Université Libre de Bruxelles
Faculté des Sciences Appliquées
CODE - Computers and Decision Engineering
IRIDIA - Institut de Recherches Interdisciplinaires
et de Développements en Intelligence Artificielle

Social Dynamics for an Exploration and Exploitation Task in Swarm Robotics.

Roman MILETITCH

Promoteur:
Prof. Marco DORIGO

Rapport d'avancement de recherche
Année Académique 2012/2013

IRIDIA
Institut de Recherches Interdisciplinaires
et de Développements en Intelligence Artificielle

Abstract

En Multi-robot exploration and navigation is a challenging task, especially within the swarm robotics domain, in which individual robots have limited capabilities and have access to local information only. An interesting approach to exploration and navigation in swarm robotics is *social odometry*, that is, a cooperative strategy in which robots exploit odometry for individual navigation, and share their own position estimation through peer-to-peer local communication to collectively reduce the estimation error. In our research, the robots have to localize both a home and resources of various quality. They then forage from the later as they navigate back and forth between resources and nest. The way in which the resources location information is aggregated influences both the efficiency in navigation/exploitation between the two areas, and the self-organized selection of better paths. We propose three new parameter-free mechanisms for information aggregation and we provide an extensive study to ascertain their properties in terms of navigation efficiency and collective decision.

Fr En robotique, l'exploration collaborative d'une zone et sa navigation est une tâche difficile, en particulier dans le domaine de la robotique essaim, dans lequel chaque robot possède des capacités limitées et ne perçoit que localement son environnement. L'*odométrie sociale* s'avère être une approche intéressante concernant l'exploration et la navigation, plus particulièrement au sein d'un essaim robotique. Il s'agit d'une stratégie de coopération dans laquelle les robots utilisent l'odométrie afin d'avoir une indication sur leur déplacement personnel, et partagent cette estimation avec les robots environnants dans un contexte pair-à-pair, réduisant ainsi l'erreur d'estimation de leur position. Dans la recherche présentée, les robots doivent localiser deux types de zone, un nid et des zones de ressources de qualité variée. Leur objectif est alors de naviguer entre ces deux zones afin de retourner au nid en possession d'objets (symbolique). La manière dont les informations de localisation des ressources sont rassemblées par l'essaim influence à la fois l'efficacité de la navigation/exploitation et la sélection autoorganisée des meilleurs chemins. Nous présentons ici trois nouveaux mécanismes dénués de paramètres, définissant la manière dont cette information est partagée au sein de l'essaim. Chacun de ces mécanismes est étudié afin de déterminer ses propriétés en termes d'efficacité pour la navigation, l'exploitation de ressources et la capacité des robots à prendre une décision collective.

Contents

Introduction

The goal of the proposed research is to develop a cooperative exploration and resource exploitation strategy based on a peer-to-peer exchange of information between robots in a swarm, and to understand its resulting dynamics. The associated task is characterised by valuable resources spread in the environment that have to be found and exploited by a robotic swarm. In such a task, the environment is usually not completely known in advance, forcing the robots to explore it to discover the position of the resources. Moreover, the environment is also dynamic since resources may get depleted as robots accomplish their work. When multiple robots are sharing such a task, allocating them efficiently and dynamically to the available evolving resources is not a trivial task. In this case, cooperative strategies can be used to improve both exploitation efficiency and adaptability. This is particularly useful in the swarm robotics domain in which individual robots cannot rely on global information or complex algorithms. In this work, we propose information aggregation mechanisms that allow the swarm to adapt to the evolution of the resources in the environment despite individual robots keeping the same individual behaviour. We then study the resulting dynamics of the swarm, both in simulation and with real robots.

The general experimental conditions mentioned above fit various applications, from search & rescue operations where robots explore disaster areas and bring victims back to a safe place, to mining where robots explore large potentially dangerous areas (mines, deep water excavations) and are expected to be mining for valuable resources or bringing back specific items. What these applications have in common is an environment with a set of areas containing resources that robots need to reach and a *home* location where the robots gather back. We created an experimental setup with robots capable of locating, grabbing and securing items spread in an arena, which realistically models and isolates various aspects of such exploration/exploitation tasks.

In these scenarios, two tasks are interlinked: exploration/navigation and exploitation. First the robots have to locate, reach, and navigate efficiently back and forth between target areas; second they have to exploit the resources depending on their quality. The presented research relies on experimental scenario to study first the navigation issue, and then with regard to these results, the exploitation task.

Exploration and navigation strategies in swarm robotics should present a low complexity to match the limited capabilities of the individual robots, which is why simple dead-reckoning techniques such as odometry are favoured. Being quite error prone, there is a need to reduce the estimation errors that robots accumulate over travelled distance. This error can be reduced through the shared effort of multiple robots exchanging structured information (Martinelli et al., 2005). By sharing the estimated position of a landmark, the robots can collectively reduce the overall odometric error. This is a straightforward mechanism that easily lends itself to implementation on very simple robots. Therefore, the collective reduction of odometry errors can be instantiated also in swarm robotics contexts, as it complies with the inherent limitations of the robots.

This mechanism was first introduced by Gutiérrez et al. (2009) and is referred to as *social odometry*. In this approach, the robots estimate the navigation path between two target areas in the environment (*i.e.*, home and goal locations) using odometry and attach to this estimate a confidence level that decreases with the distance travelled. At the same time, the robots share their navigation information within the swarm in a local peer-to-peer manner. Thanks to this process, information about target areas spreads gradually within the swarm, helping reduce the error in the position estimation. Overall, this decentralized process results in an increased efficiency in the swarm navigation abilities.

An interesting aspect of social odometry is that it naturally leads to the emergence of collective decisions within the swarm (Gutiérrez et al., 2010). Indeed, when there are multiple goal areas to localize (*e.g.*, multiple resources to exploit), by sharing the available information the robots not only improve the accuracy of their localization but can also decide which area to target. The sum of individual decisions leads to a self-organized behaviour that makes the swarm choose between focusing on a single area/resource or exploiting in parallel several ones.

In the navigation studies, the only variable of the arena setup that impacts the decision of the swarm is the distance of the resources to the nest. This condition changes in the second part of our study in which we focus on the exploitation case. In this case, the resources are defined both by their distance from home and by their quality (*e.g.*, rate of regeneration and size of a resource). When the resources vary in quality, they are not only valued based on their distance from home but also on the ease in finding/processing items from them. In this case, the swarm must adapt to the dynamics of the environment and find a balance between exploiting close resources which are easier to reach or farther resources that might be of better quality. In doing so, the swarm must continuously choose between focusing on one single goal or splitting among many.

The efficiency of social odometry as a navigation and resource exploitation mechanism and the resulting collective dynamics of decision-making depend heavily on the way information is shared and aggregated in the robot swarm. In particular, we found that even small variations in some parameters of the individual behaviour may lead to huge differences in the swarm dynamics. For this reason, in this work we propose three new parameter-free mechanisms for information aggregation and processing that make the swarm adapt to resource allocation in the environment.

We expect that our information aggregation mechanisms will not only find the best split among resources at a given time, but will also react to the upcoming variations in quality as the resources get depleted and hence continuously exploit the environment in an efficient way. For that, we expect a fine balance between exploration and exploitation so that the swarm can react quickly to variations while keeping a steady pace of exploitation.

Following this introduction and after describing the state of the art in both navigation and communication in swarm robotics, we describe *social odometry* and the three information processing mechanisms we have devised. The two following chapters present the experimental setups and results for respectively the navigation experimentations, and the exploitation experimentations. We then discuss the obtained results and conclude with some final remarks.

Chapter 1

State of the Art

Exploiting from resources in an environment is not a new topic in robotics, would that be for simple robots or in the context of swarm robotics. When robots are faced with this high level objective, various lower level tasks arise, would that be how to explore an environment, how to self-localise and/or the targeted areas, how to communicate and reach a consensus with other robots, or how to exploit resources. Before presenting our work in details, this chapter presents a few way to tackle these issues.

1.1 Navigation for Single Robots

The simplest way for a robot to explore and navigate in a closed area is through random walk. While not being the most efficient way, it assures that the robots reach every part of the environment, even if this may require a long time. In order to improve on a purely random exploration, the robots can memorize and map their surroundings to avoid previously explored zones (Thrun, 2008) to reach specific areas of interest. For this purpose, each robot can position itself on the map and navigate in an environment using dead-reckoning techniques such as odometry. Odometry relies on the integration over time of the movement vector—as perceived through the robots' (proprioceptive) sensors—, in order to maintain an estimate of the robots' position. However, this approach is quite error prone since estimation errors are cumulated over time, therefore requiring techniques for error reduction such as Kalman filters (Thrun et al., 2005).

1.2 Navigation in Swarm Robotics

There are various ways to improve navigation through information-sharing within a swarm (Martinelli et al., 2005). Ducatelle et al. (2011) model a swarm as a communication network that propagates relevant information. Each robot in the swarm maintains a table with navigation information about all known robots, similar to how nodes in a mobile ad hoc network maintain routing tables. Then, the robots propagate the available information and use the table to find the best path to reach a target robot within the swarm. Sperati et al. (2011) also study navigation in a swarm robotics context. In this case, communication is performed through visual signals only and therefore the information exchanged is much less structured. For this reason, they used artificial evolution to synthesize effective navigation strategies.

Several studies in swarm robotics implement navigation and exploration algorithms without sharing structured information, sometimes exploiting robots as physical landmarks. Rekleitis et al. (2001) divided the swarm into two teams, one moving and the other stationary, serving as a reference for navigation. The teams alternate between stationary and moving states. Nouyan et al. (2009) exploit robots to form complex structures such as chains, in which one end of the chain connects to a central place while the other end explores the environment. Once the goal location is reached, the chain can be exploited by other robots for navigation purposes, or a bucket brigade method can be used to transport objects along the chain (Ostergaard et al., 2001).

1.3 Collective Decisions

When there are several goal/resource locations present in the environment, the robots may make a collective decision and focus on the exploitation of a single one. This can be beneficial if it is necessary to aggregate a sufficient number of robots in support of collective localization, or if exploitation requires several robots at the resource. However, this may lead to congestion (*i.e.*, the path to the resource is overused and the robots have trouble navigating) or overexploitation of the resource. In this case, the swarm is better off exploiting several resources in parallel.

In order to agree on one option, the robots can either switch to the best option available in their neighbourhood, or average out all the available information. Social odometry allows doing both simply by tuning a single parameter (Gutiérrez et al., 2010). Olfati-Saber et al. (2007) study the swarm as a multi-agent network and present a theoretical framework for analysing consensus algorithms. It is also possible to obtain collective decisions by amplifying the various opinions present in the swarm. Following this approach, the more an opinion is represented in the swarm, the higher the probability of disagreeing robots switching their opinion (Garnier et al., 2007, 2009; Montes de Oca et al., 2011). This approach requires gathering the opinion of several neighbours, while social odometry works with peer-to-peer interactions, which is easier to implement.

1.4 Exploitation of Resources

Tasks in which the robots have to exploit resources are classic test-bed applications in swarm robotics (Winfield, 2009) in which collective behaviours are studied and compared. In this application, robots have to retrieve items (preys) spread around the environment or in specific goal areas (resources) and to bring them back to a specific location (nest). Such exploitation patterns are often found in biological systems. Among others species, ants display complex foraging behaviours through which they are successfully able to adapt to a dynamic environment and retrieve preys (Camazine, 2003).

In order to more efficiently locate the various resources and to optimally forage from them, Gutiérrez et al. (2010) developed a strategy based on consensus reaching, on which we base our present work. In this strategy, robots locally interact by sharing information on the position of resources, aggregate this information and decide which resource to forage from based on their respective distance to there current position. When robots forage from the same resource, interferences arise as congestion builds up. Rybski et al. (2007) showed in their work that the introduction of communication in real foraging experiments does not always increase the performance of the system because of an increase in interference. In order to tackle this problem and reduce interferences among robots, task allocation methods can be used (Campo and Dorigo, 2007; Liu et al., 2007). Shell and Mataric (2006) use a bucket brigade method in which the robots do not directly travel back and forth between the preys and the nest but instead have a limited radius of operation.

While not using task allocation to limit the effect of interferences, we studied the impact of these interferences over our various social mechanisms in order to find optimal behaviours for different setups.

Chapter 2

Social odometry

In our experiments, the goal of the robots is to locate both a home area and a goal/resource area and then to efficiently navigate/forage between them. In all our scenarios and experiments, once a target area is discovered, its position is kept in memory and updated using odometry.

The information about target areas is shared with other robots upon encounter, following the social odometry principle. The way in which the information exchanged is shared and processed is independent from the individual behaviour of the robots, which is different in the two scenarios we present in this report. Therefore, we start by introducing the information processing mechanisms we have devised.

2.1 Information Sharing

While robots navigate between target areas, they share the information they have on the relative locations in order to counterbalance the dropoff in information confidence. How and when this information is shared has a strong influence on the overall quality of the information in the swarm, and on its decision-making. Not all information is shared at the same time. When randomly exploring, the robots share the sole information they have. In the other cases, the robots share only the information of the last visited location.

Given that robots do not share a global coordinates system or a common reference frame, a transformation of the shared position is needed in order to fit the frame of the receiving robot (Gutiérrez et al., 2009). To that end, the robots use as a reference the communication axis defined by the usage of their range&bearing device. This transformation is presented in Fig. 2.1 for two robots i and j , j receiving a message from i.

For that transformation to be possible, robot i first needs to know the direction of robot j. To make it happen, our communication protocol follows a ping-pong method in which robots constantly broadcast their needs (nest or resource) while the other robots in range answer back with relevant information only when they possess it. When receiving calls for information, the robots not only come to know which information they should share but also the direction of the robot (γ^i).

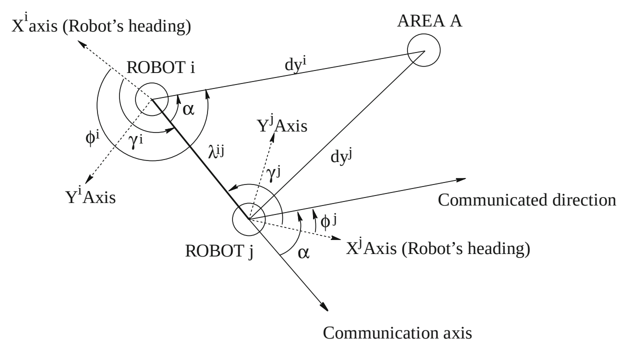

Figure 2.1: Diagram of the transformation of the shared position of area A between the frame of reference of robot i (emitting) and robot j (receiving), reprinted from Gutiérrez et al. (2009)

Once robot i receives a call for information from robot j, it shares back the distance (dy^i) to the relative area as well as its direction (α), the later in the new frame of reference defined by the axis of communication: $\alpha = \phi^i - \gamma^i$

Now j needs to transforms the received data in its own frame of reference. For that it must first find the communicated direction of robot i: $\phi^j = \gamma^j + \alpha - \pi$. It can then calculate the position of the target area in its own coordinate system:

$$dy^i_x = \lambda^{ij} \cdot cos(\gamma^j) + dy^i \cdot cos(\phi^j)$$

$$dy^i_y = \lambda^{ij} \cdot sin(\gamma^j) + dy^i \cdot cos(\phi^j)$$

with λ^{ij} being the distance between two robots, provided by the range&bearing device.

2.2 Information Processing

Once the information is received by robot i, it is aggregated with the robot's own knowledge. The way this aggregation is performed depends on the information processing mechanism implemented. Let $\mathbf{p}_i$, $\mathbf{p}_j$ be the estimated position of an area (either home or goal) for robots i and j, and c_i, c_j be the confidence over their respective estimation. The result of any aggregation is the updated couple $\langle \mathbf{p}_i, c_i \rangle$.

Here, we first describe the information aggregation mechanism used by Gutiérrez et al. (2009), and then we introduce our contributed mechanisms.

Fermi distribution The aggregation mechanism used by Gutiérrez et al. (2009) is based on a Fermi distribution. A weight is calculated from the difference in confidence in order to make a linear combination of the positions:

$$\langle \mathbf{p}_j, c_j \rangle \leftarrow k \cdot \langle \mathbf{p}_j, c_j \rangle + (1 - k) \cdot \langle \mathbf{p}_i, c_i \rangle$$

$$k = \frac{1}{1 + e^{-\beta(c_j - c_i)}}$$

The parameter β measures the importance of the relative confidence levels in the information aggregation. For low values, the aggregation is close to an average, ignoring the confidence. For higher values, the aggregation is stiff: only the information with highest confidence is kept.

Finding the right value of β is often a process of trial and error. Our contribution in this paper is the introduction of three parameter-free aggregation mechanisms: *Hard Switch* (*HS*), *Random Switch* (*RS*) and *Weighted Average* (*WA*).

Hard Switch (*HS*) In this winner-take-all mechanism, the robots keep the information with highest confidence (either the current information or the received one) and discard the other one. This mimics the Fermi mechanism with a high β.

$$\langle \mathbf{p}_j, c_j \rangle \leftarrow \langle \mathbf{p}_x, c_x \rangle, \qquad x = \operatorname*{arg\,max}_{k \in \{i,j\}} c_k$$

Random Switch (*RS*) As in the mechanism above, here the robots keep one piece of information and discard the other. In this case, however, the switch is stochastic: the higher the confidence, the higher the probability of accepting the information. In practice, this mechanism is a stochastic version of the *HS*.

$$P(\langle \mathbf{p}_j, c_j \rangle \leftarrow \langle \mathbf{p}_i, c_i \rangle) = \frac{c_i}{c_j + c_i}$$

Weighted Average (*WA*) This mechanism consists in a linear combination of both estimated positions with their confidence as weight. On the one hand this implies no loss of information; on the other, when information about different goals is aggregated, the new position may not coincide with a real goal location, leading to the apparition of artefacts. While the Fermi mechanism focuses on the difference between the two confidences, here we directly use each of them as weights.

$$\langle \mathbf{p}_j, c_j \rangle \leftarrow \left\langle \frac{c_j \cdot \mathbf{p}_j + c_i \cdot \mathbf{p}_i}{c_j + c_i}, \frac{c_j + c_i}{2} \right\rangle$$

Chapter 3

Navigation Task

In this chapter, we focus on the navigation ability of the swarm as supported by the social odometry navigation mechanism. To this purpose, robots have to navigate between target areas represented as grey circles painted on the ground. We will study the influence that the different information aggregation mechanisms described in Chapter 3 have on navigation efficiency and collective decisions.

First of all, we introduce the individual behaviour of the robots. Then, we introduce the experimental setup and finally we discuss the obtained results.

3.1 Individual Behaviour

The behaviour of the robot is defined by a finite state automaton with five states: *Explore*, *Go Home*, *Go to Goal*, *Leave Home*, *Leave Goal* (Fig. 3.1). Robots start in the *Explore* state and return to it whenever they lack relevant information. The other four states form a loop that corresponds to the robot navigating back and forth between the target areas: go to a target area, enter and leave it, then go to the next one. On top of these control states, both short and long range collision avoidance are implemented.

The robots start without any prior knowledge about the location of the target areas. Therefore, they first have to explore the arena. When in the *Explore* state, the robots perform a random walk until they discover the position of both target areas (home and goal). This can happen in two ways: either they receive relevant information from team-mates or they stumble upon a target location (*Got(Area)* becomes true, with $Area \in \{Home, Goal\}$). In both the *Go to Goal* and *Go Home* states, the robots move straight to the target location, possibly avoiding other robots and obstacles. Along their way, they update the target areas location using odometry and update their confidence in the information. The confidence is defined as the inverse of the distance that the robot had travelled from the target area. Therefore, a straight path results in a higher confidence than a curved one.

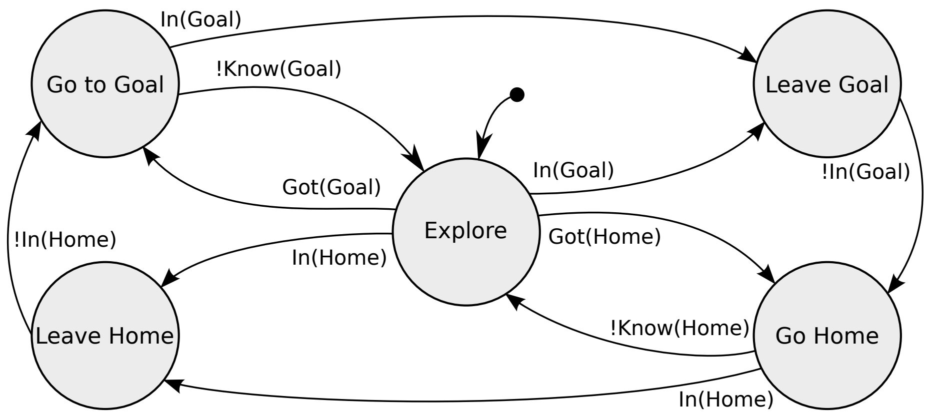

Figure 3.1: Robot's finite-state automaton. The circles define the states while the arrows define the transitions. *In(Area)*, *Area* $\in \{Home, Goal\}$, is true when the robot senses the grey level of the area, *Know(Area)* is true when the robot knows the position of the area, *Got(Area)* is true when it just gets this estimation. The robots start in the *Explore* state.

Once a robot reaches an area (*i.e.*, *In(Area)* is true), it traverses it in a straight line (possibly dodging other robots to avoid collisions) and stores the area location. In order to get an estimated position closer to the center of the area, the robot averages its entering and its exiting positions. No matter how many goals there are in the arena, the robots always memorize only one home and one goal (the last seen or agreed upon).

3.2 Experiments

We used an experimental setup with as few variables as possible: a circular arena (radius: 11 m) with the home in the center and the goals scattered around (Fig. 3.2). The goals are defined by their distance from home (d_i) and the angle between each other ($\alpha_{ij} \in [\pi/3, \pi]$). Both goal and home are of a radius of 50 cm, and are differently coloured in grey levels to be distinguished by the robots.

Our experiments are performed in the ARGoS open source multi-robot simulator (Pinciroli et al., 2012) and the robots we use are the marXbots (Bonani et al., 2010). To accomplish their task, the robots are equipped with several sensorimotor and communication devices. In our experiments, the robots use the infrared ground sensors to check whether they entered an area and to detect its type (home or goal) depending on the area's grey level. They also use the infrared proximity sensors for short range collision avoidance and the range&bearing device for both communication and long range collision avoidance among robots (Bonani et al., 2010). This last device gives both the angle

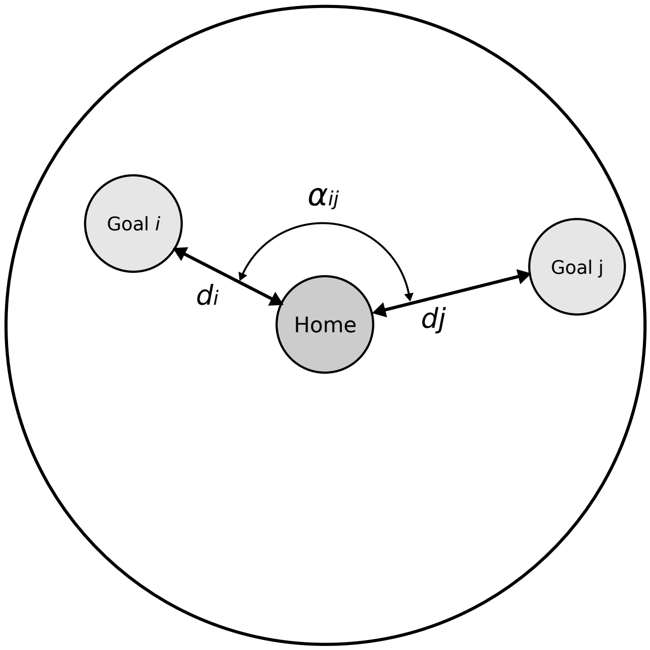

Figure 3.2: Setup of the experimental arena. The home area is placed in the center of a circular arena of an 11 m radius, surrounded by walls. The goals are characterised by their distance to the home d_i, d_j and the angles they form with each other α_{ij}.

and distance between neighbouring robots and allows them to send short messages. Wheel encoders provide the movement vector for odometry. A simulated gaussian noise with a 5% standard deviation models the odometry estimation error. The control loop is executed 10 times per second. Unless stated otherwise, we used 75 robots spawned randomly.

By varying the number of goals, we study different aspects of the collective behaviour, such as the impact of the density of robots on their navigation abilities, the collective decision made by the swarm in a two goals setup, and how this generalizes in multiple goals setups. In the following, we briefly describe the experiments we present in this paper.

Single Goal When a single goal is present, we expect that all robots will converge on the same path. The more robots in the arena, the harder it is for them to avoid each other. As density rises, the robots have to handle more and more congestion on their path, which leads them to travel bigger distances and to accumulate more error. This also corresponds to fewer round trips between the home and goal, hence lowering the efficiency of the swarm. We define the density on a path as the number of robots on it divided by its length.

In order to study the impact of density on navigation, we devised an experimental setup in which we vary both the distance between the home and goal and the number of robots. All three information processing mechanisms are tested and compared with a benchmark condition in which the robots are provided with perfect information (PI) about the goal and home locations. In each experiment, we measure the navigation speed, computed as the number of round trips over time. We study its evolution for values of density between 2 and 40 robot/m. For each density value, we run 100 trials in which we randomly draw the distance between the home and goal in the interval [3,8] m, and we compute the corresponding number of robots to obtain the specified density value (which is in the range [6,320]).

Two Goals When there is more than one goal, a decision has to be made about how to spread the robots among the available paths. In this setup, we study if and how the robots converge on a single path as well as the implications of such a convergence on efficiency. In order to study this decision-making process, we count the number of robots committed to each goal, as well as the uncommitted ones. Given that robots do not distinguish between different goals and only store one estimated position $\mathbf{p}_g$, a robot is considered to be committed to a goal i among n possible if it has information about both goal ($c_g \neq 0$) and home ($c_h \neq 0$), and if goal i is the closest one to the robot's estimated goal position $\mathbf{p}_g$.

In this setup, we have two goals which can either be at a short distance (5 m) or a long distance (8 m). We run experiments with both equal and different distances for the goals: Short/Short (SS), Short/Long (SL) and Long/Long (LL). For each condition, we perform 1000 replications by randomly varying the angle between the sources with $\alpha_{ij} \in [\pi/3, \pi]$ (cf. Fig. 3.2).

Multiple Goals The environment in which a swarm evolves is rarely as simple as in the two goals setup. Through a multiple goals setup, we enquire about the scalability of the previously gathered results. M goals are uniformly distributed around the home location, with an angular separation between adjacent goals of π/M, where $M \in [3, 6]$. To investigate both the navigation and the decision-making abilities, we test three different conditions. Either all goals are at the same distance, short (SSS) or long (LLL), or a single goal is closer to home (SLL). For each condition, we performed 250 trials.

3.3 Results

Each trial in all the previous setups lasts 20 minutes of simulated time. We use the same random initialization in all the runs for the different opinion processing. For each run we compute the number of robots on each path to study the dynamics of collective decisions, the number of round trips to study the navigation efficiency and the error made by the robots on the estimated position of the nest to gauge the quality of information in the swarm

3.3.1 Congestion

As we can see in Fig. 3.3, all the proposed mechanisms and the control condition with perfect information (*PI*) follow the same tendency. For low densities, we can observe a linear increase in the number of round trips. With higher densities, the growth slows down. As expected, robots with perfect information are the most efficient at first, but their efficiency reaches a peak because of the artefacts created by perfect information. With *PI*, since all robots aim for the center of the target areas (either home or goal), as the density rises they have increased difficulties avoiding collisions and entering or exiting the target areas.

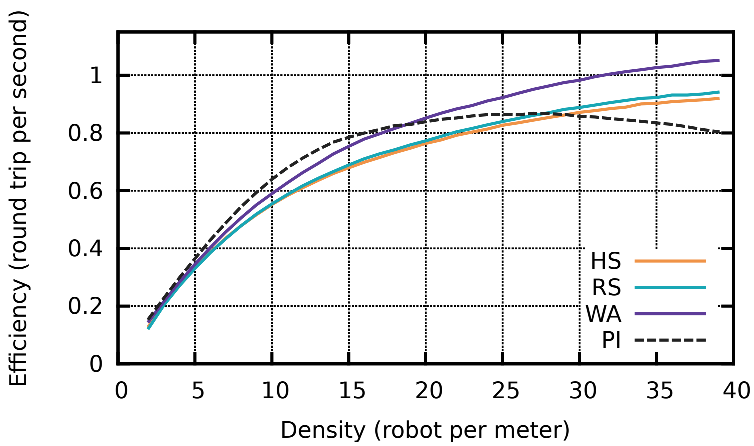

Figure 3.3: Impact of density on navigation efficiency for each mechanism and in the perfect information control condition. Each line is the mean over 100 trials.

Congestion has a lower impact on navigation efficiency with social odometry. In this case, *WA* proves to be more resilient to congestion than *HS* and *RS*. This is due to a smoother navigation in the surrounding of the home and goals, where robots try to enter small and densely populated area. First, since the *WA* mechanism never discards information but averages it, the precision on the estimated position is better than with *HS* or *RS*. Second, the reception of even slightly better information is smoothly integrated in the *WA* mechanisms resulting in better average information (Fig. 3.4), while in both *HS* and *RS* it may cause a large leap of the new location, which may be difficult to reach in case of high densities. Contrary to what could be expected, the quality of such information does not rise with the density of robots. Once there are enough robots to manage a steady connexion between locations, the quality of information is virtually at its best. As the number of robots rises, congestion creates issues for them to reach each location, implying longer travelling distances and hence worse information kept in memory, despite enhanced communication relying on a denser net of robots.

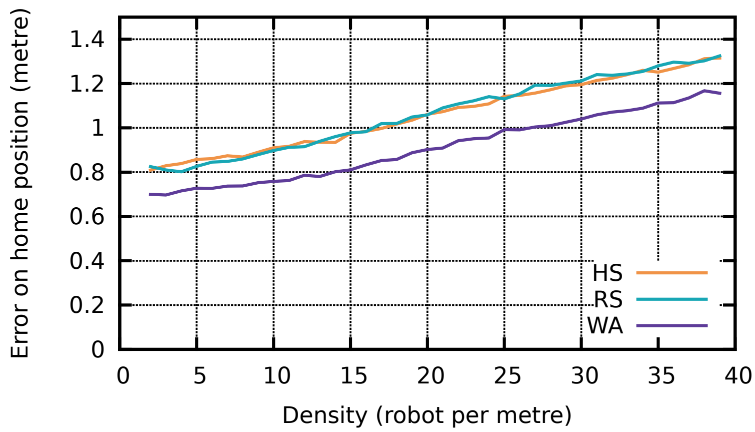

Figure 3.4: Evolution of the error of the estimated position of the centre of the nest for each mechanisms. Each line is a mean over 100 trials.

3.3.2 Collective Decision

Congestion explains why sometimes it is better to spread along multiple paths when there is more than one goal/resource. This decisions impacts not only the efficiency but also the spatial arrangement of the swarm and the way it reacts to changes in the environment.

Decision

The decision pattern of the swarm results from the sum of local decisions made by the robots. The dynamics of the collective decision are shown in Fig. 3.5, which plots the convergence pattern generated by the *HS* and *WA* mechanisms when confronted with the SL experimental condition. In all cases, the swarm decides to focus on the closest area/resource and most robots converge on the associated path. This behaviour is typical of all three social mechanisms when there is a goal closer to home. We can already see a strong difference between the two mechanisms, where *HS* converges quicker, with less variations among experiments.

We can observe three different phases. At first (0-120 s), most robots are uncommitted and explore for goal areas, reinforcing each as they discover them. Then (120-400 s), a competition among the two alternative paths occurs. The shorter path is reinforced more because of the improved information the robots have when encountering robots coming from the other goal. Eventually, the swarm enters a maximization state in which mostly one path is exploited while uncommitted robots continue to join.

Fig. 3.6 shows the percentage of robots that choose path A (*i.e.*, the shortest path in the *SL* condition). We note that in the *SL* case, all information aggregation mechanisms lead to a single path convergence of at least 90% of the robots. Both *HS* and *RS* always lead to a convergence on the closest goal. The same is the case for *WA*, which, however, also presents a low probability for the robots to converge on the distant goal. This happens because with *WA* no information is discarded. When a large number of robots discovers the distant goal early in the experiment, they may influence the whole swarm despite the lower confidence of their information. This cannot happen with *HS* and *RS*, because low quality information is instantly discarded. In both the *SS* and *LL* experimental conditions, when there is no better choice, *HS* and *RS* lead to a split in the swarm, and robots spread among the two paths (Fig. 3.6). In these experimental conditions, the more robots on a path, the higher the congestion, and the larger the distance the robots travel. This causes robots to have worse confidence in their information with respect to these from a less congested path. Therefore, switches to the other path are very likely. Congestion creates a sort of negative feedback that leads to an oscillating dynamic in which no decision ends up being taken. On the contrary, *WA* is not affected by such negative feedback and systematically leads to convergence (randomly on either path, the setup being symmetrical). Indeed, the poorr confidence that results from congestion is counterbalanced by the larger number of robots with which the information is shared and averaged. Therefore, the swarm converges to the more populated path.

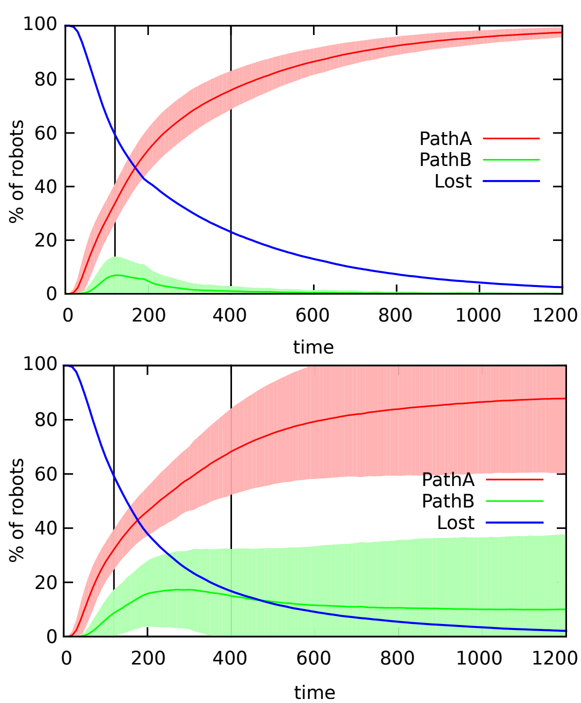

Figure 3.5: Evolution of the robots' repartition between the two target areas using *Hard Switch* (top) and *Weighted Average* (bottom) in the Short/Long condition. Bold lines indicate the mean over 1000 repetitions, and the shaded areas indicate the standard deviation. These two figures present the two extremes in convergence pattern in case of the existence of a shorter path.

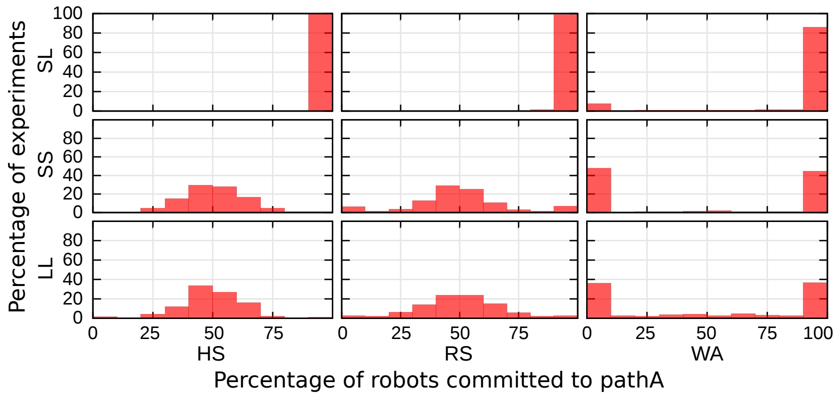

Figure 3.6: Robots repartition on path A. Each histogram shows the observed frequencies of the number of robots committed to path A (the shortest possible path).

Efficiency

The robot behaviour does not explicitly encode the ability to make collective decisions. Instead, it is conceived to provide efficient navigation ability thanks to the information shared within the swarm. The decision process is an emergent result of this behaviour and so is the variation in efficiency depending on the setup and the mechanisms involved, as shown in Fig. 3.7. In the *SL* condition, all three mechanisms make the robots converge on the closest path, therefore resulting in density of 15 robot/m. As shown in Fig. 3.3, *WA* is more resilient to congestion, which is why it is the most efficient mechanism in this setup, followed by *RS* and *HS*. In the *SS* condition, both *HS* and *RS* result in the swarm splitting between the two paths as discussed above. By exploiting two paths with a low density of 7.5 robot/m (instead of one with high density of 15 robot/m) the robots create less congestion, which explains why the performance for *HS* and *RS* is slightly better than in the *WA* case. Indeed, *WA* makes the swarm converge on a single path with a high density, and navigation is slightly less efficient. Congestion has a lower impact in the *LL* conditions as both densities (9.4 robot/m on a single path, 4.7 robot/m on two paths) fall in the linear part of the congestion curve (see Fig. 3.3), explaining why the mechanisms result in the same efficiency.

Switching Patterns

Social odometry and the various mechanisms studied above not only influence the efficiency of navigation and decision-making, but also the physical shape of the swarm. This can be seen not only by studying the movement of the robots, but by focusing on their switching patterns. A robot switches from one resource to another when it encounters better information that directs it to

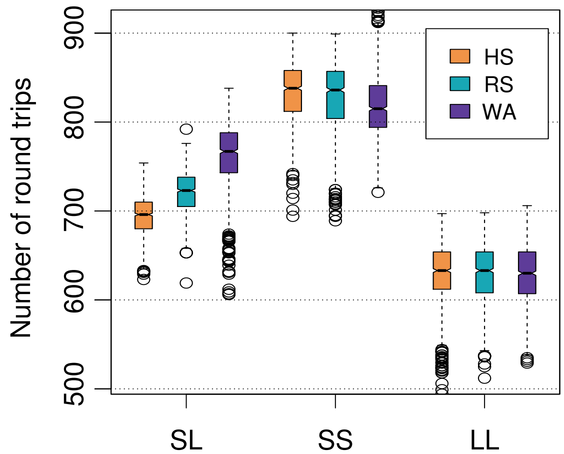

Figure 3.7: Efficiency of the swarm for two goals, for all mechanisms and conditions. Each box represents the inter-quartile range, whiskers extend to 1.5 times the corresponding quartiles, and the dots represent outliers.

another resource. The switching patterns for each mechanism are displayed in Fig. 3.4, in case of both resources being at the same distance, or in the presence of a closer resource.

First, we note that most of the switches occur in or near the nest. This is because the nest is the destination that all robots have in common, no matter their choice of resource. This is where the density of information, and even more its variety, is at its highest. Furthermore, as mentioned above, robots do not share and request both pieces of information (the nest's and the resource's position) at the same time. In order to switch from one resource to another, a robot has first to enter the nest to request new directions. The halo of switches around the nest is the result of the range of communication allowed by the range&bearing device. Its shape varies for different arena setup (for instance more centred when the resources are on each side of the nest).

All mechanisms do not show the same pattern of switches. For instance, when there is a better solution, the *HS* mechanism only needs a few switches for all the robots to converge on the closest resource. On the contrary, *RS* and *WA* sport a much higher number of switches. Both observations are coherent with the speed of each mechanism's convergence. When no closer resource is present, all mechanisms present a high number of switches, as robots oscillate between one possible solution and another. *WA* converges as in the previous condition, but with a higher number of switches.

In both *LL* and *SL* conditions, the switches pattern displayed have the tendency to grow toward the barycentre of both resource. This effect is even stronger in the case of *WA* because of its averaging aggregation of information. This leads to the creation of a trail of switches, in which they are no longer the result of a communication with the nest, but with either path connecting the resources.

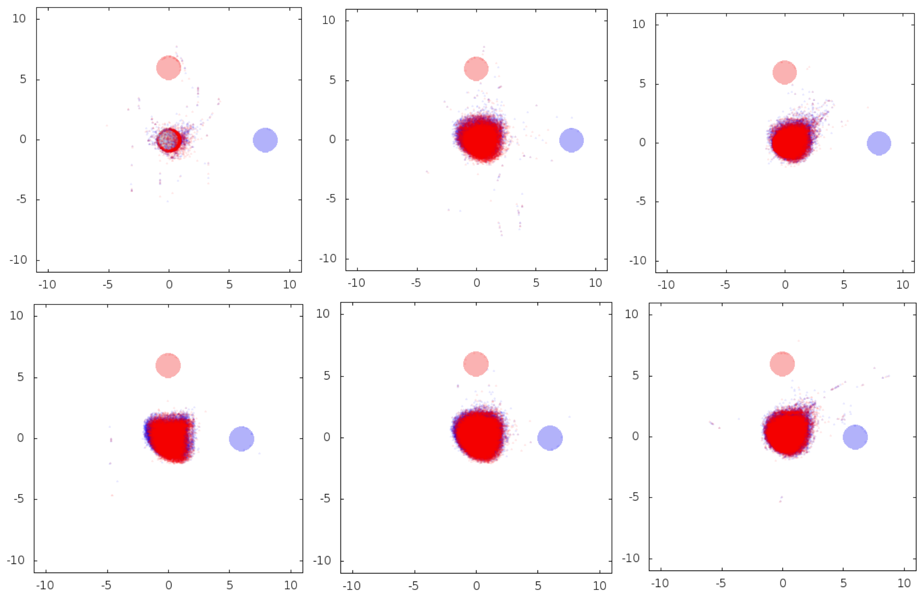

Figure 3.8: Cloud dot of the positions of the robots' switches from one resource to another. In red are the switches to the red resource (north, always the closest to the nest) and in blue the switches to the blue resource. Red switches are drawn on top of the blue switches. Top: SL setup condition, bottom: LL setup condition - in both case the resources form a 90° angle. From left to right: *Hard Switch*, *Random Switch* and *Weighted Average*.

3.3.3 Generalization to Multiple Goals

The dynamics we observe with multiple goal locations are similar to the ones displayed in the two goals setup, no matter the number of added goals. Fig. 3.10 shows the percentage of robots that choose path A (*i.e.*, the shortest path in the *SLL* condition), when multiple goal locations are present. All mechanisms leads to convergence in the *SLL* case, even if *WA* sometimes leads to the selection of one of the distant goals, for the same reasons discussed in the two goals setup. We can observe a similar splitting behaviour in the *SSS* and *LLL* conditions for both *HS* and *RS*, while convergence is observed for *WA*. When the swarm splits, the repartition of robots is no longer centred on 50% but is closer to 33%, implying that the repartition is no longer between only two paths. Nonetheless, not all are exploited at the same time, as can be inferred from the existence of paths selected by no robot. This can be explained by the oscillation dynamics discussed earlier. When the amplitude of the oscillations is greater than the

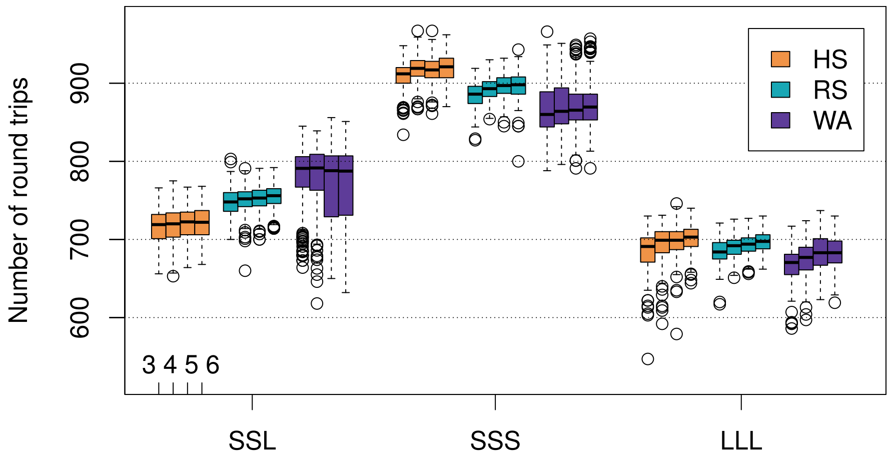

Figure 3.9: Efficiency of the swarm for multiple goals and all mechanisms and conditions. Each box represents the inter-quartile range, whiskers extend to 1.5 times the corresponding quartiles, and the dots represent outliers.

number of robots on a path, all the robots on this path switch to another one. This happens in the case of multiple goals because the robots are spread among more paths, and their number on each is therefore lower.

To better understand the exploitation of the available resources/goals, in Tab. 3.1 we report the average percentage of robots on the different paths, ordered from the most to the least exploited path. We note that the number of exploited goal locations is usually no greater than 3. This explains why the efficiency of the swarm does not vary with the number of available resources, as shown in Fig. 3.9. The slight increase in performance can be attributed to the fact that the more goals there are, the easier it is for uncommitted robots to join a path earlier in the experiment. Overall, we note similar patterns over efficiency between the multiple goals condition and the two goals condition.

When there are multiple goals, *WA* in the *SLL* condition leads to a frequent selection of a distant goal instead of the closest one, as shown in Fig. 3.10. If several distant locations are present, they end up reinforcing each other as their angular distance becomes smaller. In other words, two distant goal locations that are close to each other attract more robots than a single closer location. This explains why the chance of *WA* leading to the selection of a distant goal increases with the number of goals.

Table 3.1: Repartition in percentage of robots for 3, 4, 5 and 6 goals. The 1^{st} goal is the one associated with the highest number of robots. The mean and maximum of the standard deviation is (4.7, 10.5) for *HS* and *RS* and (6.1, 16.9) for *WA*.

	SL			*SS*			*LL*		
	HS	*RS*	*WA*	*HS*	*RS*	*WA*	*HS*	*RS*	*WA*
1^{st}	98.5	98.1	96.0	48.0	52.6	93.0	48.8	47.0	90.8
2^{nd}	0.1	0.6	2.3	34.3	37.3	5.7	33.4	32.5	7.0
3^{rd}	0.0	0.0	0.0	17.2	9.7	0.0	17.4	19.7	0.1
1^{st}	98.4	97.7	95.2	50.6	54.1	92.3	44.8	43.8	89.5
2^{nd}	0.2	1.0	3.6	35.2	38.0	6.8	32.0	31.0	9.2
3^{rd}	0.0	0.1	0.0	12.5	6.9	0.0	17.6	17.2	0.1
4^{th}	0.0	0.0	0.0	2.1	0.7	0.0	5.1	7.0	0.0
1^{st}	98.6	97.3	92.4	51.1	51.1	94.8	44.9	42.6	89.4
2^{nd}	0.2	1.0	6.8	35.2	37.0	4.5	31.6	30.0	9.5
3^{rd}	0.0	0.1	0.0	12.5	10.4	0.2	17.7	17.7	0.5
4^{th}	0.0	0.0	0.0	1.1	1.1	0.0	5.1	7.3	0.0
5^{th}	0.0	0.0	0.0	0.0	0.0	0.0	0.3	1.4	0.0
1^{st}	98.6	97.3	93.6	50.1	53.0	94.7	43.7	42.4	88.5
2^{nd}	0.2	1.5	5.4	34.9	36.1	4.7	31.7	28.2	10.4
3^{rd}	0.0	0.1	0.5	13.5	9.2	0.2	17.2	17.3	0.6
4^{th}	0.0	0.0	0.0	1.4	1.3	0.0	6.0	8.3	0.0
5^{th}	0.0	0.0	0.0	0.1	0.2	0.0	0.8	2.3	0.0
6^{th}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0

3.3.4 Discussion

The experiments above reveal the specificities of the three information aggregation mechanisms. *WA* leads to convergence to a single path in all conditions, but this is slower and error-prone. On the whole, *WA* leads to better cohesion of the swarm and deals better with congestion thanks to more accurate information about the target areas. *HS* and *RS* also lead to convergence when there is a shorter path to exploit, and handle better the presence of multiple distant goal locations. When congestion results in inefficient navigation, both mechanisms lead to the exploitation of multiple paths, spreading the load of robots in a balanced way with similar dynamics, although *HS* appears to be stiffer than *RS*.

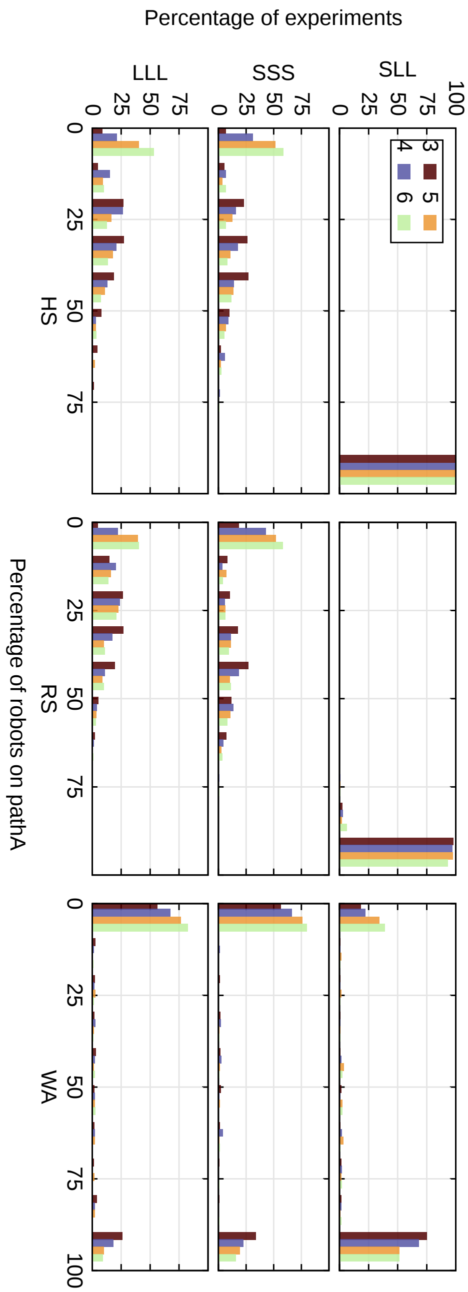

Figure 3.10: Robots repartition on path A for different number of goal areas (3,4, 5 and 6). Each histogram shows the observed frequencies of the number of robots committed to path A (the shortest possible path).

Chapter 4

Exploitation Task

In this chapter, we focus on the exploitation ability and patterns of the swarm as supported by the social odometry navigation mechanism. To this end, the resources are represented as a spreading of cylindrical items on the ground. By using objects, we give a topology to the goal areas, and hence we can study how the different information aggregation mechanisms described in Chapter 3 react when confronted with physical resources, and their exploitation efficiency.

As in the previous chapter, we start by introducing the now updated individual behaviour of the robots. Then, we introduce the experimental setup and finally discuss the obtained results.

4.1 Individual Behaviour

In this chapter, the behaviour of the robot is defined by a slightly different finite state automaton (Fig. 4.1). The state *Leave Goal* is now replaced by the state *Grab Item* for resources are no longer painted areas on the ground but items to be retrieved. The robots now go toward the goal area, grab an item, return home, drop the item, and start again following the foraging loop defined by the states *Go to Goal*, *Grab Item*, *Go Home* and *Leave Home*. When lacking information, the robots fall back to the *Explore* state in which they start at the beginning of each experiment. On top of these control states, both short and long range collision avoidance are implemented.

In the *Go to Goal* state, the robot moves straight to the target location, possibly avoiding other robots and obstacles. As in previous experiments, each robot updates its information (position and confidence) using odometry. Whenever a robot sees a resource item, it probabilistically enters the *Grab Item* state with a probability $P(grabbing)$. We wanted on average to allow the robots to cross the resource, which led to $P(grabbing) = 1/(v \cdot d)$, where v is the speed of the robots and d is two times the standard deviation of the Gaussian spread of items characterising all resources. If the robot reaches the estimated location of

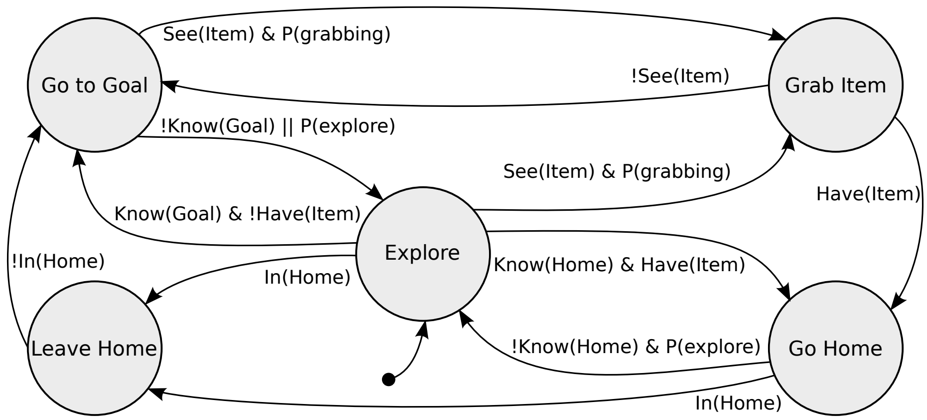

Figure 4.1: Robot's finite-state automaton. The circles define the states while the arrows define the transitions. *In(Home)* is true when the robot senses a grey colour on the ground. *Know(Area)*, *Area* $\in \{Home, Goal\}$, is true when the robot has an estimation of the position of the area. *Have(Item)* is true when the robot is holding an item. *See(Item)* is true when the robot is able to see a grabbable item with its camera sensor. $P(grabbing)$ is the probability a robot will go grab the closest item.

the target goal before getting to grab an item, it goes back to the *Explore* state. The robot has a small probability $P(explore)$ of going back to the explore state, which ensures that the robot does not remain idle in case it cannot reach the estimated position of the goal area. This allow the swarm to reach a balance between the maximisation of currently known resources and the exploration of potential new ones.

Once in the *Grab Item* state, the robot moves toward the closest item. Then, once in contact, it grabs it and at the same time stores the resource location as the average position of all grabbable items in sight. The robot always selects and goes toward the closest grabbable item, which may change over time due to robot movements or changes in the environment. Having grabbed the item, the robot enters the *Go Home* state. If for any reason no further items are in sight, the robot goes back to the *Go to Goal* state.

The *Go Home* state works closely as the *Go to Goal* state. In this case, the robot moves straight toward home. If it reaches the grey painted area, it enters the *Leave Home* state, and iterates the loop anew. If not, it goes back to the *Explore* state, either because it has reached its estimated position of the home location (without entering the grey area, implying that the robots had bad information memorised), or because of the probability $P(explore)$ to explore again.

Finally, when in the *Leave Home* state, the robot moves in the home area following a random walk pattern (possibly dodging other robots to avoid collisions) and probabilistically drops its item with probability $P(dropping)$. Once out, it stores the home location as the average of its entering and exiting positions.

When a robot is out of the foraging loop, it is in the *Explore* state. In this state, it performs a random walk, either searching for grabbable objects or for the home location. If the robot sees an item, it enters the *Grab Item* state with probability $P(grabbing)$. Otherwise, the robot exits from the *Explore* state only when it obtains the position of either the goal or home, either from its own sensors or through social interaction.

4.2 Experiments

At the beginning of an experiment, the robots are spread inside an arena containing a home (circular grey area painted on the ground) and one or more goals of varying quality, as depicted in Fig. 4.2. The goals are regions with items to be grabbed and brought back home. Using real objects lets us shape these regions and define their topology through the items themselves, as opposed to as regions painted on the ground, which are defined symbolically (allowing for only abstract interactions). Goals/resources are Gaussian scatterings of items around their centre with a fixed standard deviation of 0.5. Goal regions were intended not as a dense bulk of numerous items (forcing interaction only on its edge) but as a balanced spread of object between which the robots can manage to get around. For this purpose, we introduced a minimum distance d_{min} between the cylinders equal to 5 times the robot's radius.

Furthermore, using real objects has the added effect of making the resources more complex, allowing for greater variations. The main motivation for using real objects (despite growing closer to real life situation and conditions) was to integrate a notion of quality in the new goal areas. The quality characterises the number of items present in a source at a given time. It is defined by the maximum number of items and their rate of replenishment, expressed in item per second. This way, the quality of a resource is grounded in reality and shares common proprieties with real life conditions. In the following experiment, we focus on the study of the rate of replenishment: the maximum number of items in a resource is fixed at 35, which means that higher qualities are provided by higher rates of replenishment.

In all the following experiments we measure when possible: the number of items brought back home per second from each resource, the number of objects in each resource, the number of robots exploiting each goal/resource, the robots' switches among goals, and the quality of their localization information.

Congestion in the presence of items Our first objective is to understand the swarm behaviour with respect to navigation between the home and the resources, and the impact of the presence of physical items on previous congestion results. In order to compare painted area goals and spread item goals, resources should always have enough items for robots to grab in this setup. For that, the resources need to have a high constant number of grabbable items, implying

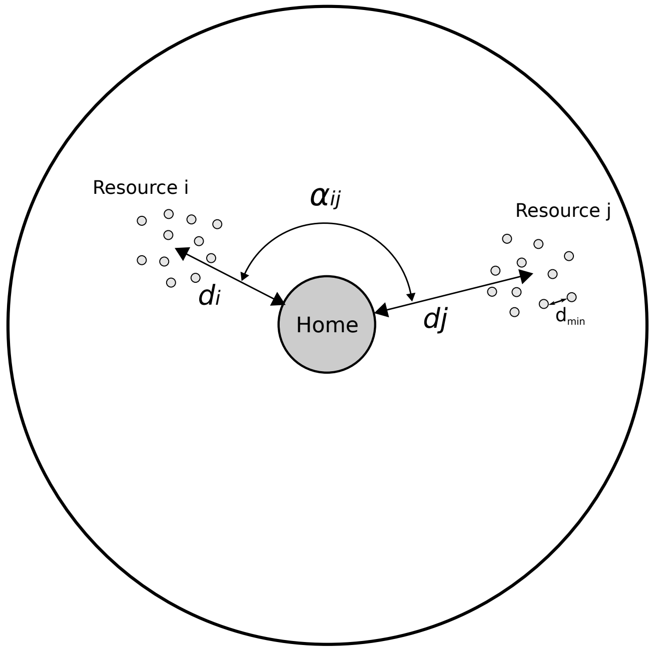

Figure 4.2: Setup of the experimental arena for the exploitation task. The home area is placed in the center of a circular arena of an 11 m radius, surrounded by walls. In this setup, the goals are resources, defined as a Gaussian spread of items with a minimum distance between them of d_{min} (five times the robots' radius). They are characterised by their distance from home d_i, d_j, the angles they form with each other α_{ij} and their respective quality.

an infinite quality. When an object is grabbed, it is immediately replaced by another one in the same position, so that the resource size remains constant over time. This way, we focus on the navigation dynamics only.

Similar to the preceding congestion experiment, here we study the impact of the density of robots on congestion and hence navigation. For that, we use a setup with only one resource. We vary both its distance from home and the number of robots to reach the wanted robot density values.

The following two sets of experiments focus on the exploitation of resources of varying quality, and the way a swarm of robots using *social odometry* reacts to a dynamic environment.

Optimal exploitation of a single source As mentioned above, the main interest of using real objects is the ability to study the impact of the quality of a resource on the decision-making process of the swarm. In this experimental setup, we redo the same experience than as above (one resource of items, varying density of robots), but with a finite rate of replenishment. We vary its rate (0.1 $item/s$, 0.5 $item/s$ and 1.0 $item/s$) and study how various density of robots (5 $robot/m$ to 23 $robot/m$) manage to exploit a resource bearing this rate. While the density variations are made by varying the number of robots, in this setup the distance between the resource and the nest is constant, at the average of both the previously defined short and long distance (*i.e.*, $d = 6.5\ m$). Through these experiences, we search for optimum rates of a resource's exploitation and their link with density and the rate of replenishment.

Optimal exploitation of two sources Finally, we study how the swarm decides and adapts in the presence of two sources. We study both the impact of the distance among sources as well as the impact of the rate of replenishment. For that, we choose among two possible distances ($d_{short} = 5\ m$ and $d_{long} = 8\ m$) and two possible rates ($rate_{min} = 0.1\ item/s$ and $rate_{max} = 1\ item/s$) for the resources. We will study each possibility. First, same rate and distance and same rate but different distance (to compare with the previous results). Then same distance but different rate (to study the impact of the rate). And last, different rate and different distance with the further resource having the best replenishment rate (to compare the effect of the rate and the distance from home). Through these experiments, we explore the dynamics of the swarm and its ability to balance between the distance and quality of a resource, and switch dynamically among goals in order to maximise its efficiency.

4.3 Results

In this section we present the current results over each experimental setup described above and compare them to the results presented in chapter 3. We kept the same duration for the trials (20 minutes of simulated time) and the same random seeds. For each run we compute the number of robots on each path to study the dynamics of collective decisions. We also compute the number of round trips to study the navigation efficiency, as well as the error made by the robots on the estimated position of the nest to gauge the quality of information in the swarm.

4.3.1 Congestion

As can be seen in Fig. 4.3 left, all mechanisms follow a commonly shared tendency (sharp rise in low value of density, stalling for higher values). We can not make a direct comparison with results on density from chapter 3 because the resources are not defined in the same way. For instance, the actual perceived distance can be much smaller for resources modelized through items because the spread can grow closer to the nest than a static painted ground area would be. Furthermore, in the updated individual behaviour, each robot has the probability $P(grabbing)$ to stop exploiting the current source and explore. The density values output in Fig. 4.3 are starting densities. In previous experimentations, the robots had no possibility to go back to explore. We observed then a much erratic curve, a quick stall and even a drop in efficiency as density rose. Allowing the robots to explore again when they are stuck on the exploitation path not only give the swarm an opportunity to find better source, but helps the swarm exploiting the current source at an optimal rate by reducing the interferences between robots. In this experimental setup, the swarm self-organises to find a balance in the number of robots: too few would be a loss of potential, too many would make navigation non-practical.

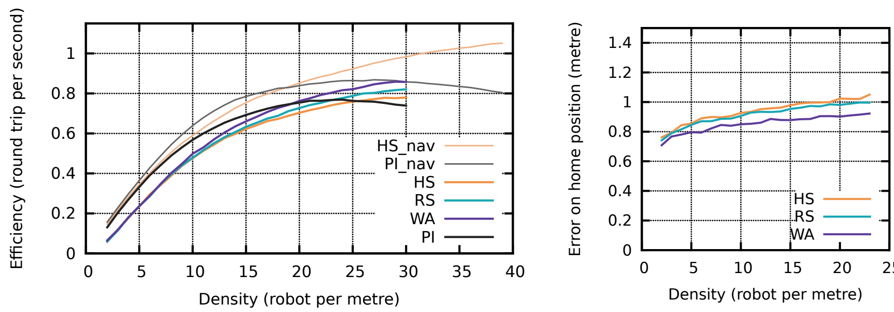

Figure 4.3: Left: Impact of density on navigation efficiency for each mechanism and in the perfect information control condition. Each line is the mean over 100 trials. *HS_Nav* and *PI_Nav* are taken from our previous results from chapter 3 on congestion with simple painted goals, for reference. Right: Evolution of the error of the estimated position of the centre of the nest for each mechanism. Each line is a mean over 100 trials.

If we cannot actually compare the efficiency in absolute value, we can compare the evolution of this efficiency. *HS_Nav* and *PI_Nav* references of previous results shows us that the tendencies of results in each arena setup are similar, with a slightly stronger *RS* compared to both *WA* and *HS*. As in the previous setup, congestion has too a lower impact on navigation efficiency with social odometry. We note that *WA* is more resilient to congestion than *HS* and *RS* for the same reasons mentioned in section 3.3.1. The same trends as with previous arena setup can be seen for the evolution of error (Fig. 4.3 right): error grows with density and *WA* is doing better through its averaging process.

Finally, if both physical setups differ a lot, the end results are similar. This proves first that adding physical interaction with items and the resulting updated individual behaviour do not change dramatically the higher level behaviour of the swarm. Second, such similarity indicates that our first abstraction of the physical setup is pertinent in a simple case with one nest and one resource.

4.3.2 Optimal Rate of Exploitation

Using infinite rate allows us to study only the navigation aspect of the exploitation task. If we want to understand the dynamics of the swarm while exploiting

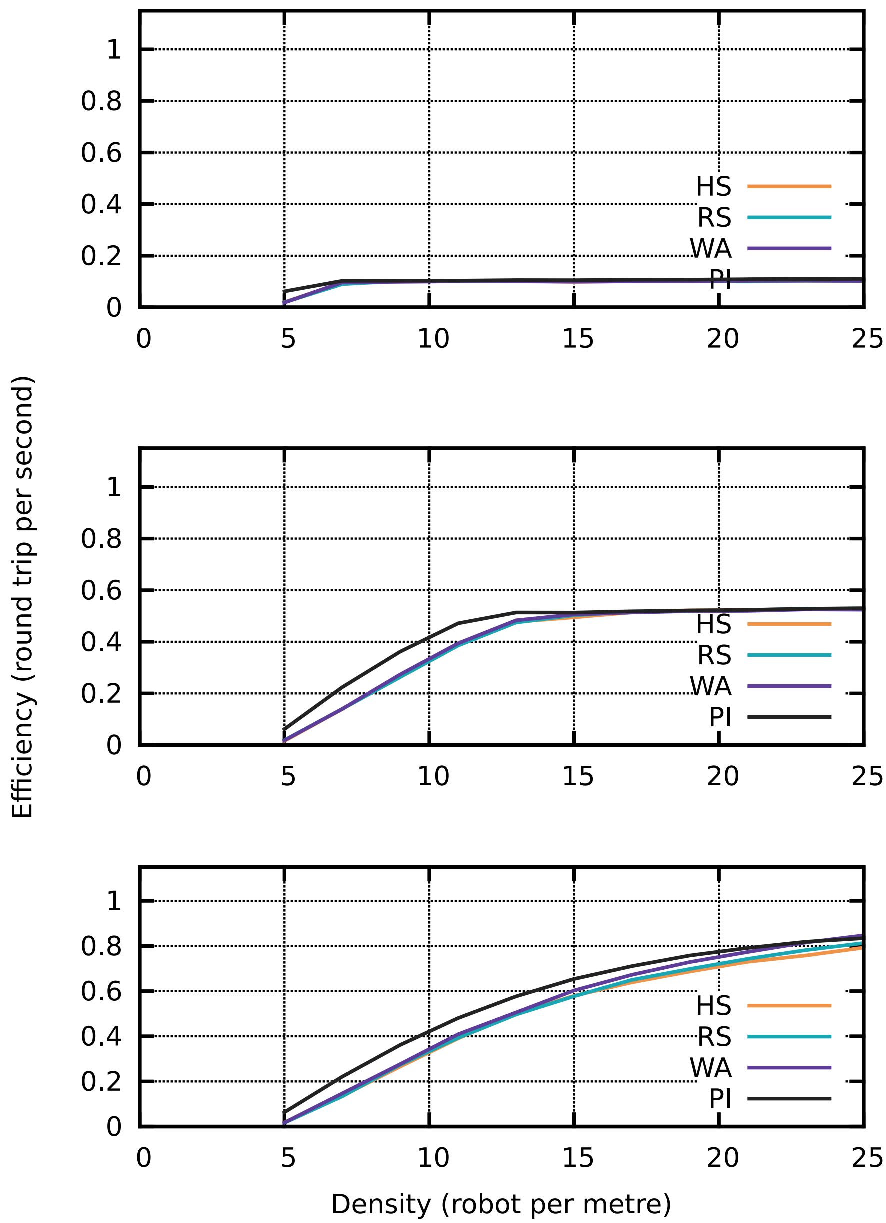

Figure 4.4: Impact of density on navigation and exploitation efficiency for each mechanism and for resources of various replenishment rate. From top to bottom, the rate is 0.1, 0.5 and 1.0. Each line is the mean over 100 trials.

a source, we need to study how the swarm reacts to the presence of a resource of various finite rates (Fig. 4.4).

As in previous subsection, we plotted the efficiency of the swarm over its density. The three chosen rates show the three archetypical results. The top figure corresponds to a low rate (0.1 $item/s$) and display a virtually constant efficiency, no matter the number of robots. Indeed, in this case the rate is so low that even a small number or robots is enough to deplete the source and hence exploit it in an optimal manner. In the bottom figure, the rate is high (1 $item/s$) and the trends of each mechanism displayed in the plot are similar to the ones displayed in previous subsection. In this case, for all density values tested, the resource had a high enough rate not to be depleted. Last, the middle figure has an in between rate (0.5 $item/s$). If at first increasing the density increase the efficiency, the curve reaches quickly a plateau around a density of 13 $robot/m$. After that, the rate is not high enough to withstand so many robots; increasing the density would only increase the number of exploring robots.

4.3.3 Exploitation of Two Resources

In this section, we study in a similar way as in section 3.3.2 the swarm dynamics and the collective decision when resources have varying distances and rates.

Decision

In this section, we find on average similar tendencies than in the previous experimental setup with two goal area painted on the ground (Fig. 4.5). *WA* always converges, even if sometimes on the longer path. *HS* and *RS* converge when a closer resource exists. If such a resource does not exist, then the swarm is split over the possible paths.

The plotted histograms reveal a few differences compared to previous experiments with two goals. First, the convergences are not as strong as previously observed. This is a result of the possibility for the robots to go back to explore when they are already on a path. Second, we see that when there is a competition between a closer source and a source with a better replenishment rate, the later is the one toward which the swarm converges. Last, we note that in the perfectly symmetrical setup (LL with equal rate for both resources), the results are not symmetrical. The reason for this asymmetry is the now significant number of uncommitted robots. Each path bear in average less robots, the effect of which is that all the graphs are translated toward the left. Since the graphs are not symmetrical anymore, it can be hard to spot a convergence just by looking at the histogram. Fig. 4.6 makes this convergence clearer by showing the evolution of the number of robots.

The black separators (at 120 s and 400 s) present in Fig. 4.6 correspond to the three different phases previously observed in chapter 3 (exploration, competition and maximization). They show that despite both experimental setup having different swarm dynamics, their resulting evolution of the number of

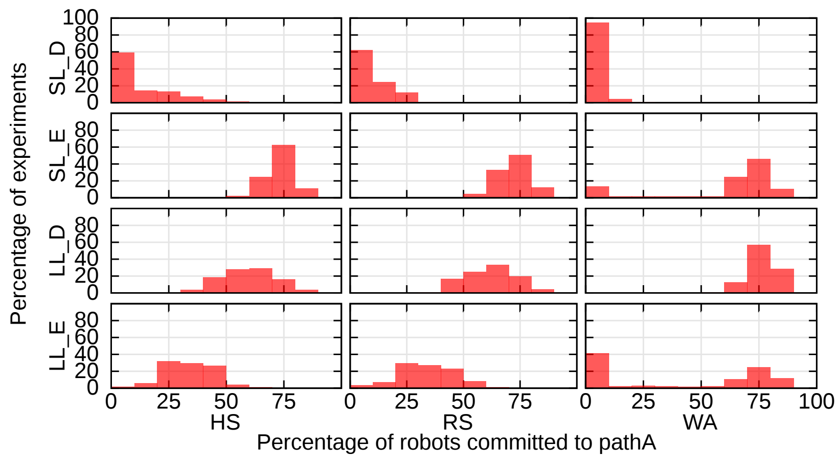

Figure 4.5: Observed frequencies of the number of robots committed to path A (the shortest possible). S: short, L: long, E: equal rate, D: different rate.

robots follow a same rhythm.

Despite a clear spread of robots in the histograms, we observe in Fig. 4.6 top a clear convergence on the path with a better rate of replenishment for the *HS* mechanism. The remaining robots are not committed to the other path but mainly exploring the environment. In this experimental setup, *HS* converges even in the case of resources at similar distance, but only if their rate of replenishment differs. Here the swarm self-organise and proves that not only it can value a resource on its distance from home but also on its rate on its own.

The Middle figure corresponds to the same conditions, but for the *WA* mechanism. The evolution of the number of robots follows a similar trend with the *HS* mechanism. It also converges more strongly on the path linked to the best replenishment rate resource. If all over chapter 3 the number of exploring robots was strictly decreasing over time, it's not the case here anymore. After the number of robots on the longer path reaches its peak, the number of exploring robots starts rising again. It happens closely at the average time at which the resource with lower replenishment rate gets depleted, inclining the robots committed to this resource's path to go explore. Such exploring robots are then integrated in the better path.

Finally, the bottom figure presents a competition between a higher replenishment rate and a closer source while using the *WA* mechanism. We observe that if at first robots converge on the closest source, the later depletes quickly. Then, the robots previously on the shortest path go back to explore and finally join the further but substantial resource.

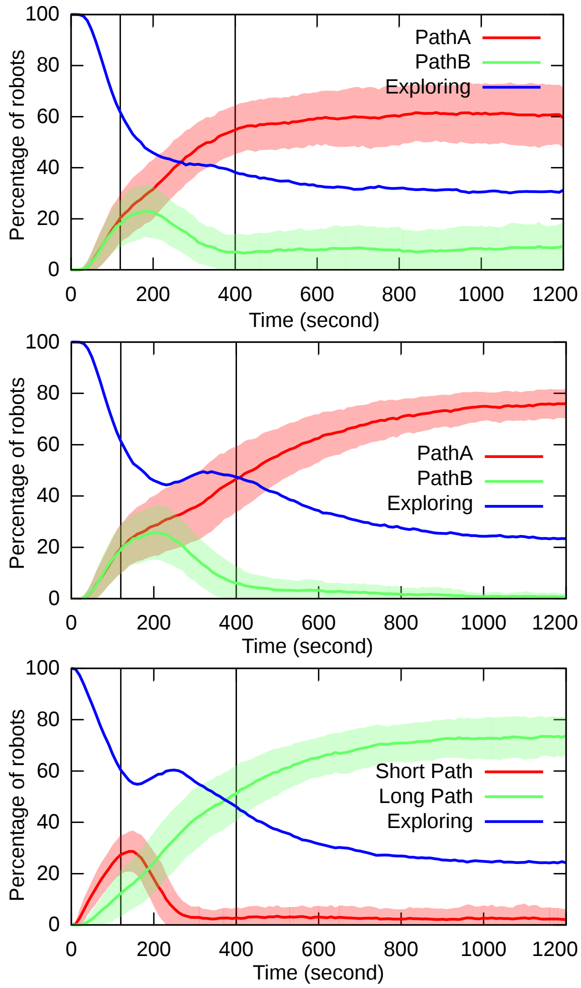

Figure 4.6: Evolution of the robots' repartition with two resources of varying distance (5 m and 8 m) and rate (0.1 $item/s$ and 1.0 $item/s$). Bold lines indicate the mean over 250 repetitions, and the shaded areas indicate the standard deviation. These three figures present the three archetypical convergence patterns with resources of finite replenishment rate.

Top: *HS*, *LL* resources with different rate.
Middle: *WA*, *LL* resources with different rate.
Bottom: *WA*, *SL* resources, the further resource having the best rate.

Efficiency

We saw that the decision process is not only influenced by the distance of the resources from home, but also by the resources' replenishment rate. Fig. 4.7 displays the variation in overall efficiency over all mechanism and experimental setup when two resources defined as a spread of items are present.

When both resources' replenishment rate are equals, the boxplots describing the efficiency of the swarm are similar to those found in section 3.3.3. The main difference is an overall lower efficiency (a slower swarm), due to the physical interaction with the resources' items. When resources bear different rate, they are on average doing worse than when they have the same rate. This can be explained by the fact that the overall rate is higher when the two sources have the same rate. Another reason is that a setup with same rate will incline the swarm to split among two paths, which makes the swarm more efficient as we proved in section 3.3.3.

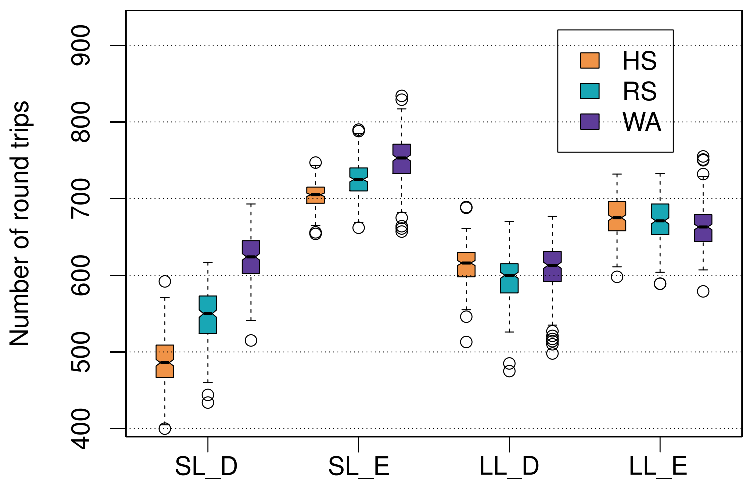

Figure 4.7: Efficiency of the swarm for two goals, for all mechanisms and conditions. Each box represents the inter-quartile range, whiskers extend to 1.5 times the corresponding quartiles, and the dots represent outliers. S: short, L: long, E: equal rate, D: different rate.

Last, in the *SL_D* condition, the swarm is less efficient than in the *SL_E* condition. The reason for that is that the closest (but with low rate) resource is regularly rediscovered and depleted, distracting the robots from the substantial resource. This creates a cycle in which the closest source is regularly depleted and abandoned until it grows back enough for exploring robots to discover it again. These robots then come back home with information in which they are very confident, and hence recruit even more robots. This cycle is even more pronounced for the *HS* mechanism when just one single robot can spread its information to numerous one as long as its confidence is better.

Conclusion

In this paper we presented an extensive analysis of three parameter-free information processing mechanisms for social odometry with abstract goal locations and with resources defined using physical objects. We studied the impact of these mechanisms on both the navigation and exploitation efficiency and on the dynamics of the swarm. In particular, we observed how the information processing mechanism can either lead to a convergence on the exploitation of a single path, or to a split over multiple comparable options. These results are meant to give future designers a guideline of which mechanism they should choose depending on the situation and objectives at hand.

For instance, if the cohesion of the swarm is an important issue, then the *AW* mechanism should be select as it ensures that the group never splits over multiple resources. As for the navigation efficiency, we observed that it highly depends of the congestion on the selected paths. As a consequence, *HS* and *RS* lead to the exploitation of multiple paths whenever congestion results in inefficient navigation. When physical objects are present, the resources' rate of replenishment influence strongly both the efficiency of the swarm and its dynamics. We observed that variations in this rate has an even stronger impact on the efficiency than the distance alone.

Our first results are showing similar trends among both kind of experimental setups, with more realistic interactions in the case of resources defined as spreads of objects. In all setups, the swarm displays a behaviour in which it balance the robots' load over the possible paths (splitting when necessary), implementing a sort of *load-balancing* mechanism. In our future work, we plan to investigate this issue further in order to provide an optimal load-balancing behaviour, which can maximize the exploitation of different paths to relevant areas/resources. Not only would the swarm choose the best distribution of robots among the available paths, but it would also be able to react in real time to changes in its environment.

A number of possible extensions to the presented mechanisms are envisaged. The first straightforward extension is to provide for our social odometry mechanisms to provide a way to deal with more complex paths, for instance in the presence of obstacles. Robots may also be provided with the ability to memorize multiple goal locations, implying that the competition among paths would not be only at the swarm level but also at the individual robots level.

Last, heterogeneity can be added in the swarm. On the one hand, individual robots may get committed to a goal with different individual preferences, leading to a better exploration of the environment. On the other hand, multiple *tribes* of robots could compete for the best source, each of them having different information aggregation mechanisms, leading to a different exploitation of resources among different groups.

Bibliography

Bonani, M., Longchamp, V., Magnenat, S., Rétornaz, P., Burnier, D., Roulet, G., Vaussard, F., Bleuler, H., and Mondada, F. (2010). The MarXbot, a miniature mobile robot opening new perspectives for the collective-robotic research. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4187–4193. IEEE Press, Piscataway, NJ.

Camazine, S. (2003). *Self-organization in biological systems*. Princeton University Press.

Campo, A. and Dorigo, M. (2007). Efficient multi-foraging in swarm robotics. In *Advances in Artificial Life*, pages 696–705. Springer.

Ducatelle, F., Di Caro, G., Pinciroli, C., Mondada, F., and Gambardella, L. (2011). Communication assisted navigation in robotic swarms: Self-organization and cooperation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4981–4988. IEEE Computer Society Press, Los Alamitos, CA.

Garnier, S., Gautrais, J., Asadpour, M., Jost, C., and Theraulaz, G. (2009). Self-organized aggregation triggers collective decision making in a group of cockroach-like robots. *Adaptive Behavior*, 17(2):109–133.

Garnier, S., Tâche, F., Combe, M., Grimal, A., and Theraulaz, G. (2007). Alice in pheromone land: An experimental setup for the study of ant-like robots. In *Swarm Intelligence Symposium, 2007*, pages 37–44. IEEE Press, New York City, NY.

Gutiérrez, A., Campo, A., Monasterio-Huelin, F., Magdalena, L., and Dorigo, M. (2010). Collective decision-making based on social odometry. *Neural computing & applications*, 19(6):807–823.

Gutiérrez, A., Campo, A., Santos, F., Monasterio-Huelin Maciá, F., and Dorigo, M. (2009). Social odometry: imitation based odometry in collective robotics. *International Journal of Advanced Robotic Systems*, 6(2):129–136.

Liu, W., Winfield, A. F., Sa, J., Chen, J., and Dou, L. (2007). Towards energy optimization: Emergent task allocation in a swarm of foraging robots. *Adaptive Behavior*, 15(3):289–305.

Martinelli, A., Pont, F., and Siegwart, R. (2005). Multi-robot localization using relative observations. In *IEEE international conference on robotics and automation (ICRA)*, pages 2797–2802. IEEE Press, New York, NY.

Montes de Oca, M. A., Ferrante, E., Scheidler, A., Pinciroli, C., Birattari, M., and Dorigo, M. (2011). Majority-rule opinion dynamics with differential latency: a mechanism for self-organized collective decision-making. *Swarm Intelligence*, 5(3-4):305–327.

Nouyan, S., Groß, R., Bonani, M., Mondada, F., and Dorigo, M. (2009). Teamwork in self-organized robot colonies. *IEEE Transactions on Evolutionary Computation*, 13(4):695–711.

Olfati-Saber, R., Fax, J. A., and Murray, R. M. (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233.

Ostergaard, E. H., Sukhatme, G. S., and Matari, M. J. (2001). Emergent bucket brigading: a simple mechanisms for improving performance in multi-robot constrained-space foraging tasks. In *Proceedings of the fifth international conference on Autonomous agents*, AGENTS '01, pages 29–30, New York, NY. ACM.

Pinciroli, C., Trianni, V., O'Grady, R., Pini, G., Brutschy, A., Brambilla, M., Mathews, N., Ferrante, E., Di Caro, G., Ducatelle, F., Birattari, M., Gambardella, L. M., and Dorigo, M. (2012). ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence*, 6(4):271–295.

Rekleitis, I., Dudek, G., and Milios, E. (2001). Multi-robot collaboration for robust exploration. *Annals of Mathematics and Artificial Intelligence*, 31(1-4):7–40.

Rybski, P. E., Larson, A., Veeraraghavan, H., LaPoint, M., and Gini, M. (2007). Communication strategies in multi-robot search and retrieval: Experiences with mindart. In *Distributed Autonomous Robotic Systems 6*, pages 317–326. Springer.

Shell, D. A. and Mataric, M. J. (2006). On foraging strategies for large-scale multi-robot systems. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 2717–2723. IEEE.

Sperati, V., Trianni, V., and Nolfi, S. (2011). Self-organised path formation in a swarm of robots. *Swarm Intelligence*, 5(2):97–119.

Thrun, S. (2008). Simultaneous localization and mapping. Springer Tracts in Advanced Robotics, pages 13–41–41. Springer Berlin Heidelberg.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. MIT Press, Cambridge, MA.

Winfield, A. F. (2009). Towards an engineering science of robot foraging. In *Distributed Autonomous Robotic Systems 8*, pages 185–192. Springer.