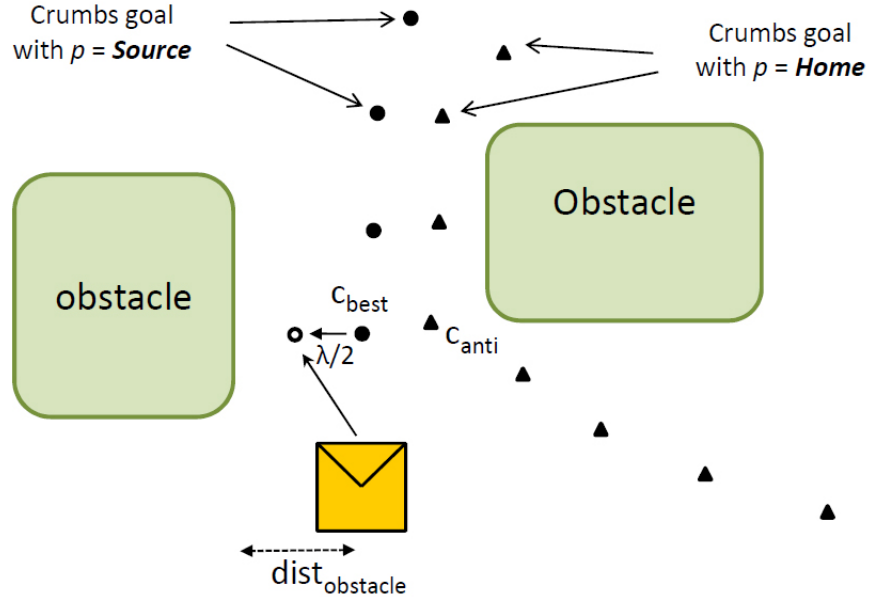


Figure 5.4: Sketch of the new LOST algorithm. While the robot was following the trail (filled circles), it sees a crumb with different goal (triangle) and thus changes its direction to a new point (the empty circle).

5.2 Experiments

5.2.1 Simulation Setup

We ran Stage simulations to evaluate the new algorithm in three different environment settings: *empty* (Figure 5.1), *cave* (Figure 5.2) and *hospital* (Figure 5.3). The size of the *empty*, *cave* and *hospital* environments are 20x20m, 40x40m and 60x30m respectively, with robot length 0.45m. 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 60 minutes, and the total number of resources delivered at the end of the trial is our performance metric. 10 trials are performed for each population size. LOST is deterministic

but the local obstacle avoidance and searching is stochastic (for robustness), hence the need for repeated trials. For all experiments the *crumb_avoid* parameter is set to 2m.

5.2.2 Results

The results of the experiments are summarized in Figure 5.5, showing the mean and standard deviation of performance over 10 repeated trials plotted for each population size. The plot shows a marked improvement in many cases (in some cases 3 times better) in performance with the new algorithm.

As expected, with few robots (20), there is not much difference in performance since the interference among robots is small. In the empty environment with population size of 10, LOST outperforms the new algorithm. This is because the benefit of interference reduction can not outweigh the penalty of increase in the length of the trails. As the population size increases and the environment becomes more constrained, improvement in performance gets bigger. This can be seen in the plot showing the results of the experiments in the hospital environment; For the smallest populations, the two methods perform about the same; however, since the hospital environment contains corridors and doorways (Figure 5.3(b)), there is a degradation in the LOST performance with more robots whereas the new algorithm improves the performance in some populations up to 3 times.

To verify that the performance results are significantly different for different algorithms, we performed hypothesis testing using a T-test. The P values for the hypothesis that the performance values for LOST and the new algorithm are from the same distribution are calculated. For all population sizes, the test suggests that the distributions are significantly different ($P < 0.02$), except for the pairs identified in Figure 5.5 with dotted ellipses.

5.3 Discussion

The new algorithm is based on the idea that laying crumbs near other crumbs with different goals increases the probability of co-location among the robots performing different tasks. This is more clear in transportation task in which the trails for ‘pick-up-resource’ and ‘drop-resource’ tasks can be formed very close to each other. In the new algorithm robots follow the trails and also try to keep a distance from other crumbs and therefore new trails are laid at a safe distance from each other. Figures 5.1(b), 5.2(b), 5.3(b) show the trails formed with the new algorithm. It is visible that different trails are separated from each other and

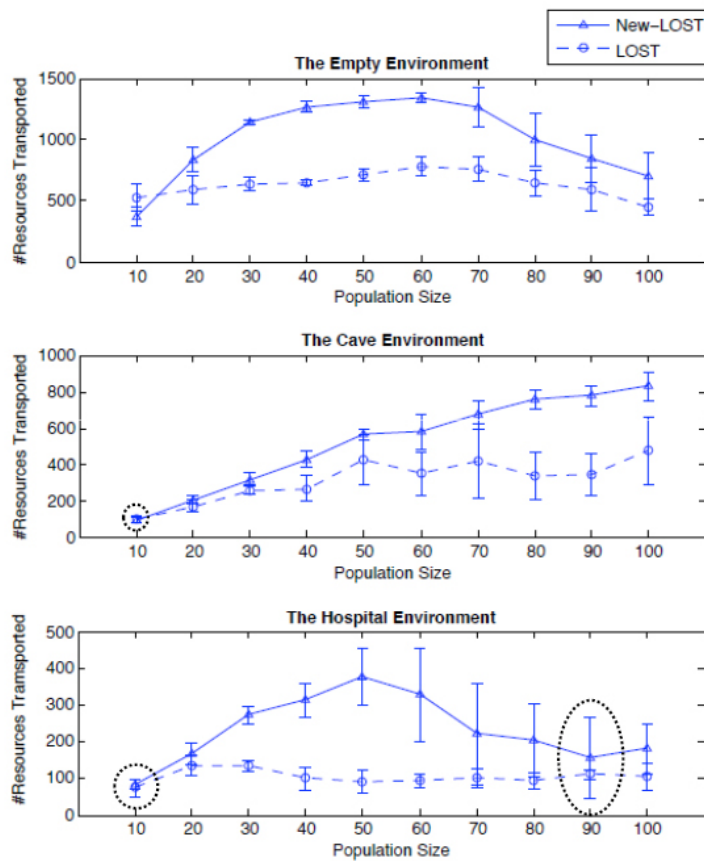


Figure 5.5: The result of the experiments in the 3 environments. The mean performance over 10 trials are shown with errorbars showing the standard deviation for both the original LOST and the new algorithm. The dotted line shows the data point for which the two algorithm do not show significant difference in distribution.

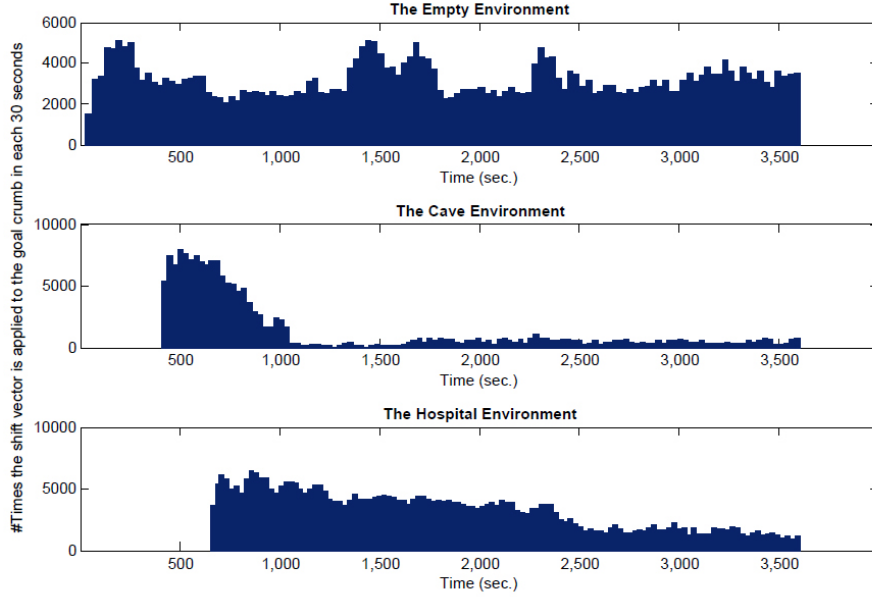


Figure 5.6: The histograms showing the number of times the goal crumb was shifted. The time bin is 30 seconds of simulation time. 30 robots are used in the empty environment and 50 robots are used in the other environments.

consequently robots do not approach the unattractive trails. The magnitude of the shift vector (*crumb_avoid*) determines the distance of the trails from each other and should be large enough to keep robots away from each other.

In order to see if the trails converge to a stable state we plotted the number of simulation cycles in which the shift vector was applied in each 30 *sec* of simulation time (Figure 5.6). In the *cave* and *hospital* environments, after the trails are formed they are gradually separated from each other due to the high use of shift vector. After some time, the trails come into a relatively stable state. The shift vector is still applied occasionally since the trails in some narrow parts of the environment (like doorways) are at their maximum distance from each other and can not go farther away. For the *empty* environment since the area is small and there is a short distance between source and sink, the robots tend to be pushed towards other trails which results in the high use of shift vector throughout the experiment.

We do not know of any biological system that uses a similar approach to reduce the destructive effects of interference among individuals, but still we believe that these techniques

can be used in systems inspired from animals and social insects to improve the efficiency of robots in performing a task.

5.4 Conclusion

In this Chapter we presented SO-LOST, a new navigation strategy to reduce interference in ant-inspired foraging-and-trail-following robot systems. The method makes use of the different trails formed in the environment to prevent robots with different goals from getting in each other's way. It is quantitatively evaluated through simulation experiments and shown to be effective in relatively constrained environments. Qualitatively, the screenshots of simulation experiments show that distinct separate trails with different goals were formed while keeping a distance from each other hence reducing the interference.

Chapter 6

Conclusion and Future Works

In this study, the general problem of using ant-inspired trail-following in multi-robot systems with relatively high number of robots was demonstrated. Investigating the scalability of ant-like trail-following, it was found that interference reduces performance in large populations. Then some strategies were suggested to improve the performance of a robot team that used trail-following for finding and sharing paths between task related locations. These methods aimed to tackle the interference problem while trying to maintain the attractive features of the ant algorithm.

It was shown (Chapter 3) that the amount of information about the crumbs (waypoints) that is available to the robots affects the performance of the team. By restricting the field of view of the robots' trail detecting sensor, robots form and maintain a number of paths, each with reduced congestion and thus, with low 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 fairness of the algorithm was also studied with selected FOVs and the superiority of narrow FOVs ($60 < \alpha < 120$ degree) was shown. The results of the simulation experiments confirm the improvement of the trail-following algorithm. This appealingly simple way of reducing interference is obtained only by changing the FOV of the robots' sensor and no modification of the original algorithm is needed.

Another approach independent from the first method was proposed in Chapter 4, focusing on the interference among robots following different trails. The trail-using algorithm was modified so that a robot following a trail to get to goal place G , interprets crumbs with goals other than G , as proxies for potentially interfering robots and avoids them by applying

a slight change in its direction. Ultimately, trails that were formed close to each other will be separated emergently decreasing the probability of interference. Experiments indicate that interference is reduced, performance increases, and that the trails come into a stable state after some time.

In future work we will implement the new algorithm on real robots and run experiments to verify our findings in simulation. Also, we will investigate methods of congestion resolution (as opposed to the congestion-avoidance methods discussed here) in trail-following robot systems. The algorithm presented in this paper is used to avoid congestion and conflicts between robots. However, there is plenty of room for improvement in mutual robot-robot avoidance methods, and development here would have an impact in many multi-robot systems.

The LOST and SO-LOST framework allows us to add various kinds of meta-data to the crumb and trail data structures. We expect that performance could be further improved by clever use of other meta-data embedded into crumbs, perhaps by gathering some global statistics. This would be unusual in ant-inspired systems, and perhaps powerful.

For now, we believe SO-LOST may be the most real-world practical trail-following algorithm yet described, since it explicitly manages the spatial interference that plagues real-world robots in any number.

Bibliography

- [1] Tucker Balch, Frank Dellaert, Adam Feldman, Andrew Guillory, Charles Isbell, Zia Khan, Stephen Pratt, Andrew Stein, and Hank Wilde. How multirobot systems research will accelerate our understanding of social animal behavior. *Proceedings of the IEEE*, 94(1):1445–1463, July 2006.
- [2] E.J. Barth. A dynamic programming approach to robotic swarm navigation using relay markers. In *American Control Conference, 2003. Proceedings of the 2003*, volume 6, pages 5264 – 5269 vol.6, 4-6 2003.
- [3] Sarah Brown, Mauricio Zuluaga, Yinan Zhang, and Richard Vaughan. Rational aggressive behaviour reduces interference in a mobile robot team. In *Proceedings of the International Conference on Advanced Robotics (ICAR)*, Seattle, Washington, July 2005.
- [4] Scott Camazine, Nigel R. Franks, James Sneyd, Eric Bonabeau, Jean-Louis Deneubourg, and Guy Theraula. *Self-Organization in Biological Systems*. Princeton University Press, Princeton, NJ, USA, 2001.
- [5] William Cohen. Adaptive mapping and navigation by teams of simple robots. *Journal of Robotics and Autonomous Systems*, 18:411–434, 1996.
- [6] Alberto Coloni, Marco Dorigo, and Vittorio Maniezzo. Distributed optimization by ant colonies. In *European Conference on Artificial Life*, pages 134–142, 1991.
- [7] Ro Crespí, Andr Badertscher, Andr Guignard, and Auke Jan Ijspeert. Amphibot i: an amphibious snake-like robot. *Robotics and Autonomous Systems*, 50:163–175, 2005.
- [8] Xinyan Deng, L. Schenato, Wei Chung Wu, and S.S. Sastry. Flapping flight for biomimetic robotic insects: part i-system modeling. *Robotics, IEEE Transactions on*, 22(4):776 –788, aug. 2006.
- [9] Marco Dorigo. *Optimization, Learning and Natural Algorithms*. PhD thesis, Politecnico di Milano, Italy, 1992.
- [10] Alexis Drogoul, Jacques Ferber, and Alexis Drogoul Jacques Ferber. From tom thumb to the dockers: Some experiments with foraging robots. In *In Proceedings of the Second*

- International Conference on Simulation of Adaptive Behavior*, pages 451–459. MIT Press, 1992.
- [11] 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)*, pages 1391–1396, Sept. 2008.
 - [12] Brian Hrotenok, Sean Luke, Keith Sullivan, and Christopher Vo. Collaborative foraging using beacons. 2010.
 - [13] C.J. Kleineidam, W. Ressler, B. Hlldobler, and F. Roces. Perceptual differences in trail-following leaf-cutting ants relate to body size. *Journal of Insect Physiology*, 53(12):1233 – 1241, 2007.
 - [14] L.S. Kuebler, C. Kelber, and C.J. Kleineidam. Distinct antennal lobe phenotypes in the leaf-cutting ant (*Paracryptus vollenweideri*). *The Journal of Comparative Neurology*, 518(3):352–365, 2010.
 - [15] D. Lachat, A. Crespi, and A. J. Ijspeert. BoxyBot: a swimming and crawling fish robot controlled by a central pattern generator. In *Proceedings of The first IEEE / RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob 2006)*, 2006.
 - [16] Adam Lein and Richard T. Vaughan. Adaptive multi-robot bucket brigade foraging. In *Proceedings of the Eleventh International Conference on Artificial Life (ALife XI)*, August 2008.
 - [17] Kristina Lerman and Aram Galstyan. Mathematical model of foraging in a group of robots: Effect of interference. *Auton. Robots*, 13(2):127–141, 2002.
 - [18] Jindong Liu, I. Dukes, and Huosheng Hu. Novel mechatronics design for a robotic fish. In *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 807 – 812, 2-6 2005.
 - [19] Marco Mamei and Franco Zambonelli. Physical deployment of digital pheromones through rfid technology. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 1353–1354, New York, NY, USA, 2005. ACM.
 - [20] Shervin Nouyan, Alexandre Campo, and Marco Dorigo. Path formation in a robot swarm – self-organized strategies to find your way home, 2008.
 - [21] Keith J. O’Hara and Tucker R. Balch. Distributed path planning for robots in dynamic environments using a pervasive embedded network. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1538–1539, Washington, DC, USA, 2004. IEEE Computer Society.

- [22] David Payton, Mike Daily, Regina Estowski, Mike Howard, and Craig Lee. Pheromone robotics. *Auton. Robots*, 11(3):319–324, 2001.
- [23] Anies Hannawati Purnamadajaja and R. Andrew Russell. Guiding robots’ behaviors using pheromone communication. *Auton. Robots*, 23(2):113–130, 2007.
- [24] Andrew Russell, David Thiel, Reimundo Deveza, and Alan Mackay-Sim. Sensing odour trails for mobile robot navigation. In *Proc. Int. Conf Robotics and Automation*, pages 2672–2677. IEEE, 1994.
- [25] Abbas Sadat and Richard T. Vaughan. So-lost: An ant-trail algorithm for multi-robot navigation with active interference reduction. In *Proceedings of the Twelfth International Conference on Artificial Life (ALife XII)*, August 2010.
- [26] S. Abbas Sadat and Richard T. Vaughan. Blinkered lost: Restricting sensor field of view can improve scalability in emergent multi-robot trail following. In *ICRA*, pages 947–952, 2010.
- [27] A. Scheidler, D. Merkle, and M. Middendorf. Congestion control in ant like moving agent systems. In *Biologically-Inspired Collaborative Computing*, volume 268, pages 33–43, 2008.
- [28] Miguel Schneider-Fontn and Maja J. Mataric. Territorial multi-robot task division. *IEEE Transactions on Robotics and Automation*, 14:815–822, 1998.
- [29] Richard T. Vaughan. Massively multi-robot simulations in stage. *Swarm Intelligence*, 2(2-4):189–208, 2008.
- [30] Richard T. Vaughan, Kasper Støy, Andrew Howard, Gaurav Sukhatme, and Maja J. Matarić. Lost: Localization-space trails for robot teams. *IEEE Transactions on Robotics and Autonomous Systems*, 18(5):796–812, 2002.
- [31] Richard T. Vaughan, Kasper Støy, Gaurav Sukhatme, and Maja J. Matarić. Exploiting task regularities to transform between reference frames in robot teams. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Washington, DC, May 2002.
- [32] Richard T. Vaughan, Kasper Støy, Gaurav S. Sukhatme, and Maja J. Matarić. Blazing a trail: insect-inspired resource transportation by a robot team. In *Proceedings of the International Symposium on Distributed Autonomous Robot Systems*, pages 111–120, Knoxville, Tennessee, USA, October 2000.
- [33] Richard T. Vaughan, Kasper Støy, Gaurav S. Sukhatme, and Maja J. Matarić. Go ahead, make my day: Robot conflict resolution by aggressive competition. In *Proceedings of the 6th International Conference on Simulation of Adaptive Behaviour (SAB)*, pages 491–500, Paris, France, August 2000.

- [34] Richard T. Vaughan, Kasper Støy, Gaurav S. Sukhatme, and Maja J. Mataric. Whistling in the dark: cooperative trail following in uncertain localization space. In *Proceedings of the Fourth International Conference on Autonomous Agents*, pages 187–194, Barcelona, Spain, June 2000.
- [35] Jens Wawerla and Richard T. Vaughan. Near-optimal mobile robot recharging with the rate-maximizing forager. In *Proceedings of the European Conference on Artificial Life (ECAL)*, pages 776–785, Lisbon, Portugal, September 2007.
- [36] Barbara Webb. Using robots to model animals: a cricket test. *Robotics and Autonomous Systems*, 16(2-4):117 – 134, 1995. Moving the Frontiers between Robotics and Biology.
- [37] Barbara Webb. What does robotics offer animal behaviour? *Animal Behaviour*, 60(5):545 – 558, 2000.
- [38] Keven Weber, Svetha Venkatesh, and Mandyam Srinivasan. Insect-inspired robotic homing. *Adapt. Behav.*, 7(1):65–97, 1999.
- [39] Alan F. T. Winfield and Julien Nembrini. Safety in numbers: fault-tolerance in robot swarms. *International Journal of Modelling, Identification and Control*, 1(1):30–37, January 2006.
- [40] Cornell Wright, Aaron Johnson, Aaron Peck, Zachary Mccord, Allison Naaktgeboren, Philip Gianfortoni, Manuel Gonzalez-rivero, Ross Hatton, and Howie Choset. Design of a modular snake robot.
- [41] V.A. Ziparo, A. Kleiner, B. Nebel, and D. Nardi. Rfid-based exploration for large robot teams. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 4606 –4613, 10-14 2007.
- [42] Jean-Christophe Zufferey. *Bio-inspired Flying Robots: Experimental Synthesis of Autonomous Indoor Flyers*. EPFL/CRC Press, Lausanne, 2008.
- [43] Mauricio Zuluaga and Richard 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.
- [44] Mauricio Zuluaga and Richard T. Vaughan. Modeling multi-robot interaction using generalized occupancy grids, with application to reducing spatial interference. In *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2008.
- [45] Esben H. stergaard, Gaurav S. Sukhatme, and Maja J. Mataric. Emergent bucket brigading - a simple mechanism for improving performance in multi-robot constrained-space foraging tasks. In *In Autonomous Agents*, pages 2219–2223, 2001.