Information flow principles for plasticity in foraging robot swarms

Lenka Pitonakova[1,3], Richard Crowder[1] and Seth Bullock[2]

[1]Institute for Complex Systems Simulation and Department of Electronics and Computer Science,

University of Southampton, United Kingdom

[2]School of Computer Science, University of Bristol, Bristol, United Kingdom

[3]contact@lenkaspace.net

Note: This paper was published in (2016) Swarm Intelligence, 10(1), 33-63, DOI: 10.1007/s11721-016-0118-1. Supplementary material is available on the journal web site.

Abstract: An important characteristic of a robot swarm that must operate in the real world is the ability to cope with changeable environments by exhibiting behavioural plasticity at the collective level. For example, a swarm of foraging robots should be able to repeatedly reorganise in order to exploit resource deposits that appear intermittently in different locations throughout their environment. In this paper, we report on simulation experiments with homogeneous foraging robot teams and show that analysing swarm behaviour in terms of information flow can help us to identify whether a particular behavioural strategy is likely to exhibit useful swarm plasticity in response to dynamic environments. While it is beneficial to maximise the rate at which robots share information when they make collective decisions in a static environment, plastic swarm behaviour in changeable environments requires regulated information transfer in order to achieve a balance between the exploitation of existing information and exploration leading to acquisition of new information. We give examples of how information flow analysis can help designers to decide on robot control strategies with relevance to a number of applications explored in the swarm robotics literature.

Keywords: Swarm robotics, Foraging, Communication, Self-organisation, Plasticity

---

1. Introduction

Plasticity in a robot swarm is the ability of the robots to repeatedly organise and reorganise in response to changing demand characteristics. As such it is central to the ability of a robot swarm to cope with real-world tasks when they are extended in time or space, since these typically feature inherent dynamism in the demand characteristics that the swarm must deal with. In this paper, we study the emergence of swarm plasticity in the context of swarm foraging with homogeneous simulated swarms that need to respond to changing resource deposits. We develop an information flow measure that allows us to understand the swarm's self-organisation and present robot behavioural strategies that lead to plasticity at the level of the entire swarm.

Swarm foraging is a behaviour that has received a considerable amount of attention in swarm robotics (e.g., Krieger and Billeter, 2000; Liu et al., 2007; Gutiérrez et al., 2010). Robots must find and collect resources distributed throughout their environment. While designing strategies for swarm foragers is challenging, such strategies allow swarms of robots to coordinate their exploitation of the environment and to potentially forage more effectively. Furthermore, foraging is often used as a paradigm for studying other swarm behaviours, such as task allocation (e.g., Jevtic et al., 2012) or labour division (e.g., Zahadat et al., 2013), where worksites need to be searched for in the environment and work is distributed among the robots, or agent dispersion (e.g., Ranjbar-Sahraei et al., 2012), where robots need to coordinate their movement via communication. We choose foraging as a testbed application in this paper so that our results are relevant to a spectrum of topics covered in the current swarm robotics literature.

If we could create swarms that are not only able to perform given tasks, but to perform them autonomously and reliably over a prolonged period of time in changeable environments, robot swarms could collect garbage from our streets, extract and gather minerals or be used in logistic applications. However, it is not their ability to forage itself, but rather their ability to adapt that makes swarm foraging systems interesting and useful (e.g., Dai, 2009; Ducatelle et al., 2011). Therefore, if robot swarms are to address real-life applications, we need to understand how swarm intelligence works in dynamic environments. In order to do this, we need to establish ways of analysing artificial swarms so that we can generalise findings from particular experiments and aim towards a framework or a set of design principles that can guide future research and engineering. Recent work on self-organisation in artificial swarm systems has shown an increasing demand for such generalisations (e.g., Parunak and Brueckner, 2004; Serugendo et al., 2006; Winfield, 2009).

In order to gain a detailed understanding of swarm foraging and identify what robot-level behaviour leads to swarm-level plasticity, we simulate and analyse foraging robot swarms of a gradually increasing complexity in a number of different environments. Our robots utilise recruitment, i.e., they inform each other about foraging locations through direct communication. Section 2 gives an overview of relevant aspects of social insect foraging behaviour and discusses state of the art in swarm robotics. We introduce our simulation environment and the robot control algorithm in Section 3 and evaluate performance of our robot swarms in simple static environments in Section 4. We then test the ability of swarms with various communication strategies to discover and discriminate between deposits of different quality in Sections 5 and 6, and we demonstrate the importance of a balance between exploration and exploitation in dynamic environments, where deposit quality changes over time. By measuring *information value*, which captures the direct effect of information flow on resource collection, we show in Sections 7-9 that if information spreads slowly through swarms they are likely to remain effective as foraging conditions change, due to their ability to balance information utilisation and acquisition. On the other hand, if information spreads quickly through swarms, while performance may be improved under specific environmental conditions, such swarms may also be less capable of plastic self-organisation. The information value measure can thus help designers to decide on robot behavioural parameters that lead to desired work modes for their swarms. Furthermore, in Section 9, we offer a set of swarm design principles focused on communication and discuss their relevance to a wider range of problems including swarm foraging from multiple sites in parallel, swarm self-regulation, emergent task allocation and evolutionary robotics.

2. Background

2.1. Swarm foraging in nature

The behaviour of swarm robot foragers is often inspired by ants or bees. Colonies of these social insects demonstrate an incredible ability to self-organise when their environment changes and have therefore been studied by both biologists and engineers.

Ants use pheromone trails to indicate paths through the environment and to inform their nest mates about foraging locations (Sumpter and Beekman, 2003; Arab et al., 2012). Pheromone paths that lead to better food sources and are used by more ants become stronger over time and attract more and more workers, while paths to inferior sources, that are not being reinforced frequently enough, gradually evaporate.

Bees are another example of animals that use recruitment when foraging. However, unlike ants, bees use direct signalling and have a designated area in the hive, called the *dance floor*, where they perform recruitment *waggle dances* for nest mates that are interested in foraging (von Frisch, 1967; Seeley et al., 1991; Biesmeijer and De Vries, 2001). While the length and strength of a waggle dance is related to the quality of a particular flower patch, the position and orientation of a bee on the dance floor relative to the sun encodes the location of a patch relative to the hive, allowing recruits to travel to specific advertised food sources. An individual's decisions about whether to waggle dance, forage or abandon a patch are affected by olfactory and taste information from nectar samples obtained from other foragers through *trophallaxis* after one bee sends a *begging signal* to another (De Marco and Farina, 2003; Farina et al., 2005). For example, when a forager discovers that other bees are processing nectar of a much better quality, it abandons its own source faster and has a higher preference for the better source for a number of days (De Marco and Farina, 2001).

Compared to ants, bees are generally better at achieving plasticity. For example, ants find it difficult to establish a new shorter route to a food source if an established trail already exists (Ribeiro et al., 2009). Furthermore, ants rely on pheromone evaporation, which takes time and makes diversion of foraging effort from a depleted to a new patch relatively slow. On the other hand, bees abandon patches based on individual decisions, making their responses faster when a patch is depleted (Sumpter and Beekman, 2003). Bee colonies also have the ability to switch between patches when their relative quality changes (Seeley et al., 1991). It has been argued that trophallaxis plays an important role in dynamic environments, as it allows information about flower patch quality to spread through the whole hive within hours, while waggle dancing only affects bees that follow dances and is thus a slower communication method (Farina et al., 2005). Additional evidence suggests that returning scouts use *stop signals* to directly inhibit waggle dances of recruiters that advertise alternative sites (Seeley et al., 2012). Bee colonies also achieve flexibility through opportunistic scouting when a recruited forager gets lost due to errors in signal propagation during waggle dance (Seeley, 1994) and through *inspection*, i.e., occasional re-evaluation of abandoned flower patches (Granovskiy et al., 2012).

2.2. Robot swarm foraging in static environments

There is a considerable volume of research that concentrates on swarm foraging during which robots are solitary and do not communicate at all. Such swarms are often used in static environments to retrieve targets and bring them back to a designated location (e.g., Arkin, 1992; Balch, 1999; Ulam and Balch, 2004).

Ant-inspired robots in simulated experiments can drop cues directly into the environment in order to help others reach items of interest (Drogoul and Ferber, 1993). The use of pheromone and its evaporation and dispersion can also be simulated (Fujisawa et al., 2014). In the real world, robots that move on a phosphorescent floor can use LEDs to create glowing paths (Mayet et al., 2010). Alternatively, robots can deposit alcohol trails and use chemical sensors to follow them (Russell, 1999; Fujisawa et al., 2014) or a centralised server can store virtual pheromone deposited by robots and use a projector to display it, allowing robots to follow pheromone trails using visual sensors (Kazama et al., 2005; Garnier et al., 2007). In order to avoid the difficulty of using pheromone-like substances that require a specific arena setup, a virtual pheromone is often represented by designated stationary robots that communicate pheromone levels to others nearby (e.g., Hoff et al., 2013; Ducatelle et al., 2011). Similarly, in a setup inspired by bee trophallaxis, a portion of the swarm is designated for propagation of values passed from robot to robot, allowing a gradient to be established between the base and a resource patch (Schmickl and Crailsheim, 2008; Nouyan et al., 2009). It is also possible to use the whole swarm as a medium that holds pheromone paths, allowing virtual ants to travel through the robots and establish the shortest path to a resource (Campo et al., 2010).

Some ant-inspired (e.g., Krieger and Billeter, 2000), as well as bee-inspired (e.g., Alers et al., 2011; Lee and Ahn, 2011) robotic systems use direct signals that allow robots to recruit others to specific deposit locations when they arrive at the base. Alternatively, robots may communicate at any point when they meet each other during foraging (e.g., Valdastri et al., 2006; Gutiérrez et al., 2010; Miletitch et al., 2013). To localise themselves and the objects of interest, robots use path integration and usually store one vector pointing towards the base and one pointing towards a found deposit.

Bee-inspired algorithms have also been applied to help robots aggregate in areas of interest (Schmickl and Hamann, 2010) or to optimise a swarm's energy intake from resource that was collected in the environment and processed in the base (Thenius et al., 2008). It is often difficult to perform experiments with large swarm sizes, to repeat experiments a sufficient number of times, or to collect statistics about robot behaviour during real-world experiments, causing many researches to rely on simulated data when a thorough analysis of swarm behaviour is required (e.g., Lee and Ahn, 2011; Liu et al., 2007; Campo and Dorigo, 2007; Miletitch et al., 2013).

2.3. Robot swarm foraging in dynamic environments

When foraging in environments that change over time, swarms need to possess some form of self-organisation on the level of the collective or adaptation at the individual level in order to cope with changing foraging conditions. Bees use a combination of both techniques. They adapt their response thresholds to various stimuli (Seeley, 1994) and learn odours of profitable flowers (Farina et al., 2005). Colonies are also capable of self-organisation that results from evolved responses of bees to individual and social information (e.g., Seeley et al., 1991; De Marco and Farina, 2003).

A frequently studied swarm robot behaviour is the ability of swarms to adjust the number of foragers and resting robots based on changing resource abundance. Robots can be equipped with means to perceive their own foraging performance and the performance of others and adapt their own control parameters accordingly (e.g., Campo and Dorigo, 2007; Liu et al., 2007). Alternatively, self-organisation on the swarm level, that emerges when robots change their actions based on individual and social information, can also be used to make swarms adapt (e.g., Sarker and Dahl, 2011).

In other experiments, foraging robots were required to distribute their foraging effort between different types of resource proportional to type abundance. It was shown that self-organising swarms could successfully solve this task if a correct balance between information sharing among robots and information acquisition by robots is achieved (e.g., Jones and Mataric, 2003; Schmickl et al., 2007).

Finally, some foraging experiments involve bucket-brigading, where robots form chains of adjacent "working areas" that run from the deposit location to the robot base, and robots move resources along these chains (e.g., Shell and Mataric, 2006; Nouyan et al., 2009; Pini et al., 2013). An individual robot's working area can be adapted on-line to give a swarm the ability to follow mobile deposits (Lein and Vaughan, 2009).

Here, in an approach similar to that of Jones and Mataric (2003); Schmickl et al. (2007); Sarker and Dahl (2011), robots do not adapt their individual control parameters, but share information with one another in order to achieve self-organisation at the level of the swarm.

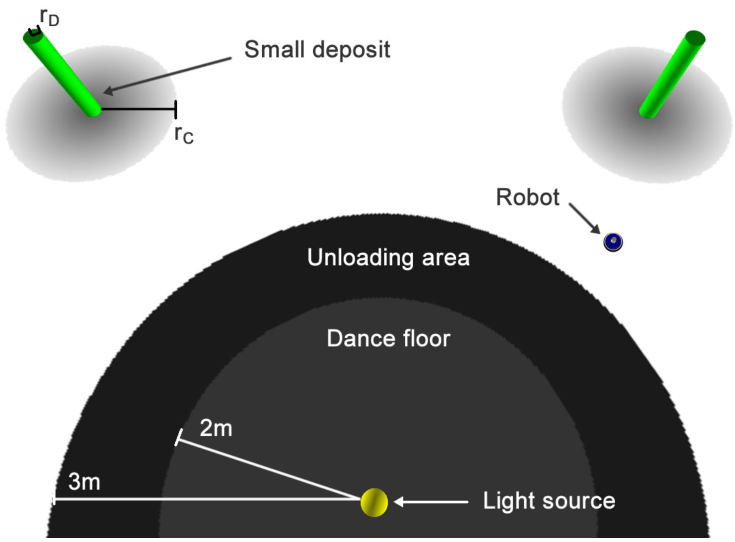


Figure 1: ARGoS simulation screenshot of a base and scattered small deposits. The base consists of a circular dance floor with a radius of 2 m, where recruitment takes place, and an unloading area that forms a ring around the dance floor where returning foragers drop collected material. There is also a light source above the centre of the base. The whole base has a radius of 3 m. A robot with radius of 8.5 cm is located near the base. Each deposit has a colour gradient around it to allow nearby robots to navigate towards it.

3. Methods

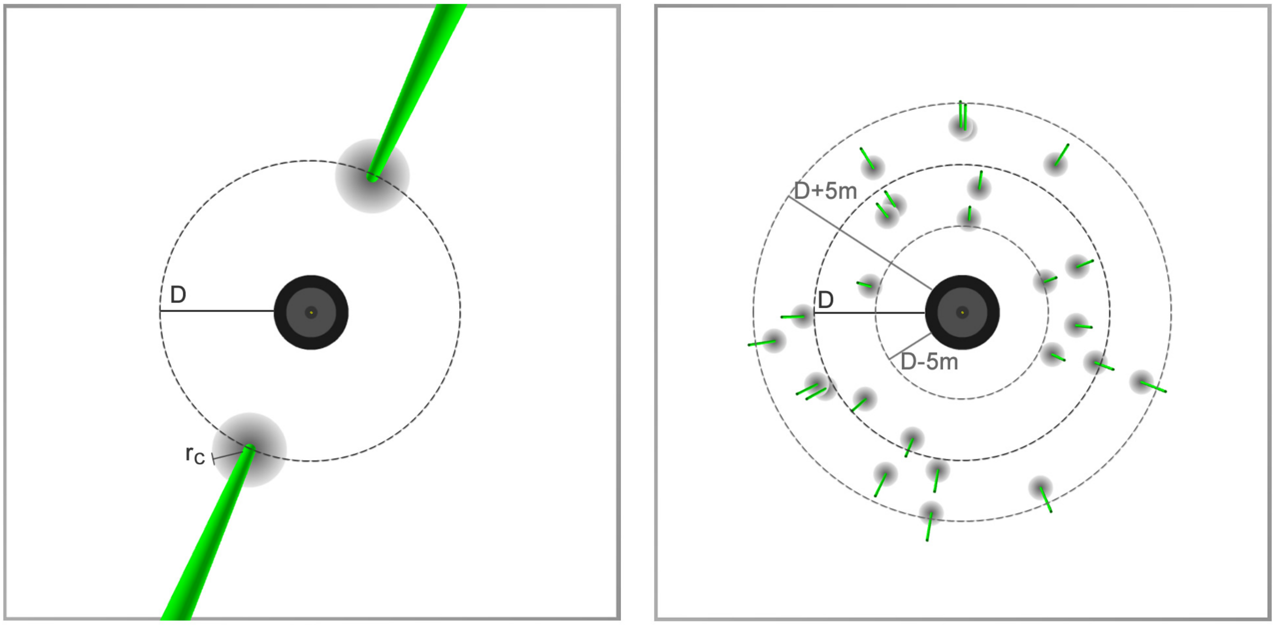
3.1. Simulation environment

All the experiments reported here are performed in the ARGoS simulation environment (Pinciroli et al., 2012) using MarXbot robots (Bonani et al., 2010). ARGoS is a C++ environment with a realistic 3D physics engine that was specifically designed for program compatibility with MarXbots (also called footbots) and e-pucks. By using this type of simulation environment, we can model not only decision making of robots, but also their physical interactions with each other and with the world. The physical aspect is important as it can lead to various interferences between robots and affect their performance, for example when multiple robots try to access a deposit at the same time or when they attempt to communicate with each other (Pitonakova et al., 2014).

The simulation takes place in continuous space and updates itself 10 times per second. The foraging arena is 50 m × 50 m large and contains a centrally located circular base surrounded by resource deposits (Figures 1 and 2). A similar setup has been used previously in simulated (e.g., Balch and Arkin, 1994; Campo and Dorigo, 2007) and real world (e.g., Labella et al., 2004; Gutiérrez et al., 2010) experiments.

The base has a radius of three metres and it is divided into two sections: an interior circular dance floor and an annular unloading area around it (Figure 1). A light source is placed above the middle of the base that the robots can use as a reference for navigation towards and away from the centre of the base (as in, e.g., Krieger and Billeter, 2000; Ferrante and Duéñez Guzmán, 2013; Pini et al., 2013).

Resource is distributed throughout the environment in the form of a number, N, of discrete deposits. Each deposit is cylindrical, with radius, r_D. The value of deposit i to a robot is related to two properties, the *volume* of the deposit, V_i, which represents the gross amount of material it contains, and the *quality* of the deposit, Q_i, which represents how rich this material is. The total amount of resource in a deposit is thus $V_i \times Q_i$. Each robot loads a maximum volume of L_{max} units per foraging trip. The total amount of resource that a robot may load from



(a) (b)

Figure 2: ARGoS simulation screenshot of the experimental arena containing a base in the centre and deposits in the (a) Heap2 and (b) Scatter25 scenarios with $D = 9$ m. The deposits are represented as cylinders, with their heights corresponding to their respective volumes, and have colour gradients with radius r_C around them to guide navigation of nearby robots.

deposit, i, in a single trip is thus $L = L_{\max} \times Q_i$ or $L = V_i \times Q_i$ if $V_i < L_{\max}$.

Consider two deposits, A and B, with volumes greater than $L_{\max}$ and where deposit A has twice the volume of B, but deposit B has twice the quality of deposit A. Robots carrying all of deposit A back to the base will have foraged the same amount of resource as robots carrying all of deposit B back to the base because $V_A \times Q_A = V_B \times Q_B$, but robots foraging from deposit A will have needed more trips to achieve this since $V_A > V_B$.

In the following environments the volume of each deposit is $100/N$ units, i.e., total volume of deposits is 100 units and is conserved across environments. N is always selected so that V_i is an integer. The default maximum value of L is $L_{\max} = 1$ unit, although $L_{\max}$ is reduced to 0.25 units in later simulations in order to increase the amount of time taken by swarms to complete the foraging task. The default value of Q for all deposits is 1, although some scenarios are specifically designed to contain deposits of varying quality.

In order to enable robots close to a deposit to move towards it, a colour gradient with radius r_C is centred on the floor around each deposit. In a real-world experiment, navigation based on the colour gradient could be replaced by visual-based navigation, for instance.

There are two types of scenario (Figure 2):

- **Heap$N$:** N deposits distributed evenly around the base at a distance $D = \{7, 9, 11, 13\}$ m from the base edge. These deposits represent heaps of resource that have large volumes and occupy a large area, with $r_D = 0.5$ m and $r_C = 3$ m. For example, a Heap2 scenario contains two deposits with volume of 50 units each.
- **Scatter$N$:** N deposits randomly distributed between $D - 5$ m and $D + 5$ m from the base edge. These deposits are small, with $r_D = 0.1$ m and $r_C = 1$ m and often numerous, containing a small V each. For example, a Scatter25 scenario contains 25 deposits with volume of four units each.

A laden robot returning to the base deposits its load in the unloading area in the form of N_p number of pellets of size 0.1 m^3, where $N_p = L \times (4\backslash L_{\max})$, set to the closest larger integer. For example, when $L_{\max} = 1$ and $L = 1$, a robot deposits four pellets. When $L_{\max} = 1$ and $L = 0.1$, the robot deposits one pellet, etc. These pellets have the potential to cause congestion in the unloading area if many are deposited at the same time. Deposited pellets disappear from the simulation after a period of *unloading area handling time*, t_H, representing their use or consumption by a hypothetical unmodelled system of robots or human users. By setting $t_H = 0$ s, the system is able to represent pellets that cause no congestion in the unloading area. Scenarios where $t_H > 0$ s are explored in Section 8.2.

3.2. Robots

The simulated MarXbots (Bonani et al., 2010) are circular, differentially steered, robots with a diameter of 0.17 m that in our simulation can reach a maximum speed of 5 cm/s. They are equipped with four colour sensors pointed to the ground, a ring of 24 infra-red proximity sensors used for collision avoidance, a light sensor used for navigation towards the base, a range and bearing module used for localisation of other robots and for communication, wheel mounted sensors utilised for odometry and a ring of eight colour LEDs used for debugging. The maximum range at which the robots can detect objects via the proximity sensors is 0.3 m. Their maximum communication range via the range and bearing module is 5 m. The range and bearing module is based on line of sight, and hence intermittent ranging and communication problems are possible, although they do not have a significant effect in the scenarios we explore here. It is assumed in the simulation that the light sensor can detect the light above the base from anywhere in the experimental arena (as in, e.g., Labella et al., 2004; Gutiérrez et al., 2010). Sensor noise and wheel slippage are not modelled. Consquenetly, there is no odometry error in our simulation.

The robot control algorithm is inspired by foraging bees and the robots are modelled as finite-state machines (Figure 3). A robot starts in a random orientation and position on the dance floor as an *observer*, ready to receive recruitment signals. Observers move randomly on the dance floor and avoid traveling into the unloading area. When a recruitment signal is received, the robot becomes a *forager* and navigates towards a deposit location obtained from the recruiter. Alternatively, an observer still on the dance floor can become a *scout* with scouting probability $p(S) = 10^{-3}$ at each time step.

Scouts leave the base and use Lévy movement (Reynolds and Rhodes, 2009) to search for a deposit within 20 metres from the base. Any scout that cannot find a deposit within 600 seconds returns to the dance floor and becomes an observer. While outside the base, a robot updates its estimation of the relative position of the base using path integration based on odometry at each time step (e.g., Borenstein, 1998; Lemmens et al., 2008; Gutiérrez et al., 2010). When a scout discovers a deposit, it becomes a forager. All foragers load L units of volume of the resource and determine an estimate of the deposit's energy efficiency (after Seeley, 1994):

$$E_E = \frac{V_i' \times Q_i}{d_i} \tag{1}$$

where V_i' is the the volume left in the deposit after the robot's visit, Q_i is the deposit quality and d_i is the odometry-estimated linear distance from the unloading area to the deposit. The robot then returns back to the base utilising phototaxis and keeps track of its relative position to the deposit using odometry.

After a forager unloads its cargo in the unloading area, it moves to the dance floor and becomes a *recruiter* with a recruitment probability $p(R)$:

$$p(R) = \begin{cases} 1.0 & \textit{if } E_E > 0 \\ 0.0 & \textit{else} \end{cases} \tag{2}$$

and performs recruitment inspired by bee waggle dancing for $T_R = 120$ seconds, randomly moving across the dance floor and advertising its deposit location to all observers located within communication range d_C. In order to minimise the influence of the particular direction from which it arrived, and thereby give it the chance to influence more observers, the recruiter travels to the middle

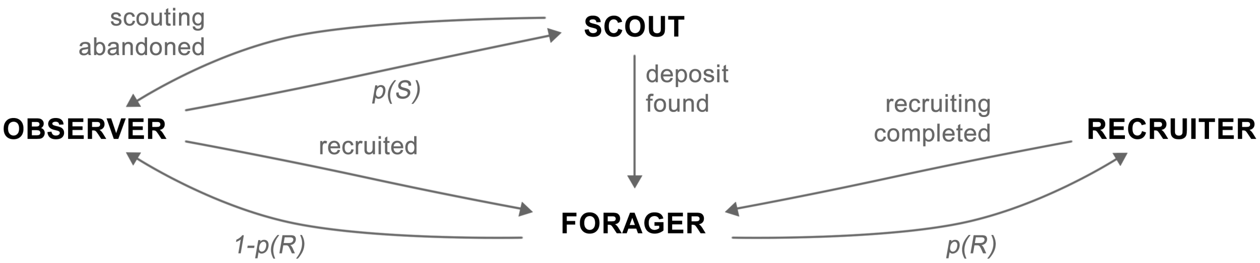


Figure 3: Finite state machine representation of the robot controller.

of the dance floor before it starts recruiting. Similarly, forager bees enter the dance floor from a single direction through a small nest entrance (Seeley and Morse, 1976), allowing potential recruits to interact with any recruiter regardless of its previous foraging location.

The deposit location is communicated to each observer in a one-to-one fashion, where local axes of the robots and their alignment relative to each other are taken into account when conveying the positional information (Gutiérrez et al., 2010). While this technique is sensitive to the accuracy of a robot's sensors (Gutiérrez et al., 2010; Miletitch et al., 2013), it removes the need for a shared reference point. Note that bees, on the other hand, orient their dances relatively to the sun (von Frisch, 1967).

The recruiter always resumes foraging from the same deposit after it completes recruitment. In cases when there is no deposit to return to ($p(R) = 0$), a robot does not recruit or forage again but becomes an observer instead.

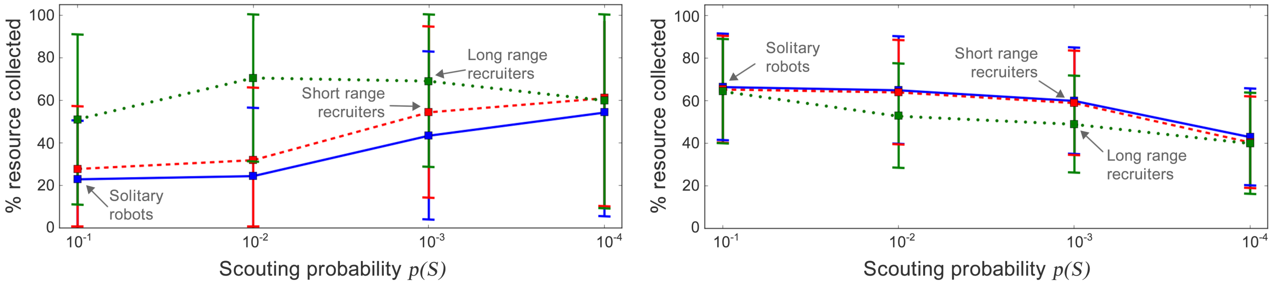
If a forager reaches a deposit location but the deposit cannot be found, the robot performs neighbourhood search that lasts 180 simulated seconds and during which it moves randomly in a circular area with a radius of two metres around the expected deposit location. If the search is unsuccessful, the robot returns to the base to become an observer. Any unsuccessful foragers and scouts are opportunistic and start foraging from a deposit if they find one on their way back to the base.

4. Foraging performance in static environments

Before exploring scenarios that require plasticity, experiments were performed in Heap1, Heap2, Heap4, Scatter10 and Scatter25 scenarios that lasted $T = 1$ simulated hour and where all deposits had a constant quality $Q = 1$ throughout the simulation. We aimed to understand how the communication range, d_C, and the scouting probability, $p(S)$, of robots affected the total amount of resource collected in these scenarios. Each experiment was repeated 50 times. By varying d_C and $p(S)$ to define three types of robot, we could encourage three types of homogeneous robot swarm: (Figure 4):

- Solitary robots, where observers left the dance floor almost instantaneously to scout ($p(S) = 10^{-1}$), and robots could never recruit each other ($d_C = 0$ m)
- Short-range recruiters, where observers spent longer on the dance floor ($p(S) = 10^{-3}$), but robots could only recruit observers that were near to them on the dance floor ($d_C = 0.6$ m)
- Long-range recruiters, where robots could recruit any observer on the dance floor ($d_C = 5$ m, $p(S) = 10^{-3}$)

In the most extreme Heap scenario, Heap1, with only a single large deposit, using large d_C and small $p(S)$ was the most beneficial as it allowed for mass



(a) (b)

Figure 4: Foraging performance of 25-robot swarms in the (a) Heap1 and (b) Scatter25 scenarios using various $p(S)$ values and $d_C = 0$ m (solid line), $d_C = 0.6$ m (dashed line) and $d_C = 5$ m (dotted line). Each point represents mean percentage of available resource that was collected in a given scenario, collated over 50 one-hour-long runs for each of the deposit distances in the set $D = \{7, 9, 11, 13\}$ m. The whiskers represent 95% confidence intervals.

recruitment once the deposit was discovered and foragers returned to the base to convey the deposit location to any unsuccessful scouts, making long-range recruiters the most suitable option (Figure 4a). In the other extreme scenario, Scatter25, where there were 25 small deposits, solitary foraging led to the best results (Figure 4b). In this case, recruitment damaged performance because of environmental interference between robots, where the robots altered each other's foraging environment by depleting low-volume deposits and where recruitment thus led to a lot of wasteful foraging trips (see Pitonakova et al., 2014, for discussion of robot-robot interference during foraging). Short-range recruiters could achieve a balance between exploration and exploitation, which made them more robust across scenarios, although they never achieved performance better than that of the other two swarm types (see the dashed line in Figure 4).

The results shown in Figure 4 were obtained using swarms of 25 robots. Similar relationships between d_C and $p(S)$ were found for swarms of 15 and 35 robots. The findings also held for less extreme environments such as Heap2, Heap4 or Scatter10, although results varied between runs more in these intermediate scenarios.

5. Heterogeneous deposit quality

Having established that recruitment is most beneficial in Heap scenarios, where deposits are large but hard to find, we now explore what type of information transfer would allow the robots to collectively choose a deposit of a better energy efficiency. We examined the following scenarios using deposit distances $D = \{7, 9, 11, 13\}$ m:

- **Heap2A:** two deposits with volume $V = 50$ each and with qualities $Q_1 = 0.5$, $Q_2 = 1.5$
- **Heap2B:** two deposits with $V = 50$ each and with qualities $Q_1 = 0.1$, $Q_2 = 1.9$
- **Heap4A:** four deposits with $V = 25$ each and with qualities $Q_{1-3} = 0.5$, $Q_4 = 2.5$
- **Heap4B:** four deposits with $V = 25$ each and with qualities $Q_1 = 0.1$, $Q_2 = 0.5$, $Q_3 = 1.5$, $Q_4 = 1.9$

The maximum volume $L_{\max}$ loaded by a robot per foraging trip was decreased from 1.0 to 0.25 in all experiments reported below. The robots still collected $L_{\max} \times Q$ units of net resource value per foraging trip and deposited a maximum of four pellets in the unloading area. Decreasing $L_{\max}$ causes swarms to carry out more foraging trips during a simulation (see online supplementary material, Figure S1), generating a larger data set which is necessary in order to analyse foraging dynamics when the environment changes over time.

The swarms used in the previous experiments were unable to discriminate between deposits based on energy efficiency as robots always followed the first recruiter that they met and remained foraging from a

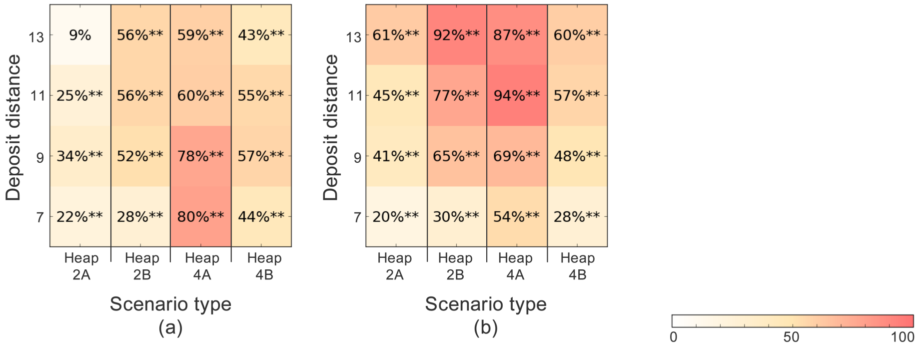


Figure 5: Difference in the amount of resource collected compared to control (non-switching) swarms using 25 beggers with (a) short-range, (b) long-range signals during one simulated hour. The measured values represent averages from 50 independent runs. Statistically significant differences are indicated with asterisks (Wilcoxon signed-rank test, $** = p < 0.01$, $* = p < 0.05$).

deposit until it was fully depleted. On average, such swarms thus distributed themselves to forage from all deposits equally. They were used as control swarms in the following experiments.

In order to allow robots to preferentially concentrate their foraging effort on deposits with higher energy efficiency, we created swarms of ***beggers***, where all robot states and state transitions remained the same, but where recruiters asked each other about the energy efficiency of advertised deposits. If another robot's deposit had higher E_E, a recruiter switched to advertise this deposit and then foraged from it. This strategy was inspired by the begging behaviour of bees described in Section 2.1[1]. As before, we tested beggers with both short-range (0.6 m) and long-range (5 m) communication signals.

Swarms of short-range beggers were able to identify and concentrate on better deposits within the first 20 minutes and could thus collect more resource than the control swarms (Figure 5a). Long-range beggers switched to better deposits faster, making them more efficient at the task, especially for larger deposit distances (Figure 5b). The size of the difference between deposit qualities did not affect the time it took to switch to a better deposit given a particular number of deposits. However, performance of both swarms was better in the Heap2B scenarios compared to Heap2A, as the difference between deposit qualities was larger in Heap2B and it was thus more advantageous to forage from the better deposit. Similarly, when robots could concentrate on a deposit with quality 2.5 in Heap4A, they performed better than in Heap4B, where the best deposit had quality of only 1.9.

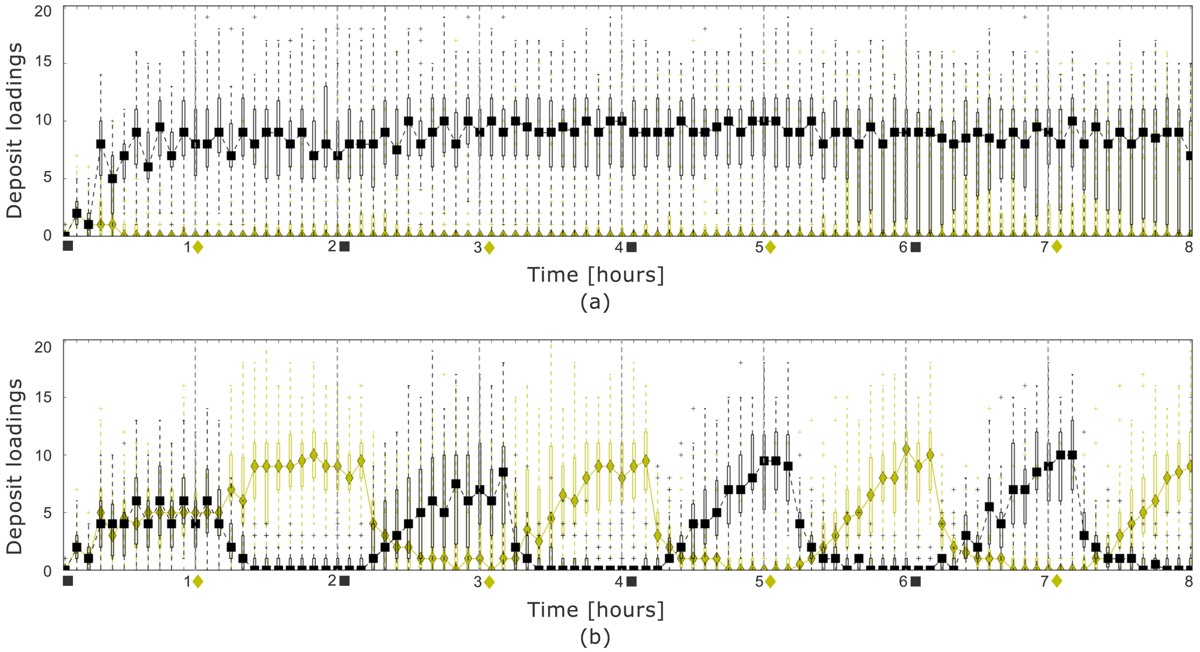
6. The ability of robots to repeatedly switch between deposits

In biological settings such as nectar foraging, the quality of foraging sites changes over time. In robotic applications, a preference for where to forage may also need to change due to the nature of the task, orders from humans, etc. It is thus important not only to be able to collectively select a better foraging site, but to be able to do so repeatedly, as the environment changes. Therefore, in dynamic environments, the ability to achieve a balance between exploitation and exploration becomes important.

To test the ability of our swarms to preferentially forage from better deposits in dynamic environments over a long period of time, we ran simulations over a number of hours and changed deposit qualities (but not their locations) at the end of each *deposit quality change interval* of length T_Q. When the change was

---

[1] The begging behaviour of our robots is simplified compared to that of bees. Bees integrate information received from multiple nest mates and their foraging preferences can be affected for a number of days (De Marco and Farina, 2001; Farina et al., 2005).



(a)

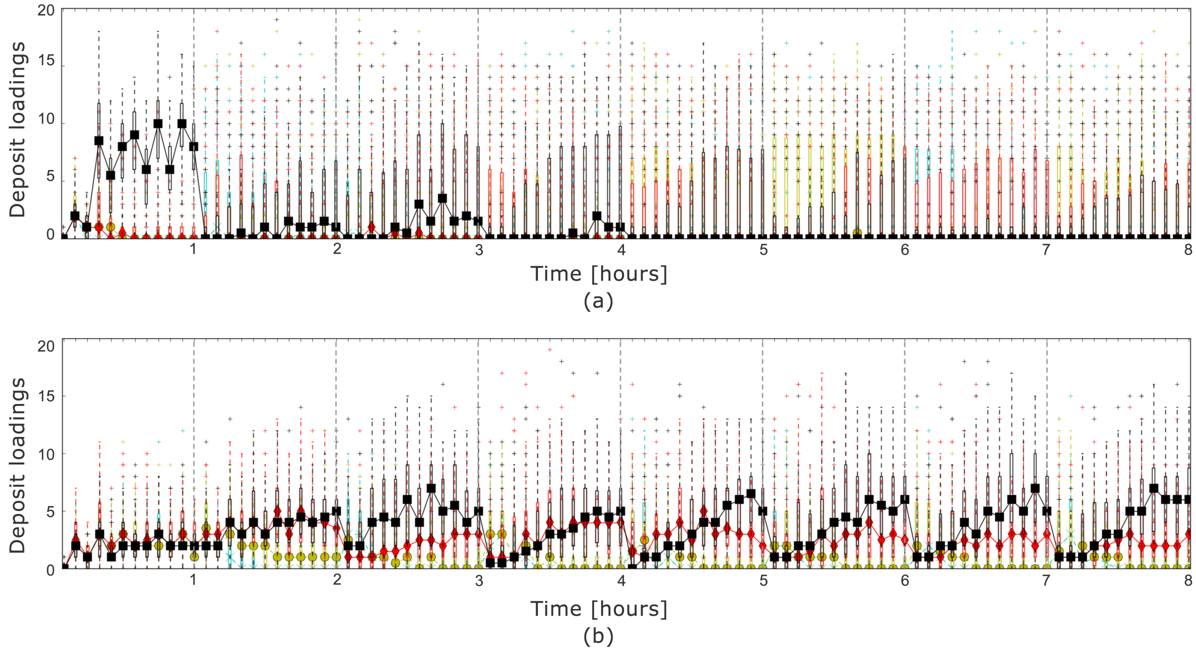
(b)

Figure 6: Median number of loadings from two deposits (square and diamond symbols) during the first eight hours of simulation using (a) 25 short-range beggers and (b) 25 short-range checkers in the Heap2B scenario with $D = 9$ m, $T_Q = 1$ h. Deposit qualities were exchanged every hour during the experiment. The symbol of the deposit with higher quality is shown along the time axis at the beginning of each hour. Each data point represents a median value for a particular time interval and is based on a set of results collected from 50 independent runs. A data point is surrounded by a box, representing the inter-quartile range or "middle fifty" of the result set, and whiskers representing the "middle 97", with outliers outside this range shown as plus signs.

made, deposits were assigned new qualities from a quality set given for a particular scenario and their volumes were replenished, so that the amount of available resource was the same at the beginning of each quality change interval.

In this task, strong commitment to a single best deposit could be a serious problem, as it might cause the swarm to loose track of other deposits and of new places to forage from when the quality change occurs. This was the case in all Heap2 scenarios, where beggers locked into foraging from a single deposit (Figure 6a), an effect that was more pronounced for long-range beggers, and that held for both short ($T_Q = 1$ h) and long ($T_Q = 2$ h) quality change intervals.

On the other hand, in all Heap4 scenarios, where deposits had only half of the volume compared to the Heap2 scenarios, and the differences between their energy efficiencies were thus smaller, the resulting foraging pattern depended on the communication range of robots and the length of the quality change interval used (T_Q). Since beggers usually rapidly recruited each other to the best deposit found, large groups of robots quickly exploited a single foraging site until its energy efficiency fell below that of alternative deposits, at which point the robots rapidly switched to a different deposit. This repeated every time a certain volume of a deposit that was currently being exploited by a majority of robots was depleted, preventing the swarms from preferentially foraging from deposits with superior quality and, on average, to forage from all deposits with an equal probability (Figure 7a). Unlike in Heap2, some members of the swarm always foraged from inferior deposits and the swarm thus retained the memory of all deposits in the environment. However, the inability to for-



(a)

(b)

Figure 7: Median number of loadings from four deposits during the first eight hours of simulation using (a) 25 short-range beggers and (b) 25 short-range checkers in the Heap4B scenario with $D = 9$ m, $T_Q = 1$ h, based on 50 independent runs. Deposit qualities were assigned randomly every hour during the experiment. Therefore, instead of identifying loadings from a particular deposit in a particular location, like in Figure 6, each symbol identifies loadings from a deposit of a particular quality in a given time interval: $Q = 1.9$ (squares), $Q = 1.5$ (diamonds), $Q = 0.5$ (circles), $Q = 0.1$ (crosses).

age from deposits of superior quality for sufficiently long caused short-range beggers when $T_Q = 1$ h and long-range beggers when $T_Q = 1$ h and 2 h to collect less resource that the control swarms (Figures 8a,b and 9a,b). On the other hand, when $T_Q = 2$ h, deposit quality remained stable long enough, allowing short-range beggers to eventually choose the best deposit after a period of indecisiveness, and more importantly to repeat this pattern during each quality change interval. This led to foraging performance better than that of the control swarms in both Heap4A and Heap4B scenarios when deposits were near the base (Figure 9a).

These results suggested that a swarm behaviour where the spread of social information is regulated could potentially lead to more effective plastic reconfiguration and consequently more sustainable preferential foraging. To test this hypothesis, we introduced a modified behaviour that explicitly regulated the spread of social information. A swarm of ***checkers*** was implemented, where the robots did not compare deposit energy efficiency directly with other recruiters, as it was the case with beggers. However, instead of having a binary recruitment probability, $p(R)$ (Equation 2), and a fixed recruitment time, T_R, checkers calculated these variables based on how the energy efficiency of a deposit had changed since the last time they had visited it:

$$\delta = \min\left(1.0, \frac{E_E}{E'_E}\delta'\right) \tag{3}$$

where E'_E and δ' were the energy efficiency of the deposit and δ measured during the last loading event,

$$p(R) = \delta \tag{4}$$

$$T_R = \delta \times 120s \tag{5}$$

Because their recruitment probability, and consequently their probability of returning to a deposit,

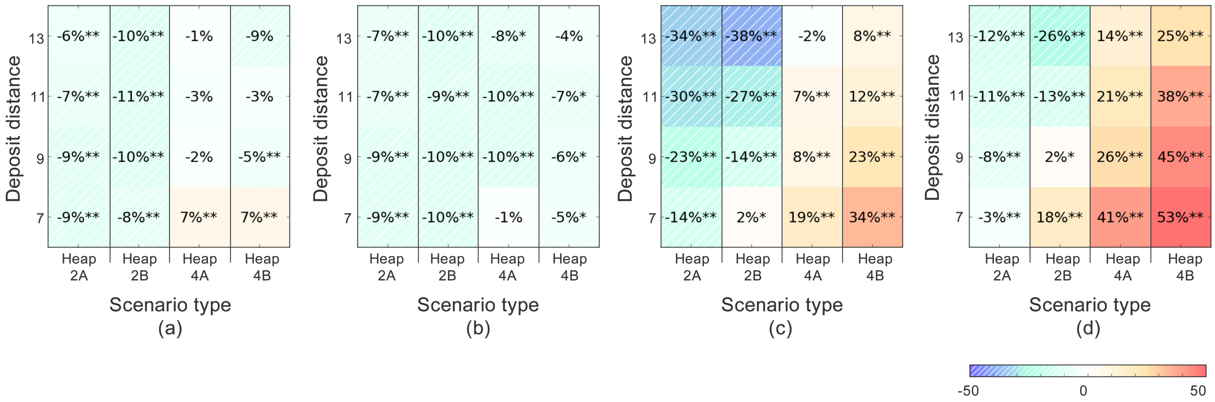


Figure 8: Difference in the amount of resource collected compared to control (non-switching) swarms using (a) 25 short-range beggers, (b) 25 long-range beggers, (c) 25 short-range checkers (d) 25 long-range checkers during 1 h $< T <$ 13 h of 50 independent runs, using $T_Q = 1$ h. Statistically significant differences are indicated with asterisks (Wilcoxon signed-rank test, $** = p < 0.01$, $* = p < 0.05$).

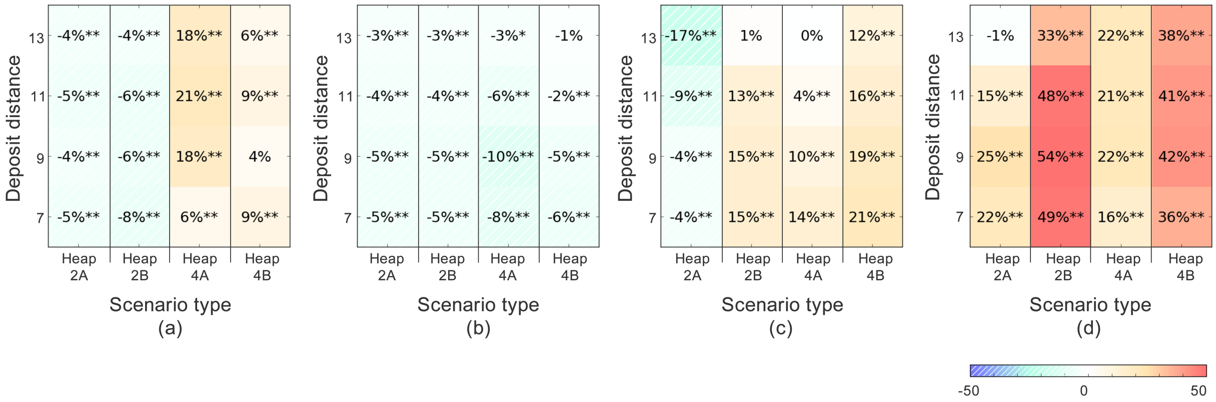


Figure 9: Difference in the amount of resource collected compared to control (non-switching) swarms using (a) 25 short-range beggers, (b) 25 long-range beggers, (c) 25 short-range checkers (d) 25 long-range checkers during 2 h $< T <$ 26 h of 50 independent runs, using $T_Q = 2$ h. Statistically significant improvements are indicated with asterisks (Wilcoxon signed-rank test, $** = p < 0.01$, $* = p < 0.05$).

decreased gradually as the deposit became depleted, checkers abandoned a foraging site one by one over time, allowing them to be recruited by other robots or to go scouting. By contrast, both the control swarms and beggers remained foraging from the same deposit until it became completely depleted or until, in the case of beggers, the robots received social information about a better deposit. Perhaps more importantly, the fact that checkers evaluated the change in energy efficiency of a deposit each time they visited it, enabled them to immediately abandon a deposit if its quality suddenly decreased, without having to rely on receiving knowledge from other robots about better deposits, as the beggers did.

The checker behavioural strategy was also inspired by bees, in particular by their food patch abandonment behaviour (Seeley, 1994; Biesmeijer and De Vries, 2001). An agent-based model, where recruitment time was influenced by quality of an advertised site but where agents, similarly to our beggers, could also switch to another site after recruitment was explored by (Valentini et al., 2014).

The checker strategy allowed the swarms to repeatedly switch to a better deposit in both Heap2 (Figure 6b) and Heap4 (Figure 7b) scenarios, unlike the beggers that either foraged from one deposit throughout the whole run or were often unable to make a clear collective decision. The checker swarms were able to concentrate on exploiting better deposits faster when deposit distance was smaller (see online supplementary material, Figures S2 and S3), as deposit visits, as well as recruitment, occurred more frequently (for example in Heap2B, the swarms switched to a better deposit after about 25 minutes when $D = 7$ m and after about 35 minutes when $D = 13$ m). However, note that checker swarms were unable to concentrate on the better deposit before the first quality change occurred, as a majority of robots had to abandon the deposit from which they initially foraged before recruitment to a better foraging site could take place. Such a significant abandonment of a deposit could only occur if the deposit was nearly depleted, which did not occur unless the whole swarm foraged from it, or when deposit qualities were changed suddenly by the simulation engine.

Nevertheless, the long-term behaviour of the checkers was much more desirable than that of the beggers and could significantly outperform the control swarms, especially when four deposits were placed in the environment, when the deposit quality change interval was long, or when long-range recruitment signals were used (Figures 8c,d and 9c,d). However, in some scenarios, the checkers collected less resource than the control swarms. This was the case in all Heap2 scenarios when $T_Q = 1$ h for both short- and long-range checkers and in the Heap2A scenario when $T_Q = 2$ h for short-range checkers. The poor performance was caused by the fact that the checkers gradually abandoned a deposit of a worse quality after deposit qualities were changed, which led to a decrease in foraging activity. The robots that stopped foraging waited on the dance floor to be recruited or left the base to become scouts, creating a period of time when the swarm could not do any foraging work. Even though the robots eventually did forage from a deposit of a better quality, they often did not have enough time to make up for the loss of foraging time. The period of low foraging activity was especially long for short-range checkers and when the deposit distance was large. For example, it took the swarms up to 35 minutes to start foraging from the better deposit in the Heap2B scenario when $D = 13$ m and $T_Q = 1$ h (see online supplementary material, Figure S2), leading to 38% decrease in the total amount of resource collected compared to the control swarm (Figure 8c).

7. Analysing the value of information

It is clear that foraging performance in dynamic environments is closely related to how information is obtained and exchanged between robots. While sudden bursts of information transfer can lead to over commitment to a single deposit, which reduces the chance of acquiring information on alternative deposit sites, a mix of scouting and more controlled information transfer can lead to plasticity. In order to characterise behavioural strategies that are beneficial, it would be useful to have a measure that could quantify the effect of communication.

It is possible to mathematically model swarms using differential equations and then use the model to test simulation parameters and predict the average result (Lerman et al., 2006; Liu and Winfield, 2010). It has also been shown that decision-making swarms can be modelled as dynamical systems that exhibit Hopf bifurcations and limit cycles (Pais et al., 2012). Finally, information theoretic measures, such as transfer entropy and local data storage, can be used to show how communication affects the state of an agent based on the states of the other agents it shares information with (Miller et al., 2014).

While mathematical models are often tractable and can generalise well (Martinoli et al., 2004), they are also often complex and difficult to apply in real-life scenarios (McFarland and Spier, 1997) where many options exist and where physical interactions substantially influence the resulting swarm behaviour. Inspired by information theoretic approaches, we seek to establish a simple measure, the information value I, that works directly from experimental data and represents a quantifiable effect of information transfer between the environment and the swarm and within the swarm itself. This measure should reflect the flow of new information though the system, should be positive when recruitment leads to exploitation of relatively good deposits and should be negative when recruitment causes the system to lower its foraging performance.

The information value I_r of a robot is defined as

$$I_r = Q_{\text{new}} - Q_{\text{old}} \tag{6}$$

where Q_{new} is the quality of a new deposit the robot finds out about and Q_{old} is the quality of a deposit it previously foraged from. Note that $Q_{\text{old}} = 0$ when a robot has not foraged before. I_r is thus equal to Q_{new} when a scout discovers a new deposit or when a robot that does not posses any information is recruited. In cases when a robot currently foraging from a deposit switches to another, I_r can be either positive or negative. For example, if a robot previously foraged from a deposit with $Q_i = 1.0$ and was recruited to a new deposit with $Q_i = 0.5$, $I_r = -0.5$. Finally, $I_r = 0$ for all robots that do not receive any information in a given time step.

We compare deposit qualities, rather than energy efficiencies, i.e., we do not take the volume left in a deposit or its distance from the base into account. The information value thus only captures utility obtained from a single foraging trip that the robot is guaranteed to make, rather than that of possible future trips. Furthermore, since deposits in our Heap scenarios have the same distance from the base, comparing their qualities is sufficient to identify which one is more profitable.

We obtain the swarm information value I normalised per robot as:

$$I = \sum_{r=1}^{N_R} I_r \times \frac{1}{N_R} \tag{7}$$

where N_R is the total number of robots. A time series of the swarm (Figure 10) is obtained by sampling the average value of I over short time intervals (in our case 300-second intervals). When a swarm working in a dynamic environment is able to repeatedly switch its attention to the current best deposit, the shape of the time series repeats for each quality change interval and has three distinguishable information value regions:

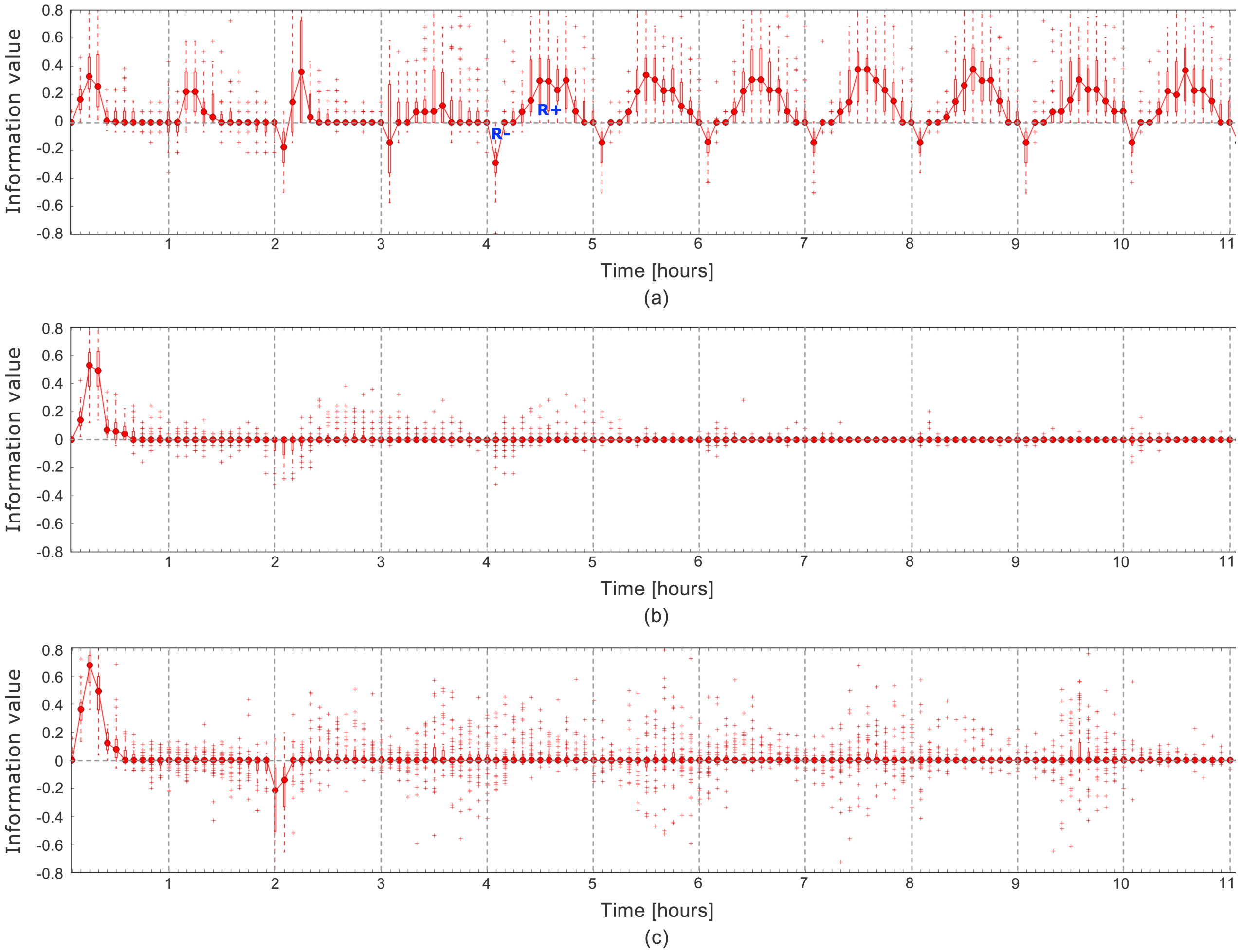
1. **A negative region** that is caused by recruitment to the previously exploited deposit(s) that have become worse since the change in deposit quality.
2. **A flat region** where new and old information coexists in the system. We refer to the negative region and the flat region together as a non-positive region.
3. **A positive region** that represents recruitment to the newly discovered better-quality deposit(s).

The size (i.e., the area between the curve and the abscissa) of negative regions represents a lag between changes in the environment and changes in the swarm's collective knowledge. This region is usually larger for long-range recruiters and for shorter deposit distances, i.e., when recruitment is stronger (see online supplementary material, Figures S4–S6). Stronger recruitment tends to result in a larger number of robots to be foraging from the better deposit at the end of a change interval. Immediately after a quality change, there is thus a higher probability of an observer being recruited by a robot that is signalling out-of-date information.

The size of positive regions is influenced by the number of observers that a recruiter holding new information can reach, i.e., by the speed of information transfer through the swarm. It is also larger for long-range recruiters and shorter deposit distances.

The length of the non-positive regions is a combination of the positive impact of information speed and the negative impact of commitment to old information.

When robots deplete and abandon the better deposit(s) before the quality change occurs, as was the case for checkers in Heap2 scenarios when $T_Q = 2$ h, there is no recruitment to a low-quality deposit after the quality change. Therefore, there are no negative regions of I and non-positive regions become very short. On the other hand, when the whole swarm



(a)

(b)

(c)

Figure 10: An information value time series based on 50 independent runs with 25 robots for (a) switching swarm: long-range checkers in Heap2B, $D = 9$ m, $T_Q = 1$ h (b) locked swarm: long-range beggers in Heap2A, $D = 9$ m, $T_Q = 2$ h (c) indecisive swarm: long-range beggers in Heap4B, $D = 9$ m, $T_Q = 2$ h. Negative regions (R-) and positive regions (R+) are shown for the switching swarm.

commits to a single deposit and is unable to abandon it throughout the experiment, as was usually the case with long-range beggers, only a single large positive region exists at the beginning of the simulation, followed by zero information value throughout the rest of the run. Using information value, we can thus distinguish three work modes of foraging swarms (Figure 10):

1. ***Switching*** swarms that alternate between non-positive and positive regions and sometimes also show negative regions. These swarms are able to concentrate on deposit(s) of better quality in each quality change interval.
2. ***Locked*** swarms that show a large positive region at the beginning of the simulation, but are unable to alter their chosen foraging site for the rest of the run and thus show zero communication value thereafter.
3. ***Indecisive*** swarms that also have a large positive region at the beginning, but are able to gain

> new information through scouting or because recruitment does not affect as many robots as in locked swarms. In contrast with switching swarms, rapid information transfer in indecisive swarms prevents the collective from settling on one solution, leading to oscillations around $I = 0$. When multiple runs are considered, time series of I has medians of zero and many outliers.

When runs that last several hours and have multiple quality change intervals are considered, locked and indecisive swarms always collect less resource than the control swarms that forage from all deposits with similar proportions. The relative performance of switching swarms depends on a number of factors. As we showed in the previous sections, switching to better deposits can sometimes occur too late or the total amount of loadings can be insufficient, which leads to worse performance. This was for example the case for checkers in Heap2 scenarios when $T_Q = 1$ h.

In the following sections, we test our communication strategies further under various experimental perturbations and use the information value measure, I, to analyse the results. We show that the switching work mode is more common when information transfer is slower and that measuring I can thus help robot designers to select behavioural parameters that lead to effective plastic self-organisation.

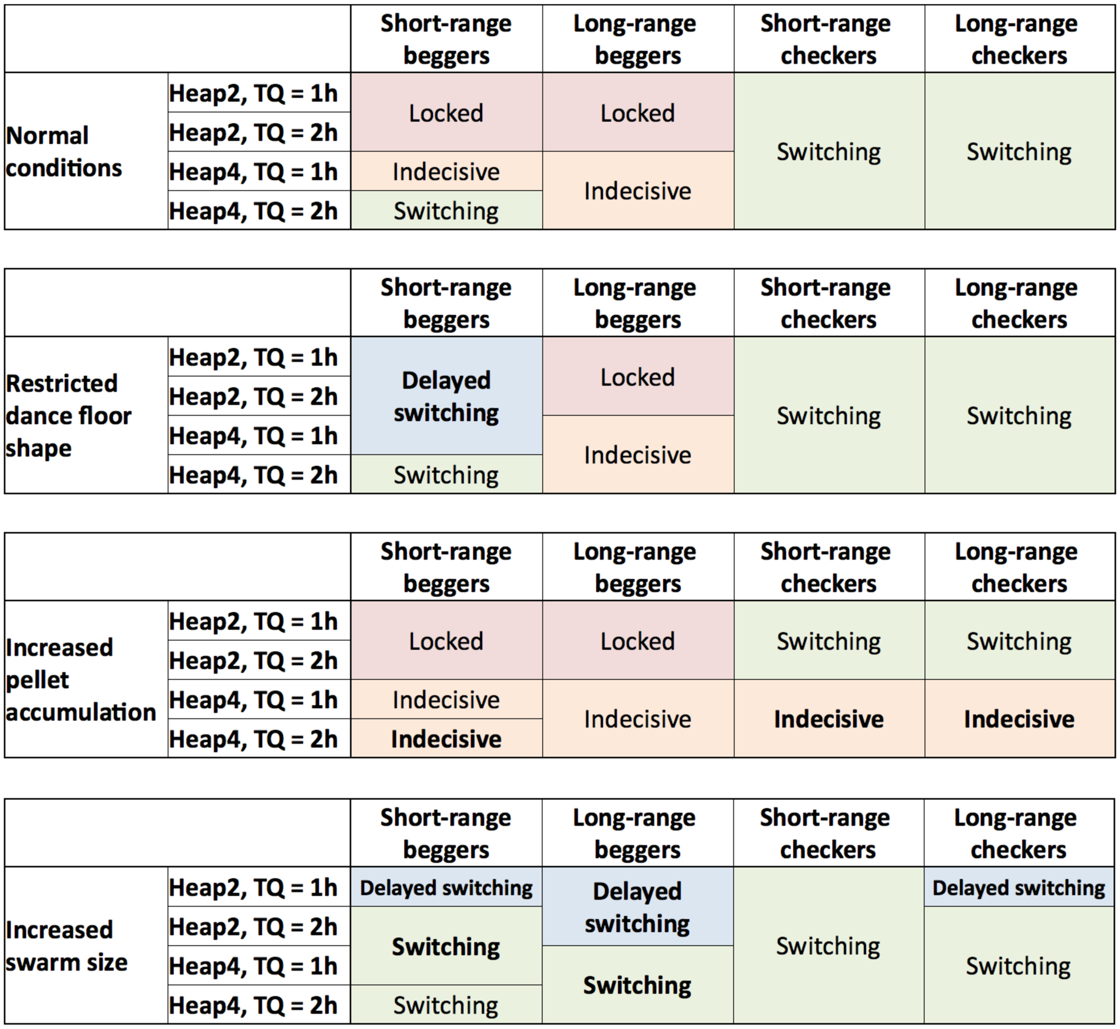
8. Varying experimental parameters

It is important to understand how the chosen experimental parameters affect the performance and work modes of our swarms. In this section we explore the influence of the shape of the dance floor, of the tendency for collected pellets to accumulate and interfere with robot movement, and of the size of the robot swarms. We evaluate how varying these aspects of the foraging scenario affects the behaviour of all four types of swarms (i.e., short-range and long-range beggers, and short-range and long-range checkers) compared to performance under normal conditions (i.e., circular dance floor, no pellet accumulation, and 25-robot swarms – see Sections 5 and 6). We also identify cases in which variation to these aspects of the scenario causes a swarm to transition from one work mode to another (e.g., from switching to indecisive). See Figure 11 for a summary of all experiments and the work modes exhibited by the swarms.

8.1. Restricted dance floor shape

In the previous experiments, a returning forager traveled to the middle of the base before it started recruiting in order to be able to interact with any observer, regardless of direction from which it returned to the base. However, the constraints of a particular foraging scenario might require a different approach. For example, we might want to create a recharging area in a part of the recruitment area or otherwise change the shape of the base and thus indirectly alter the way in which robots meet each other and communicate. Information flow throughout a swarm will tend to be influenced by such physical logistics, potentially changing the ability of the swarm to maintain plasticity and thereby influencing performance levels.

In the following experiment, we designated an inaccessible circular area in the middle of the base, making the dance floor doughnut-shaped and 35 cm thick. While the robots could still see other robots across the inaccessible central area, and could thus communicate with them if their communication range allowed for it, they could not move through the central restricted area. Consequently, while undertaking a random walk within the dance floor, a robot tended to remain relatively close to the point at which it first entered the base. The restricted shape of the dance floor also resulted in higher congestion and constrained movement of the robots within and immediately around the dance floor. The congestion was more severe when long-range recruitment was used or when deposits were closer to the base, i.e., when the number of foragers returning to the base at the same time was high. Additionally, the annular shape of the dance floor restricted communication within swarms with short-range recruitment, as it made interactions between robots that foraged



		Short-range beggers	Long-range beggers	Short-range checkers	Long-range checkers
Normal conditions	**Heap2, TQ = 1h**	Locked	Locked	Switching	Switching
	Heap2, TQ = 2h				
	Heap4, TQ = 1h	Indecisive	Indecisive		
	Heap4, TQ = 2h	Switching			

		Short-range beggers	Long-range beggers	Short-range checkers	Long-range checkers
Restricted dance floor shape	**Heap2, TQ = 1h**	**Delayed switching**	Locked	Switching	Switching
	Heap2, TQ = 2h				
	Heap4, TQ = 1h		Indecisive		
	Heap4, TQ = 2h	Switching			

		Short-range beggers	Long-range beggers	Short-range checkers	Long-range checkers
Increased pellet accumulation	**Heap2, TQ = 1h**	Locked	Locked	Switching	Switching
	Heap2, TQ = 2h				
	Heap4, TQ = 1h	Indecisive	Indecisive	**Indecisive**	**Indecisive**
	Heap4, TQ = 2h	**Indecisive**			

		Short-range beggers	Long-range beggers	Short-range checkers	Long-range checkers
Increased swarm size	**Heap2, TQ = 1h**	**Delayed switching**	**Delayed switching**	Switching	**Delayed switching**
	Heap2, TQ = 2h	**Switching**			Switching
	Heap4, TQ = 1h		**Switching**		
	Heap4, TQ = 2h	Switching			

Figure 11: Work modes exhibited by the swarms under various experimental setups. Work modes that differ from those exhibited under normal conditions (see text) are shown in bold.

from different areas of the environment less probable and prevented information from spreading as easily throughout the swarm.

Although most swarms did not change their work modes, information value analysis showed that restricting dance floor shape decreased the rate at which new information about the best deposit(s) spread through the swarm (see online supplementary material, Figures S7 and S8). For example, the total size of positive regions (i.e., sum of all $I > 0$) was 40% smaller for short-range checkers in Heap2s ($D = 9$ m and $T_Q = 1$ h) compared to normal conditions. The size of positive regions was less affected for long-range checkers (only 28% smaller), but higher congestion caused the positive regions to appear much later than under normal conditions (usually after 35 instead of 25 minutes).

Short-range beggers were the only swarm that exhibited different work modes compared to the normal conditions (Figure 11). The decreased probability of interactions between recruiters and observers on the dance floor caused the swarms to escape the

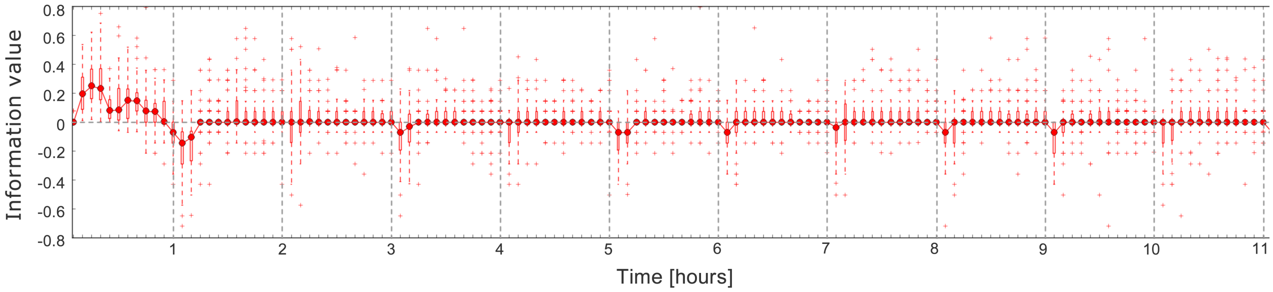
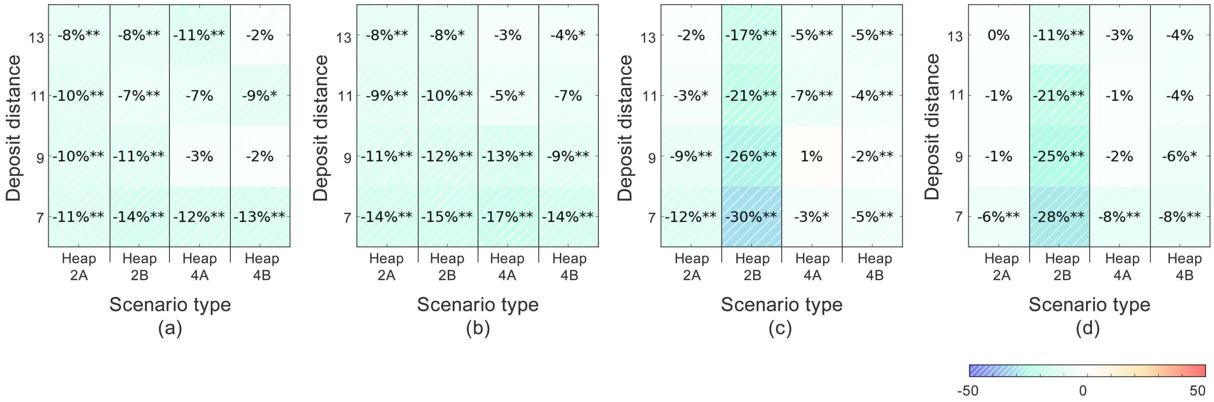


Figure 12: An information value time series based on 50 independent runs for a delayed switching swarm of 25 short-range beggers in Heap2B, $D = 9$ m, $T_Q = 1$ h with restricted dance floor shape.



(a)

Deposit distance	Heap 2A	Heap 2B	Heap 4A	Heap 4B
13	-8%**	-8%**	-11%**	-2%
11	-10%**	-7%**	-7%	-9%*
9	-10%**	-11%**	-3%	-2%
7	-11%**	-14%**	-12%**	-13%**

(b)

Deposit distance	Heap 2A	Heap 2B	Heap 4A	Heap 4B
13	-8%**	-8%*	-3%	-4%*
11	-9%**	-10%**	-5%*	-7%
9	-11%**	-12%**	-13%**	-9%**
7	-14%**	-15%**	-17%**	-14%**

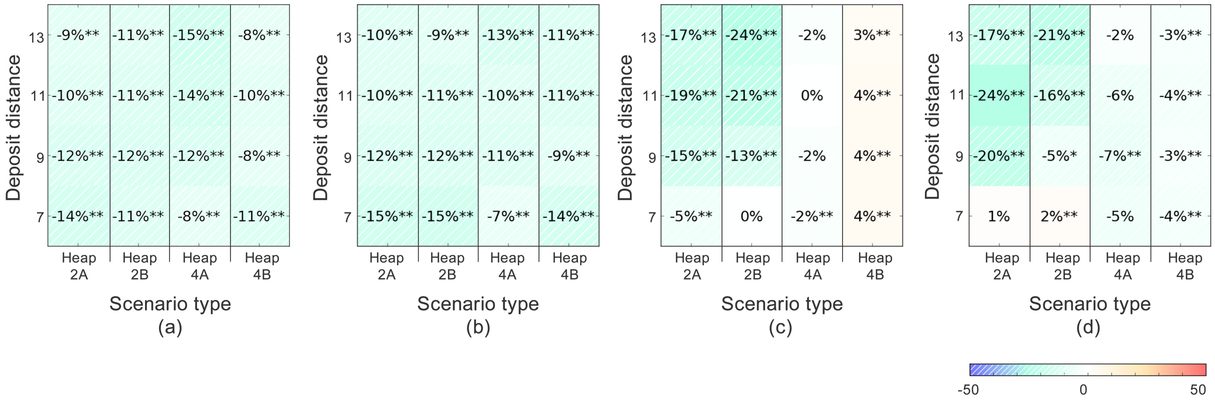
(c)

Deposit distance	Heap 2A	Heap 2B	Heap 4A	Heap 4B
13	-2%	-17%**	-5%**	-5%**
11	-3%*	-21%**	-7%**	-4%**
9	-9%**	-26%**	1%	-2%**
7	-12%**	-30%**	-3%*	-5%**

(d)

Deposit distance	Heap 2A	Heap 2B	Heap 4A	Heap 4B
13	0%	-11%**	-3%	-4%
11	-1%	-21%**	-1%	-4%
9	-1%	-25%**	-2%	-6%*
7	-6%**	-28%**	-8%**	-8%**

Figure 13: Difference in the amount of resource collected using an annular dance floor compared to the same communication strategy operating over a full dance floor with (a) 25 short-range beggers (b) 25 long-range beggers, (c) 25 short-range checkers, (d) 25 long-range checkers during 1 h $< T <$ 13 h of 50 independent runs, using $T_Q = 1$ h. Statistically significant differences are indicated with asterisks (Wilcoxon signed-rank test, $** = p < 0.01$, $* = p < 0.05$).



(a)

Deposit distance	Heap 2A	Heap 2B	Heap 4A	Heap 4B
13	-9%**	-11%**	-15%**	-8%**
11	-10%**	-11%**	-14%**	-10%**
9	-12%**	-12%**	-12%**	-8%**
7	-14%**	-11%**	-8%**	-11%**

(b)

Deposit distance	Heap 2A	Heap 2B	Heap 4A	Heap 4B
13	-10%**	-9%**	-13%**	-11%**
11	-10%**	-11%**	-10%**	-11%**
9	-12%**	-12%**	-11%**	-9%**
7	-15%**	-15%**	-7%**	-14%**

(c)

Deposit distance	Heap 2A	Heap 2B	Heap 4A	Heap 4B
13	-17%**	-24%**	-2%	3%**
11	-19%**	-21%**	0%	4%**
9	-15%**	-13%**	-2%	4%**
7	-5%**	0%	-2%**	4%**

(d)

Deposit distance	Heap 2A	Heap 2B	Heap 4A	Heap 4B
13	-17%**	-21%**	-2%	-3%**
11	-24%**	-16%**	-6%	-4%**
9	-20%**	-5%*	-7%**	-3%**
7	1%	2%**	-5%	-4%**

Figure 14: Difference in the amount of resource collected using an annular dance floor compared to the same communication strategy operating over a full dance floor with (a) 25 short-range beggers (b) 25 long-range beggers, (c) 25 short-range checkers, (d) 25 long-range checkers during 2 h $< T <$26 h of 50 independent runs, using $T_Q = 2$ h. Statistically significant differences are indicated with asterisks (Wilcoxon signed-rank test, $** = p < 0.01$, $* = p < 0.05$).

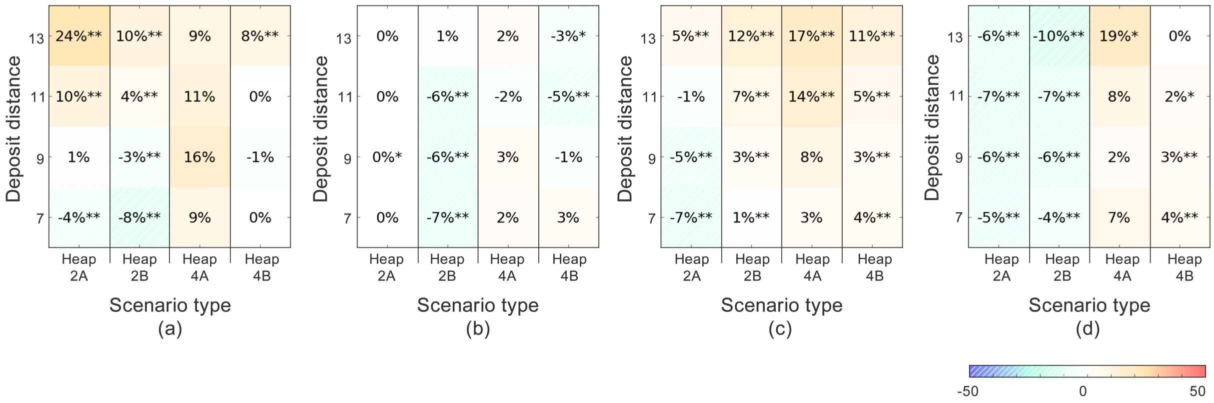


Figure 15: Improvement of resource collected using 25 long-range checkers compared to 25 short-range checkers during 50 independent runs with (a) unloading area handling time $t_H = 10$ s, $T_Q = 1$ h and 1 h $< T <$ 13 h, (b) $t_H = 20$ s, $T_Q = 1$ h and 1 h $< T <$ 13 h, c) $t_H = 10$ s, $T_Q = 2$ h and 2 h $< T <$ 26 h, d) $t_H = 20$ s, $T_Q = 2$ h and 2 h $< T <$ 26 h. Statistically significant improvements are indicated with asterisks (Wilcoxon signed-rank test, $** = p < 0.01$, $* = p < 0.05$).

locked mode they experienced with a full dance floor in Heap2 scenarios, allowing them to preferentially forage from better deposits. However, clear positive regions of information value were absent while negative regions remained, meaning that the information about new better deposits spread too slowly to be of use. We term this a *delayed switching* work mode (Figure 12). The swarms also changed their work mode from indecisive to delayed switching in Heap4 scenarios with $T_Q = 1$ h, as constrained information transfer prevented the confusion that had previously resulted from strong competition between deposits with equivalent energy efficiencies. Despite the fact that, when in the delayed switching mode, short-range beggers were able to alternate between deposits as the deposit quality changed, the foraging performance of the swarms deteriorated compared to normal conditions (Figure 13a and 14a). First, the total number of foraging trips was smaller due to congestion caused by a smaller dance floor. Second, the delayed switching mode led to a similar number of loadings from deposits of higher and lower quality when the whole run was considered, making the proportion of loadings from the better deposit(s) similar to when the swarms were in the locked or indecisive modes (see online supplementary material, Figures S9 and S10).

The impact of congestion caused by a restrictive dance floor was most clearly observed for long-range beggers. While a recruiter's signal could reach any observer on the dance floor regardless of their relative positions and the work mode of the swarms thus remained locked, foraging performance still deteriorated as the robots spent significantly more time avoiding each other (Figures 13b and 14b).

While the restricted movement of recruiters affected both the short- and long-range checkers negatively in almost all scenarios (Figures 13c,d and 14c,d), it had a surprisingly positive effect on short-range checkers in Heap4B scenarios when $T_Q = 2$ h (Figure 14c). In this case, the impaired ability of robots to recruit each other caused the swarms to forage from the best two deposits in parallel and thus to collect a total amount of resource that was higher than under normal conditions.

8.2. Pellet accumulation

The material collected by robots in the previous experiments disappeared instantly when it was placed in the unloading bay, i.e., the unloading area handling time $t_H = 0$ s. However, in real world applications, material would have to be stored somewhere for later use by humans or other robots. It is therefore reasonable to assume that it would accumulate and thus affect the swarm's ability to collect more. To test performance of our swarms under such conditions, we set $t_H = \{10, 20\}$ s, i.e., pellets dropped by robots could accumulate in the unloading area and create congestion. Since the congestion prevented robots from foraging with maximum efficiency, and from reaching the dance floor in order to communicate, swarms always collected less resources than under normal conditions.

When pellets accumulated, both the negative and positive regions of information value were more flat, while the non-positive regions usually became longer, indicating that limited access to and from the base prevented information transfer. This effect was observed more strongly for long-range recruiters and the higher value of t_H (see online supplementary material, Figures S11 and S12), i.e., when more foragers returned to the base at the same time and when pellets remained in the unloading area for a longer period of time. For example, when $t_H = 20$ s, the total size of positive regions (i.e. sum of all $I > 0$) in Heap2s ($D = 9$ m and $T_Q = 1$ h) was 67% smaller for short-range checkers and 75% smaller for long-range checkers compared to the normal conditions. The length of non-positive regions decreased by 25% for short-range checkers and increased by 60% for long-range checkers.

Since the rate at which the robots communicated was affected, the ability of swarms to switch between deposits was impaired. This caused short-range beggers, short-range checkers and long-range checkers to become indecisive in Heap4 scenarios (Figure 11). There were no significant changes in work modes observed for long-range beggers, as they already exhibited the locked (in Heap2) and indecisive (in Heap4) work modes under normal conditions.

Pellet accumulation had also an interesting effect on the advantage of long-range over short-range recruitment. For example, under normal conditions, long-range checkers always outperformed short-range checkers in terms of the total amount of resource collected due to their ability to recruit to the best deposit faster. However, when pellets accumulated, this faster recruitment resulted in stronger congestion and in subsequent worse performance compared to short-range checkers in some scenarios (Figure 15). This was especially true in the Heap2 scenarios when the unloading area handling time $t_H = 20$ s and when the swarms were given a long time to forage from the selected deposit ($T_Q = 2$ h) (Figure 15d).

8.3. Increased swarm size

All previous experiments were performed with swarms of 25 robots. Here we simulate swarms of 45 and 65 robots.

While short- and long-range beggers previously exhibited the locked work mode in Heap2 scenarios, higher scouting success of the swarm, caused by a higher number of robots that were scouting at the beginning of a simulation run, led to a more even distribution of robots between the deposits. This allowed larger swarms to preferentially forage from better deposits and exhibit the delayed switching work mode, or the switching work mode in the case of short-range beggers when $T_Q = 2$ h or when $N_R = 65$ (see Figure 11 and online supplementary material, Figure S13). However, the significant delays in recruitment to a better deposit usually caused a stronger disadvantage for the beggers over the control swarms compared to when the number of robots $N_R = 25$ (compare Figure 16a, b with Figure 8a,b and Figure 17b with Figure 9b). In Heap4 scenarios, where the beggers were previously indecisive, the swarms were able to achieve the switching mode regardless of the value of T_Q (Figure 11). The switching was effective enough to allow both short- and long-range beggers to outperform the control swarms when $N_R = 45$. For example, short-range beggers collected 17% more resources than the control swarms of the same size in Heap4A when $D = 7$ m and $N_R = 45$. However, their advantage largely disappeared when $N_R = 65$ (Figures 16a and 17a) due to rapid exploitation of the better foraging sites and consequent ineffective foraging from other deposits, as well as due to increased congestion in the base. The disadvantage of 65-robot swarms was more pronounced for long-range beggers (Figures 16b and 17b) that shared information with each other faster and thus exploited deposits more rapidly.

Checkers generally retained their ability to switch between deposits when the swarm size increased. The information value of the swarms, normalised per robot, did not change significantly, although the positive regions at the beginning of simulation runs

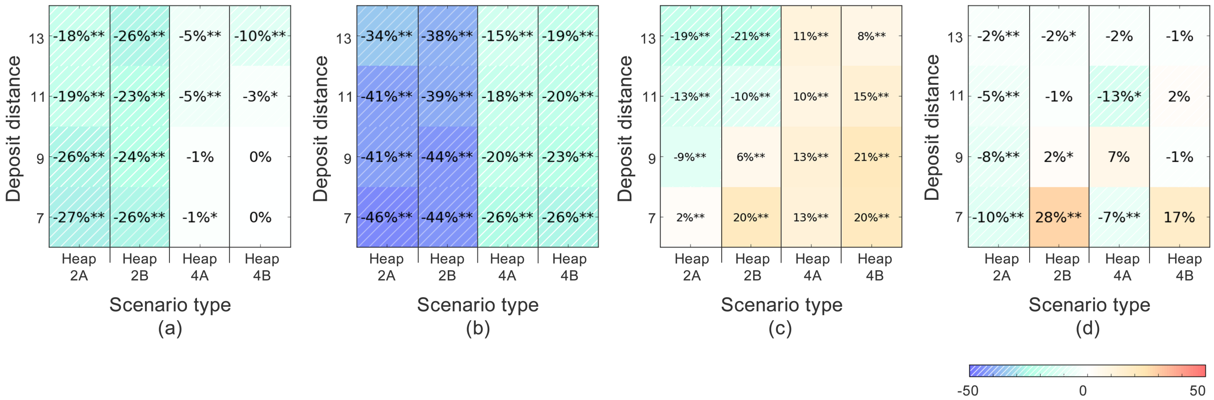


Figure 16: Improvement of resource collected compared to control (non-switching) swarms of 65 robots using (a) 65 short-range beggers, (b) 65 long-range beggers, (c) 65 short-range checkers (d) 65 long-range checkers during 1 h $< T <$ 13 h of 50 independent runs, $T_Q = 1$ h. Statistically significant improvements are indicated with asterisks (Wilcoxon signed-rank test, $** = p < 0.01$, $* = p < 0.05$).

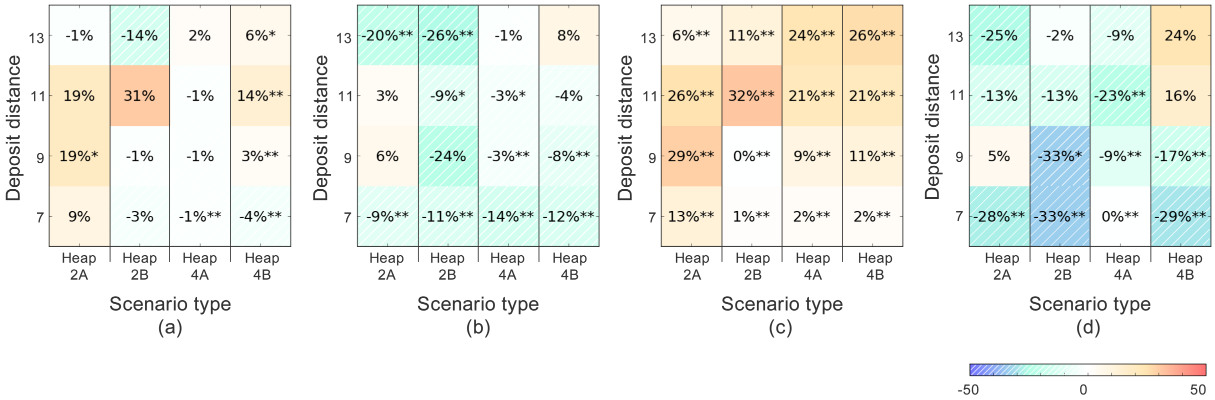


Figure 17: Improvement of resource collected compared to control (non-switching) swarms of 65 robots using (a) 65 short-range beggers, (b) 65 long-range beggers, (c) 65 short-range checkers (d) 65 long-range checkers during 2 h $< T <$ 26 h of 50 independent runs, $T_Q = 2$ h. Statistically significant improvements are indicated with asterisks (Wilcoxon signed-rank test, $** = p < 0.01$, $* = p < 0.05$).

were more prominent compared to when 25 robots were used (see online supplementary material, Figure S14). This suggests that information spread faster at the beginning of a run, causing the swarms to be able to exhibit periodic switching behaviour sooner. As with beggers, checkers performed better than the control swarms when $N_R = 45$. Long-range checkers lost their advantage when $N_R = 65$ for similar reasons to the beggers (Figures 16d and 17d). It was also observed that 65-robot swarms of long-range checkers exhibited the delayed switching work mode instead of the switching work mode in Heap2 scenarios when $T_Q = 1$ h as a result of very significant commitment to a single deposit in each quality change interval and consequent insufficient scouting. Furthermore, there was a large variance in the total amount of collected resource across multiple runs when long-range checkers were used and when $N_R = 65$, $T_Q = 2$ h. On the other hand, large swarms of short-range checkers, where information could not spread as quickly, performed well in most of the tested scenarios (Figures 16c and 17c). This result is not that surprising. If there is an optimal number of robots to find and exploit resources in a given environment, then swarms with long-range communication, where information spreads faster, reach that optimal number sooner and thus also become disadvantaged sooner as N_R increases.

9. Discussion

9.1. Summary of results

We have demonstrated that the nature of communication between individual robots affects a swarm's plasticity and thereby its ability to forage effectively in various types of environment. In particular, the following principles apply:

- Recruitment is most useful when deposits are large and hard to find. Conversely, recruitment can harm foraging performance when resource is distributed over numerous small deposits (see Figure 4).
- Longer communication range increases foraging performance when deposits are large and hard to find (see Figure 4), unless rapid exploitation of deposits can cause congestion, e.g., when a lot of the collected material accumulates in the base (see Figure 15d) or when swarm size is large (see Figures 16b,d and 17b,d).
- If robots need to choose foraging sites based on deposit energy efficiency in *static* foraging environments, maximising information flow in the swarm is beneficial as it maximises exploitation of the best sources (see Figure 5).
- When environments are *dynamic* and deposit qualities change over time, both continuous exploitation and exploration of the environment are important. Regulation of information flow, for example by short communication range of robots or by an implementation of individual-based decisions on when to ignore social information, is beneficial as it prevents overexploitation of a single foraging site (see Figures 8c,d and 9c,d). A slower information flow thus helps to avoid situations in which a swarm locks into foraging from a single deposit, or is unable to discriminate deposits of varying qualities and forages from multiple sources simultaneously.
- Inhibition of social information spread can also be achieved when robots are to some extent physically prevented from meeting and recruiting each other, for example when the shape of the base prevents robots from communicating with those that approach from the opposite side (see online supplementary material, Figures S7 and S8), or when collected material accumulates in the base (see online supplementary material, Figures S11 and S12).
- On the other hand, information is collected and spreads faster when swarm size increases (see online supplementary material, Figures S13 and S14). This can be beneficial especially when robot communication is short range (see Figures 16c and 17a,c) or in scenarios where smaller swarms cannot alternate between deposits that change their qualities (see Figure 11).

9.2. Related work

Our results are consistent with existing studies in the literature. For example, it has been shown that communication between robots is most beneficial when deposits are scarce (e.g., Liu et al., 2007; Pitonakova et al., 2014) and that short-range communication is most suitable for large robot groups, as too much information can lower task specialisation (Sarker and Dahl, 2011). Previous work on the effects of communication range also demonstrated that global information exchange leads to over-commitment to a single resource (Tereshko and Loengarov, 2005). Similarly, our swarms of beggers, where information spreads quickly, often found themselves committed to a single source in scenarios with two deposits.

The importance of recruitment when flower patches have a high return has been observed in honey bee colonies (Donaldson-Matasci and Dornhaus, 2012). Bees, like our robots in Heap scenarios, can benefit from communicating about a location that affords many successful foraging trips. The fact that recruitment increases the probability of individuals foraging is also important in our Heap scenarios, where robots found it difficult to discover deposits. Similarly, in an agent-based model of a bee colony, recruitment was important in environments with few flower patches (Dornhaus et al., 2006).

A similar scenario has been explored by the HoFoReSim model (Schmickl et al., 2012), where simulated bees foraged from nectar sources, the quality of which changed once during a simulation run. The colony consisted of foragers that brought nectar into the nest and receivers that processed it. If a forager could not find a receiver to unload the nectar to, i.e., when the nest's nectar intake rate was too high for the receivers to handle, the forager did not waggle dance, but instead performed a tremble dance in order to activate additional dormant receivers. The authors argued that the ratio of active receivers to foragers played an important role in the ability of the colony to switch between different sources of nectar when quality changes occurred. Foragers were able to switch to a better source of nectar more quickly when there were relatively few receivers. In this situation, successful foraging overloaded receivers, causing foragers to spend time activating more receivers rather than recruiting more foragers to their foraging sites. This reduction in forager recruitment allowed bees to be less committed to high quality foraging sites, making responses to a change in source quality faster. The role played by scarce receivers was therefore one of information regulation. In our simulation, regulation of information spread is achieved differently, e.g., by shorter communication range or constraints imposed on movement of the robots, but it has the same effect, allowing the swarms to respond to changes in the environment more appropriately. Under this reading, information flow considerations help explain a swarm's ability to exhibit plasticity both in our simulation and in HoFoReSim.

9.3. Swarm-level plasticity

In order to analyse foraging swarms, we quantified the value of information transfer and identified the following four work modes that a swarm could reach: switching, delayed switching, locked and indecisive. Swarms in the switching mode could respond to changes in deposit qualities well and usually performed better than swarms in other work modes (for example, see Figure 8c,d, Figure 9c,d and Figure 9a for Heap4). Delayed switching, locked and indecisive modes were associated with worse levels of performance, although the extent of their disadvantage varied based on the environment and the swarm size.

Figure 18 shows the work modes exhibited by swarms using the different behavioural strategies explored here, and the conditions under which they changed from one mode to another. The strategies are ordered from top to bottom by their ability to spread information, i.e., by the size of the positive information value region following the first deposit quality change. We tested the swarms under normal conditions, explored in Section 6, and under three kinds of perturbation (restricted dance floor shape, pellet accumulation and increase swarm size), explored in Section 8, giving us four foraging conditions in total. In addition, we explored four different scenario types: Heap2 vs. Heap4, and $T_Q = 1$ h vs. $T_Q = 2$ h. We could thus compare the behavioural strategies across a total of $4 \times 4 = 16$ different experimental setups (Figure 11).

Short-range checkers were in the switching mode in all four scenario types under normal foraging conditions and exhibited the indecisive mode only when material accumulation increased in the Heap4 scenarios with $T_Q = 1$ h and $T_Q = 2$ h. We can say that the swarms were in the switching mode in 14 out of 16 (87.5%) of cases. Long-range checkers also found themselves in the switching mode in all scenarios under normal conditions, but there were three conditions under which they became either indecisive or delayed. These swarms were in the switching mode in 13 out of 16 (81.25%) cases. Moving along the information speed axis, short-range beggers were in the switching mode in only 5 out of 16 (31.25%) cases, and long-range beggers in only 2 out of 16 (12.5%) cases.

It is clear that rapid information spread often leads to the delayed, locked or indecisive modes, i.e., that it prevents plastic self-organisation. However, it is important to point out that foraging performance in static environments was higher when informa-

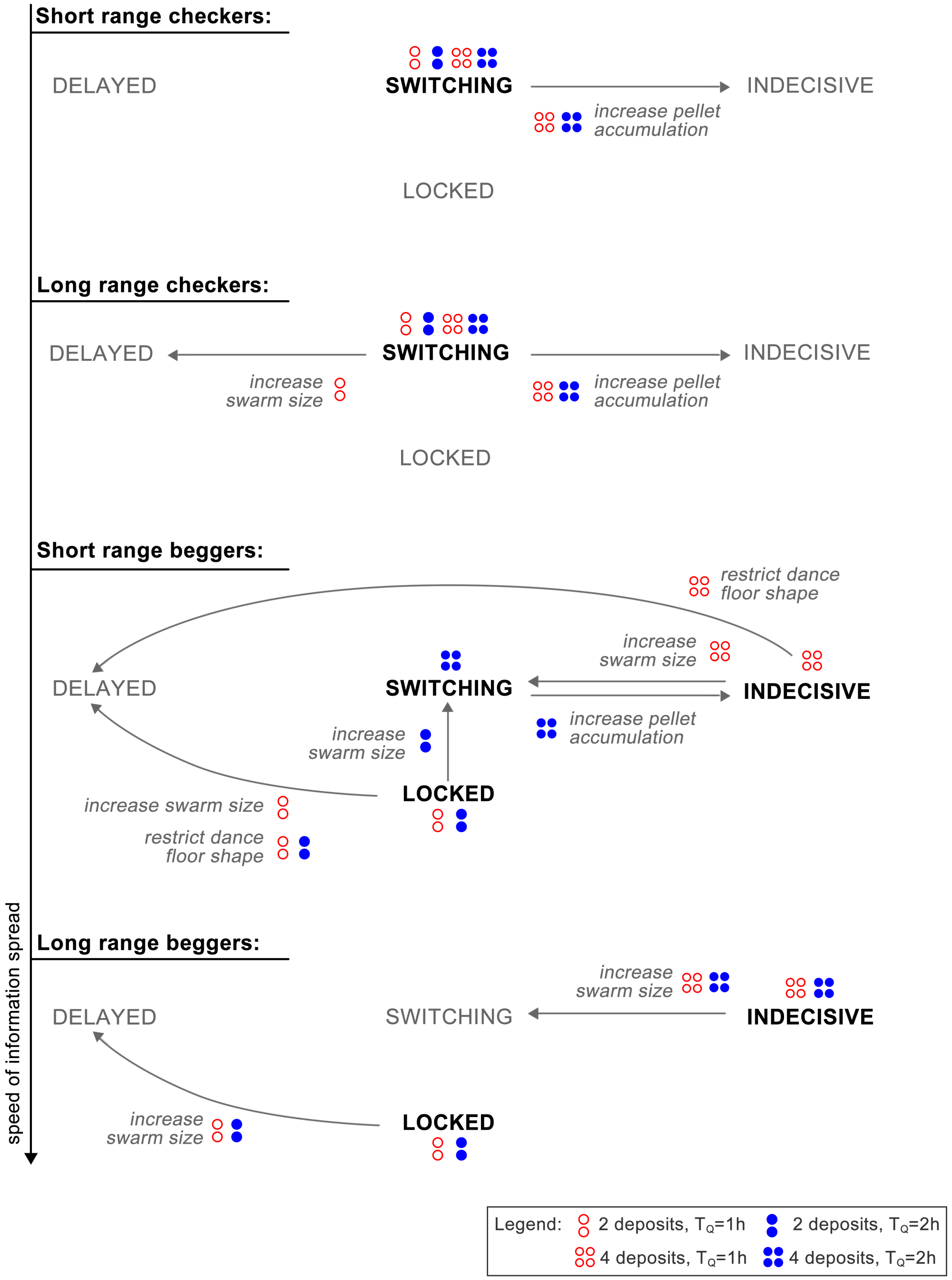


Figure 18: Work modes and mode transitions exhibited by swarms with different behavioural strategies. The default modes that swarms operated in under normal conditions i.e., full dance floor, no pellet accumulation, 25 robots - see Sections 5 and 6) are shown in bold. Scenario markers related to a combination of the number of deposits and the length of deposit quality change interval, T_Q, are described in the figure legend. The markers are placed above default work modes and the transition arrows to indicate the scenarios in which they occurred.

tion spread was fast. Furthermore, while the use of long-range recruitment in dynamic environments decreased the number of experimental setups where swarms were in the switching mode, it also led to a greater amount of resource being collected when the switching mode did occur and when congestion was not too high due to large swarm size or material accumulation. We could thus say that while using a robot behavioural strategy where information spreads quickly might be desirable in some application scenarios, designers of swarms have to be more certain about foraging conditions like swarm size, material accumulation, number of deposits, or the length of the deposit quality change interval. On the other hand, while they were never the best in terms of foraging performance, swarms of short-range checkers, where information spread slowly, exhibited plasticity more commonly, as they retained their ability to switch between deposits under most tested conditions. Such a strategy would thus be more suitable in unknown or uncontrollable environments.

These findings on the effects of information flow could be applied to a wider range of problems. In particular, the following swarm robotic tasks are the most relevant:

- A number of researchers are interested in collection from different deposit types at the same time (e.g., Balch, 1999; Campo and Dorigo, 2007). Results here show that inhibition of social information spread, which can be achieved by reducing communication range or restricting the opportunities for communication to take place (in our study, this is effected by restricting the shape of the recruitment area or altering the character of pellet accumulation), can lead to the desired outcome of foraging from different deposits in parallel.

- Communication about resource availability was used in robots that could adapt their resting and waking thresholds based on food density (Liu and Winfield, 2010). The speed of information spread, dependent on design decisions about communication type and range, could potentially play an important role in such a task and affect a swarm's flexibility.

- Task allocation has received substantial interest (e.g., Zhang et al., 2007; Sarker and Dahl, 2011; Jevtic et al., 2012; Zahadat et al., 2013). It is somewhat similar to deposit selection, as robots need to collectively explore the space of possible work sites and perform work on them. Principles of information transfer could be applied here, for example by substituting deposit energy efficiency for work site importance or utility during action selection and during transmission of social information. Similarly, information value, introduced in Section 7, could be adapted to relate to work site importance rather than deposit quality. We aim to explore this domain in our future work.

- As an alternative to design by hand, swarm behaviour design can be achieved by reinforcement learning (e.g., Pérez-Uribe, 2001) or artificial evolution (e.g., Doncieux et al., 2015; Ferrante et al., 2015; Francesca et al., 2015). Knowing what types of communication strategies are beneficial under various conditions or at least which ones are more flexible could minimise the parameter search space for these optimisation algorithms and potentially decrease complexity of the resulting behaviours or save some simulation time.

Finally, it would be possible to utilise information value in an adaptive robot control algorithm in order to make a swarm more autonomous. For example, if robots could identify pathological effects of fast or slow information transfer, they could alter their individual behaviour or their communication range in order to achieve a more effective work mode.

9.4. Conclusion

Swarm foraging has many robotic applications and can also be used as a paradigm for other swarm behaviours such as task allocation, labour division or robot dispersion. Social insects like ants and bees give an example of how powerful swarm intelligence can be and how plastic adaptive behaviour can emerge from interactions between relatively limited system parts. However, before we can use robotic swarms in the real world, we need to understand how to design individual agents for reliable collaborative work in dynamic environments.

We argued that the way in which the character of robot-robot communication influences the flow of information through a swarm and the swarm's ability to obtain new information from the environment is an important factor that affects swarm plasticity and, as a consequence, foraging performance. Behavioural strategies that limit the spread of information promote plastic self-organisation and are thus more suitable for unknown or highly dynamic environments. We have also demonstrated a method for measuring the value of information that can help us identify such plastic behavioural strategies, not only in foraging swarms but also in a wider range of swarm behaviours where social information is utilised.

Acknowledgements This work was supported by an EPSRC Doctoral Training Centre grant (EP/G03690X/1). All source code and data supporting this study are openly available from the University of Southampton repository at http://dx.doi.org/10.5258/SOTON/385724

References

Alers, S., Bloembergen, D., Hennes, D., Jong, S. D., Kaisers, M., Lemmens, N., Tuyls, K., and Weiss, G. (2011). Bee-inspired foraging in an embodied swarm. In Tumer, K., Yolum, P., Sonenberg, L., and Stone, P., editors, *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, pages 1311–1312, New York. ACM.

Arab, A., Carollo Blanco, Y., and Costa-Leonardo, A. M. (2012). Dynamics of foraging and recruitment behavior in the asian subterranean termite coptotermes gestroi (*Rhinotermitidae*). *Psyche: A Journal of Entomology*, 2012:Article ID 806 782.

Arkin, R. C. (1992). Cooperation without communication: Multiagent schema-based robot navigation. *Journal of Robotic Systems*, 9(3):351–364.

Balch, T. (1999). The impact of diversity on performance in multi-robot foraging. In Etzioni, O., Müller, J. P., and Bradshaw, J. M., editors, *Proceedings of the Third Annual Conference on Autonomous Agents*, pages 92–99, New York. ACM.

Balch, T. and Arkin, R. C. (1994). Communication in reactive multiagent robotic systems. *Autonomous Robots*, 1(1):27–52.

Biesmeijer, J. C. and De Vries, H. (2001). Exploration and exploitation of food sources by social insect colonies: A revision of the scout-recruit concept. *Behavioral Ecology and Sociobiology*, 49(2):89–99.

Bonani, M., Longchamp, V., Magnenat S., Philippe, R., 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 *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)*, pages 4187 – 4193, Piscataway, NJ. IEEE Press.

Borenstein, J. (1998). Experimental results from internal odometry error correction with the OmniMate mobile robot. *IEEE Transactions on Robotics and Automation*, 14(6):963–969.

Campo, A. and Dorigo, M. (2007). Efficient multi-foraging in swarm robotics. In Almeida e Costa, F., editor, *Proceedings of the 9th European Conference on Advances in Artificial Life (ECAL 2007)*, pages 696–705, Berlin. Springer.

Campo, A., Gutiérrez, Á., Nouyan, S., Pinciroli, C., Longchamp, V., Garnier, S., and Dorigo, M. (2010). Artificial pheromone for path selection by a foraging swarm of robots. *Biological Cybernetics*, 103(5):339–352.

Dai, H. (2009). Adaptive control in swarm robotic systems. *The Hilltop Review*, 3(1):54–67.

De Marco, R. and Farina, W. M. (2001). Changes in food source profitability affect the trophallactic and dance behavior of forager honeybees (*Apis mellifera L.*). *Behavioral Ecology and Sociobiology*, 50(5):441–449.

De Marco, R. and Farina, W. M. (2003). Trophallaxis in forager honeybees *Apis mellifera*: Resource uncertainty enhances begging contacts? *Journal of Comparative Physiology A*, 189:125–134.

Donaldson-Matasci, M. C. and Dornhaus, A. (2012). How habitat affects the benefits of communication in collectively foraging honey bees. *Behavioral Ecology and Sociobiology*, 66(4):583–592.

Doncieux, S., Bredeche, N., Mouret, J.-B., and Eiben, A. E. (2015). Evolutionary robotics: What, why, and where to. *Frontiers in Robotics and AI*, 2(4).

Dornhaus, A., Klugl, F., Oechslein, C., Puppe, F., and Chittka, L. (2006). Benefits of recruitment in honey bees: Effects of ecology and colony size in an individual-based model. *Behavioral Ecology*, 17(3):336–344.

Drogoul, A. and Ferber, J. (1993). From Tom Thumb to the Dockers: Some experiments with foraging robots. In Meyer, J., Roitblat, H. L., and Wilson, S. W., editors, *From Animals to Animats II*, pages 451–459. MIT Press, Cambridge, MA.

Ducatelle, F., Di Caro, G. A., Pinciroli, C., and Gambardella, L. M. (2011). Self-organized cooperation between robotic swarms. *Swarm Intelligence*, 5(2):73–96.

Farina, W. M., Grüter, C., and Díaz, P. C. (2005). Social learning of floral odours inside the honeybee hive. *Philosophical Transactions of the Royal Society of London, Series B*, 272:1923–1928.

Ferrante, E. and Duéñez Guzmán, E. (2013). GESwarm: Grammatical evolution for the automatic synthesis of collective behaviors in swarm robotics. In Blum, C., editor, *Proceedings of the Fifteenth International Conference on Genetic and Evolutionary Computation Conference Companion (GECCO 2013)*, pages 17–24. ACM.

Ferrante, E., Turgut, A. E., Dué, E., and Dorigo, M. (2015). Evolution of self-organized task specialization in robot swarms. *PLoSComputational Biology*, 11(8):e1004273.

Francesca, G., Brambilla, M., Brutschy, A., Garattoni, L., Miletitch, R., Podevijn, G., Reina, A., Soleymani, T., Salvaro, M., Pinciroli, C., Mascia, F., Trianni, V., and Birattari, M. (2015). AutoMoDe-chocolate: automatic design of control software for robot swarms. *Swarm Intelligence*, 9(2–3):125–152.

Fujisawa, R., Dobata, S., Sugawara, K., and Matsuno, F. (2014). Designing pheromone communication in swarm robotics: Group foraging behavior mediated by chemical substance. *Swarm Intelligence*, 8(3):227–246.

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 *Proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007)*, pages 37–44, Piscataway, NJ. IEEE.

Granovskiy, B., Latty, T., Duncan, M., Sumpter, D. J. T., and Beekman, M. (2012). How dancing honey bees keep track of changes: The role of inspector bees. *Behavioral Ecology*, 23(3):588–596.

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

Hoff, N., Wood, R., and Nagpal, R. (2013). Distributed colony-level algorithm switching for robot swarm foraging. In Martinoli, A., Mondada, F., Correll, N., Mermoud, G., Hsieh, M. A., Parker, L. E., and Stoy, K., editors, *Distributed Autonomous Robotic Systems*, volume 83 of *Springer Tracts in Advanced Robotics*, pages 417–430. Springer, Berlin.

Jevtic, A., Gutiérrez, A., Andina, D., and Jamshidi, M. (2012). Distributed bees algorithm for task allocation in swarm of robots. *IEEE Systems Journal*, 6(2):296–304.

Jones, C. and Mataric, M. J. (2003). Adaptive division of labor in large-scale minimalist multi-robot systems. In *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, pages 1969 – 1974, vol. 2, Piscataway, NJ. IEEE Press.

Kazama, T., Sugawara, K., and Watanabe, T. (2005). Traffic-like movement on a trail of interacting robots with virtual pheromone. In Murase, K., Sekiyama, K., Naniwa, T., Kubota, N., and Sitte, J., editors, *Proceedings of the 3rd International Symposium on Autonomous Minirobots for Research and Edutainment (AMiRE 2005)*, pages 383–388, Berlin. Springer.

Krieger, M. J. B. and Billeter, J.-B. (2000). The call of duty: Self-organised task allocation in a population of up to twelve mobile robots. *Robotics and Autonomous Systems*, 30(1-2):65–84.

Labella, T. H., Dorigo, M., and Deneubourg, J.-L. (2004). Efficiency and task allocation in prey retrieval. In Ijspeert, A. J., Murata, M., and Wakamiya, N., editors, *Biologically Inspired Approaches to Advanced Information Technology*, volume 3141 of *Lecture Notes in Computer Science*, pages 274–289. Springer, Berlin.

Lee, J. H. and Ahn, C. W. (2011). Improving energy efficiency in cooperative foraging swarm robots using behavioral model. In Abdullah, R., Khader, A. T., Venkat, I., Wong, L.-P., and Subramanian, K. G., editors, *Proceedings of the Sixth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA'11)*, pages 39–44, Piscataway, NJ. IEEE Press.

Lein, A. and Vaughan, R. T. (2009). Adapting to non-uniform resource distributions in robotic swarm foraging through work-site relocation. In *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2009)*, pages 601–606, Piscataway, NJ. IEEE Press.

Lemmens, N., de Jong, S., Tuyls, K., and Nowe, A. (2008). Bee behaviour in multi-agent systems. In Tuyls, K., Nowe, A., Guessoum, Z., and Kudenko, D., editors, *Adaptive Agents and Multi-Agent Systems III. Adaptation and Multi-Agent Learning*, volume 4865 of *Lecture Notes in Computer Science*, pages 145–156. Springer, Berlin.

Lerman, K., Jones, C., Galstyan, A., and Mataric, M. J. (2006). Analysis of dynamic task allocation in multi-robot systems. *The International Journal of Robotic Research*, 25:225–242.

Liu, W. and Winfield, A. F. (2010). Modelling and optimisation of adaptive foraging in swarm robotic systems. *The International Journal of Robotics Research*, 29(14):1743–1760.

Liu, W., Winfield, A. F., Sa, J., Chen, J., and Dou, L. (2007). Strategies for energy optimisation in a swarm of foraging robots. In Sahin, E., Spears, W. M., and Winfield, A. F., editors, *Swarm Robotics*, volume 4433 of *Lecture Notes in Computer Science*, pages 14–26. Spinger, Berlin.

Martinoli, A., Easton, K., and Agassounon, W. (2004). Modeling swarm robotic systems: A case study in collaborative distributed manipulation. *The International Journal of Robotics Research*, 23(4):415–436.

Mayet, R., Roberz, J., Schmickl, T., and Crailsheim, K. (2010). Antbots: A feasible visual emulation of pheromone trails for swarm robots. In Dorigo, M., Birattari, M., Di Caro, G. A., Doursat, R., Engelbrecht, A. P., Floreano, D., Gambardella, L. M., Groß, R., Sahin, E., Sayama, H., and Stützle, T., editors, *Swarm Intelligence*, volume 6234 of *Lecture Notes in Computer Science*, pages 84–94. Springer, Berlin.

McFarland, D. and Spier, E. (1997). Basic cycles, utility and opportunism in self-sufficient robots. *Robotics and Autonomous Systems*, 20:179–190.

Miletitch, R., Trianni, V., Campo, A., and Dorigo, M. (2013). Information aggregation mechanisms in social odometry. In Liò, P., Miglino, O., Nicosia, G., Nolfi, S., and Pavone, M., editors, *Proceedings of the Twelfth European Conference on the Synthesis and Simulation of Living Systems (ECAL 2013)*, pages 102–109, Cambridge, MA. MIT Press.

Miller, J. M., Wang, X. R., Lizier, J. T., Prokopenko, M., and Rossi, L. F. (2014). Measuring information dynamics in swarms. In Prokopenko, M., editor, *Guided Self-Organisation: Inception*, volume 9 of *Emergence, Complexity and Computation*, pages 343–364. Springer, Berlin.

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.

Pais, D., Caicedo-Núnez, C. H., and Leonard, N. E. (2012). Hopf bifurcations and limit cycles in evolutionary network dynamics. *SIAM Journal on Applied Dynamical Systems*, 11(4):1754–1784.

Parunak, H. and Brueckner, S. A. (2004). Engineering swarmings systems. In Bergenti, F., Gleizes, M.-P., and Zambonelli, F., editors, *Methodologies and Software Engineering for Agent Systems*, volume 11 of *Multiagent Systems, Artificial Societies, and Simulated Organizations*, pages 341–376. Springer, Berlin.

Pérez-Uribe, A. (2001). Using a time-delay actor-critic neural architecture with dopamine-like reinforcement signal for learning in autonomous robots. In Wermter, S., Austin, J., and Willshaw, D., editors, *Emergent Neural Computational Architectures Based on Neuroscience*, volume 2036 of *Lecture Notes in Computer Science*, pages 522–533. Springer, Berlin.

Pinciroli, C., Trianni, V., O'Grady, R., Pini, G., Brutschy, A., Brambilla, M., Mathews, N., Ferrante, E., 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.

Pini, G., Brutschy, A., Pinciroli, C., Dorigo, M., and Birattari, M. (2013). Autonomous task partitioning in robot foraging: An approach based on cost estimation. *Adaptive Behavior*, 21(2):118–136.

Pitonakova, L., Crowder, R., and Bullock, S. (2014). Understanding the role of recruitment in collective robot foraging. In Lipson, H., Sayama, H., Rieffel, J., Risi, S., and Doursat, R., editors, *Proceedings of the Fourteenth International Conference on the Synthesis and Simulation of Living Systems (ALIFE 14)*, pages 264–271, Cambridge, MA. MIT Press.

Ranjbar-Sahraei, B., Weiss, G., and Nakisaei, A. (2012). A multi-robot coverage approach based on stigmergic communication. In Timm, I. J. and Guttmann, C., editors, *Multiagent System Technologies*, volume 7598 of *Lecture Notes in Computer Science*, pages 126–138. Springer, Berlin.

Reynolds, A. M. and Rhodes, C. J. (2009). The Lévy flight paradigm: Random search patterns and mechanisms. *Ecology*, 90(4):877–887.

Ribeiro, P. L., Helene, A. F., Xavier, G., Navas, C., and Ribeiro, F. L. (2009). Ants can learn to forage on one-way trails. *PloS one*, 4(4):e5024.

Russell, R. A. (1999). Ant trails - an example for robots to follow? In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, pages 2698 – 2703 vol. 4, Piscataway, NJ. IEEE Press.

Sarker, M. O. F. and Dahl, T. S. (2011). Bio-Inspired communication for self-regulated multi-robot systems. In Yasuda, T., editor, *Multi-Robot Systems, Trends and Development*, pages 367–392. InTech.

Schmickl, T. and Crailsheim, K. (2008). Throphallaxis within a robotic swarm: Bio-inspired communication among robots in a swarm. *Autonomous Robots*, 25(1):171–188.

Schmickl, T. and Hamann, H. (2010). BEECLUST: A swarm algorithm derived from honeybees. In Xiao, Y., editor, *Bio-inspired Computing and Networking*, pages 95–137. Routledge, Abingdon.

Schmickl, T., Möslinger, C., and Crailsheim, K. (2007). Collective perception in a robot swarm. In Sahin, E., Spears, W. M., and Winfield, A. F., editors, *Swarm Robotics*, volume 4433 of *Lecture Notes in Computer Science*, pages 144–157. Springer, Berlin.

Schmickl, T., Thenius, R., and Crailsheim, K. (2012). Swarm-intelligent foraging in honeybees: Benefits and costs of task-partitioning and environmental fluctuations. *Neural Computing and Applications*, 21(2):251–268.

Seeley, T. D. (1994). Honey bee foragers as sensory units of their colonies. *Behavioral Ecology and Sociobiology*, 34(1):51–62.

Seeley, T. D., Camazine, S., and Sneyd, J. (1991). Collective decision-making in honey bees: How colonles choose among nectar sources. *Behavioral Ecology and Sociobiology*, 28:277–290.

Seeley, T. D. and Morse, R. a. (1976). The nest of the honey bee (*Apis mellifera L.*). *Insectes Sociaux*, 23:495–512.

Seeley, T. D., Visscher, P. K., Schlegel, T., Hogan, P. M., Franks, N. R., and Marshall, J. a. R. (2012). Stop signals provide cross inhibition in collective decision-making by honeybee swarms. *Science*, 335(6064):108–111.

Serugendo, G. M., Gleizes, M.-P., and Karageorgos, A. (2006). Self-organisation and emergence in MAS: An overview self-organisation. *Informatica*, 30:45–54.

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

Sumpter, D. J. T. and Beekman, M. (2003). From nonlinearity to optimality: Pheromone trail foraging by ants. *Animal Behaviour*, 66(2):273–280.

Tereshko, V. and Loengarov, A. (2005). Collective decision-making in honey bee foraging dynamics. *Computing and Information Systems Journal*, 9(3):1–7.

Thenius, R., Schmickl, T., and Crailsheim, K. (2008). Optimisation of a honeybee-colony's energetics via social learning based on queuing delays. *Connection Science*, 20(2-3):193–210.

Ulam, P. and Balch, T. (2004). Using optimal foraging models to evaluate learned robotic foraging behavior. *Adaptive Behavior*, 12(4):213–222.

Valdastri, P., Corradi, P., Menciassi, A., Schmickl, T., Crailsheim, K., Seyfried, J., and Dario, P. (2006). Micromanipulation, communication and swarm intelligence issues in a swarm microrobotic platform. *Robotics and Autonomous Systems*, 54(10):789–804.

Valentini, G., Hamann, H., and Dorigo, M. (2014). Self-organized collective decision making: The weighted voter model. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014)*, pages 45–52, New York. ACM.

von Frisch, K. (1967). *The dance language and orientation of bees.* Harvard University Press, Cambridge, MA.

Winfield, A. F. (2009). Towards an engineering science of robot foraging. In Asama, H., Kurokawa, H., and Sekiyama, K., editors, *Distributed Autonomous Robotic Systems 8*, pages 185–192. Springer, Berlin.

Zahadat, P., Crailsheim, K., and Schmickl, T. (2013). Social inhibition manages division of labour in artificial swarm systems. In Liò, P., Miglino, O., Nicosia, G., Nolfi, S., and Pavone, M., editors, *Proceedings of the Twelfth European Conference on the Synthesis and Simulation of Living Systems (ECAL 2013)*, pages 609–616, Cambridge, MA. MIT Press.

Zhang, D., Xie, G., Yu, J., and Wang, L. (2007). Adaptive task assignment for multiple mobile robots via swarm intelligence approach. *Robotics and Autonomous Systems*, 55(7):572–588.