Chapter Twelve

Collective Foraging: Cleaning, Energy Harvesting and Trophallaxis

Alan F. T. Winfield, Serge Kernbach, Thomas Schmickl

12.1 Introduction

Foraging is a powerful benchmark problem in collective robotics for several reasons. Firstly, sophisticated foraging observed in social insects, recently becoming well understood, provides both inspiration and system level models for artificial systems. Secondly, collective foraging is a complex task involving the coordination of several - each also difficult - tasks including efficient exploration (searching) for objects, food or prey; physical collection (harvesting) of objects almost certainly requiring physical manipulation; homing or navigation whilst transporting those objects to collection point(s), and deposition of the objects before returning to foraging. And thirdly, effective multi-robot foraging requires cooperation between individuals involving either communication to signal to others where objects may be found (e.g. pheromone trails, or direction giving) and/or cooperative transport of objects too large for a single individual to transport.

There are, at the time of writing, no known examples of collective foraging robots successfully employed in real-world applications. Most foraging robots are to be found in research laboratories or, if they are aimed at real-world applications, are at the stage of prototype or proof-of-concept. The reason for this is that foraging is a complex task which requires a range of competencies to be tightly integrated within the physical robots and, although the principles of robot foraging are now becoming established, many of the sub-system technologies required for foraging robots remain very challenging. In particular, sensing and situational awareness; power and energy autonomy; actuation, locomotion and safe navigation in unknown physical environments and proof of safety and dependability all remain difficult problems in collective robotics.

This chapter first defines collective foraging with reference to an abstract

model, presented in Sect. 12.2, then uses the terminology of the abstract model to review and introduce strategies for cooperation in collective foraging (Sect. 12.3), including information sharing, physical cooperation and division of labour. The rest of the chapter then describes a series of three case studies in collective foraging, and Fig. 12.1. shows how these case studies and their contributions are linked. Energy foraging is the special case of collective foraging in which robots are foraging for their own energy, and our first case study in Sect. 12.4 describes examples in which robots are required to find and harvest their own energy from the environment, with division of labour. Trophallaxis refers to an energy (food) exchange between adult social insects, or adults and their larvae, and case study two in Sect. 12.5 introduces an information-sharing trophallaxis approach to collective cleaning (i.e. foraging for dirt). The same trophallaxis-inspired approach is enhanced with individual adaptation in Sect. 12.5.6 and the case study concludes with a thought experiment which would extend the approach to true energy trophallaxis. Sect. 12.6 then describes our third case study – a kinetic model of foraging - in which a robot collective must maintain energy homeostasis by foraging for energy from fixed charging stations.



Figure 12.1. The relationship and thematic links between the three case studies in collective foraging of this chapter

12.2. An Abstract model of Collective Foraging 5

12.2 An Abstract model of Collective Foraging

Foraging robots are mobile robots capable of searching for and, when found, transporting objects to one or more collection points.



Figure 12.2. Finite State Machine for Basic Collective Foraging

Fig. 12.2. shows a Finite State Machine (FSM) representation of each robot in a foraging collective. In the model each robot is in always in one of five states: *searching, grabbing, homing, depositing* or *avoiding*. Implied in this model is, firstly, that the environment or search space contains more than one of the target objects; secondly, that there is a single collection point (hence this model is sometimes referred to as central-place foraging), and thirdly, that the process continues indefinitely. Clearly not all robots will be in the same state at the same time and the FSM of Fig. 12.2. can be viewed as a representation of the average number of robots in each state, in the collective. The five states are defined as follows.

- (1) Searching. In this state a robot is physically moving through the search space using its sensors to locate and recognise the target items. At this level of abstraction we do not need to state how the robot searches: it could, for instance, wander at random, or it could employ a systematic strategy such as moving alternately left and right in a search pattern. The fact that the robot has to search at all follows from the pragmatic real-world assumptions that either the robot's sensors are of short range and/or the items are hidden (behind occluding obstacles for instance); in either event we must assume that the robot cannot find items simply by staying in one place and scanning the whole environment with its sensors. Object identification or recognition could require one of a wide range of sensors and techniques. When the robot finds an item it changes state from searching to grabbing. If the robot fails to find the target item then it remains in the searching state forever; searching is therefore the 'default' state.
- (2) Grabbing. In this state a robot physically captures and grabs the item ready to

transport it back to the home region. Here we assume that the item is capable of being grabbed and conveyed by a single robot. As soon as the item has been grabbed the robot will change state to homing. Clearly grabbing may fail if there is competition between robots to grab the same object, in which case a robot will resume searching.

- (3) Homing. In this state a robot must move, with its collected object, to a home or nest region. Homing clearly requires a number of stages, firstly, determination of the position of the home region relative to where the robot is now, secondly, orientation toward that position and, thirdly, navigation to the home region. Again there are a number of strategies for homing: one would be to re-trace the robot's path back to the home region using, for instance, odometry or by following a marker trail; another would be to home in on a beacon with a long range beacon sensor. When the robot has successfully reached the home region it will change state to depositing.
- (4) Depositing. In this state the robot deposits or delivers the item in the home region, and then immediately changes state to searching and hence resumes its search.
- (5) Avoiding. Clearly in a collective robot system robots are likely to collide with each other, especially if two or more robots converge on the same object. Thus, if a robot's collision avoidance sensors are triggered when the robot is in any of the four states above, then it must take avoiding action before returning to that state.

There are clearly numerous variations on this basic foraging model. Some are simplifications: for instance if a robot is searching for one or a known fixed number of objects then the process will not loop indefinitely. Real robots do not have infinite energy and so a model of practical foraging would need to take account of energy management. However, many variations entail either complexity within one or more of the four basic states (consider, for instance, objects that actively evade capture - a predator-prey model of foraging), or complexity in the interaction or cooperation between robots. Thus the basic model stands as a powerful top-level abstraction and a useful basis for extension to more complex foraging systems. For a full description of single and multi-robot foraging, including a taxonomy of robot foraging, refer to [Winfield (2009)].

12.3 Strategies for Cooperation in Collective Foraging

Foraging is clearly a task that lends itself to multi-robot systems and, even if the task can be accomplished by a single robot, foraging should — with careful design of strategies for cooperation — benefit from multiple robots. Swarm intelligence is the study of natural and artificial systems of multiple agents in which there is no centralised or hierarchical command or control. Instead, global swarm behaviours emerge as a result of local interactions between the agents and each other, and between agents and the environment, [Bonabeau *et al.* (1999a)]. Swarm robotics is concerned with the design of artificial robot swarms based upon the principles

12.3. Strategies for Cooperation in Collective Foraging 7

of swarm intelligence, thus control is completely distributed and robots, typically, must choose actions on the basis only of local sensing and communications, [Beni (2005); Sahin (2005)].

Foraging is therefore a benchmark problem within swarm robotics, not least because of the strong cross-over between the study of self-organisation in social insects and their artificial counterparts within swarm intelligence [Dorigo and Birattari (2007)]. This section will therefore focus on examples of collective foraging from within the field of swarm robotics. Three strategies for cooperation will be outlined: information sharing, physical cooperation and division of labour.

12.3.1 Information sharing



Figure 12.3. Finite State Machine for multi-robot foraging with recruitment - adapted from [Matarić and Marjanovic (1993)]

Matarić and Marjanovic provide what is believed to be the first description of a multi-robot foraging experiment using real (laboratory) robots in which there is no centralised control [Matarić and Marjanovic (1993)]. They describe a system of 20 identical 12" 4-wheeled robots, equipped with: a two-pronged forklift for picking up, carrying and stacking metal pucks; proximity and bump sensors; radio transceivers for data communication and a sonar-based global positioning system. Matarić and Marjanovic extend the basic five state foraging model (wandering, grabbing, homing, dropping and avoiding), to introduce information sharing as follows. If a robot finds a puck it will grab it but also broadcast a radio message to tell other robots it has found a puck. Meanwhile, if another robot in the locale hears this message it will first enter state *tracking* to home in on the source of the message, then state searching - a more localised form of wandering. The robot will return to wandering if it finds no puck within some time out period. Furthermore, while in state *tracking* a robot will also transmit a radio signal. If nearby robots hear this signal they will switch from wandering into *following* to pursue the tracking robot. Thus the tracking robot actively recruits additional robots as it seeks the original successful robot (a form of secondary swarming, [Melhuish (1999)]); when the tracking robot switches to searching its recruits will do the same. Figure 12.3. shows a simplified FSM.

12.3.2 Physical cooperation



Figure 12.4. Cooperative grabbing: Khephera robots engaged in collective stick-pulling. Would need to get permission of A. Martinoli.

cooperative grabbing Consider the case of multi-robot foraging in which the object to be collected cannot be grabbed by a single robot working alone. Ijspeert *et al* describe an experiment in collaborative stick-pulling in which two robots must work together to pull a stick out of a hole [Ijspeert *et al.* (2001); Martinoli *et al.* (2004)]. Each *Khephera* robot is equipped with a gripper capable of grabbing and lifting the stick, but the hole containing the stick is too deep for one robot to be able to pull the stick out alone; one robot must pull the stick half-way then wait for another robot to grab the stick and lift it clear of the hole, see figure 12.4.. Ijspeert and co-workers describe an elegant minimalist strategy which requires no direct communication between robots. If one robot finds a stick it will lift it and wait. If another finds the same stick it will also lift it, on sensing the force on the stick from the second robot the first robot will let go, hence allowing the second to complete the operation.

cooperative transport Now consider the the situation in which the object to be collected is too large to be transported by a single robot. Parker describes the ALLIANCE group control architecture applied to an example of cooperative boxpushing by two robots [Parker (1994)].

Arguably the most accomplished demonstration of cooperative multi-robot foraging to date is within the *swarm-bot* project of Dorigo and co-workers [Dorigo *et al.* (2005)] (also described in Chapter X of this handbook). The *s-bot* is a modular robot equipped with both a gripper and a gripping ring, which allows one robot to grip another [Mondada *et al.* (2005)]. Importantly, the robot is able to rotate its wheelbase independently of the gripping ring so that robots can grip each other at any arbitrary point on the circumference of the grip ring but then rotate and align their wheels in order to be able to move as a single unit (a *swarm-bot*). Groß *et al* describe cooperative transport which uses visual signalling [Groß*et al.* (2006)]. *s*-bots are attracted to the (large) object to be collected by its ring of red LEDs. The s-bot's LEDs are blue, but when an s-bot finds and grabs the attractor object it switches its LEDs to red. This increases the red light intensity to attract further s-bots which may grab either the object, or arbitrarily a robot already holding the object. The s-bots are then able to align and collectively move the object. 12.3. Strategies for Cooperation in Collective Foraging 9



Figure 12.5. Cooperative transport by s-bots. Here s-bots are attempting to transport two objects, each too large to be moved by a single s-bot. Image included with kind permission of M. Dorigo.

12.3.3 Division of labour

In multi-robot foraging it is well know that overall performance (measured, for instance, as the number of objects foraged per robot in a given time interval), does not increase monotonically with increasing team size because of interference between robots (overcrowding), [Balch and Arkin (1994); Goldberg and Matarić (1997); Lerman and Galstyan (2002)]. Division of labour in ant colonies has been well studied and in particular a response threshold model is described in [Bonabeau *et al.* (1996)] and [Bonabeau *et al.* (1998)]; in essence a threshold model means that an individual will engage in a task when the level of some task-associated stimulus exceeds its threshold.

For threshold-based multi-robot foraging with division of labour figure 12.6. shows a generalised finite state machine for each robot. In this foraging model the robot will not search endlessly. If the robot fails to find a food-item because, for instance, its searching time exceeds a maximum search time threshold T_s , or its energy level falls below a minimum energy threshold, then it will abandon its search and return home without food, shown as *failure*. Conversely *success* means food was found, grabbed and deposited. Note, however, that a robot might see a food-item but fail to grab it because, for instance, of competition with another robot for the same food-item. The robot now also has a *resting* state during which time it remains in the nest conserving energy. The robot will stop resting and leave home which might be according to some threshold *T*_r, or the overall nest energy falling below a given threshold.

12.3.4 Mathematical modelling of collective foraging

A multi-robot system of foraging robots is typically a stochastic non-linear dynamical system and therefore challenging to mathematically model, but without such models any claims about the correctness of foraging algorithms are weak. Experiments in computer simulation or with real-robots (which provide in effect an



Figure 12.6. Finite State Machine for Foraging with Division of Labour, adapted from [Liu et al. (2007)]

'embodied' simulation) allow limited exploration of the parameter space and can at best only provide weak inductive proof of correctness. Mathematical models, on the other hand, allow analysis of the whole parameter space and discovery of optimal parameters. In real-world applications, validation of a foraging robot system for safety and dependability will require a range of formal approaches including mathematical modelling.

Lerman, Martinoli and co-workers have developed the *macroscopic* approach to directly describe the collective behaviour of the robotic swarm. A class of macroscopic models have been used to study the effect of interference in a swarm of foraging robots [Lerman and Galstyan (2002)] and collaborative stick-pulling [Martinoli *et al.* (2004)]. Lerman *et al* [Lerman *et al.* (2006)] successfully expanded the macroscopic probabilistic model to study dynamic task allocation in a group of robots engaged in a puck collecting task. More recently Liu *et al* [Liu *et al.* (2009)] have applied the macroscopic approach to develop a mathematical model for adaptive foraging with division of labour (the algorithm described below in Sect. 12.4); [Liu and Winfield (2010)] shows how the mathematical model of adaptive foraging can be used in combination with a real-coded genetic algorithm to optimise parameters within the foraging algorithm.

12.4 Case study 1: Collective foraging for Energy

Let us consider the special case of multi-robot foraging in which robots are foraging for their own energy. For an individual robot foraging costs energy, whereas resting conserves energy. We can formally express this as follows. Each robot consumes energy at *A* units per second while searching or retrieving and *B* units per second while resting, where A > B. Each discrete food item collected by a robot provides *C* units of energy to the swarm. The average food item retrieval time, is a function of the number of foraging robots *x*, and the density of food items in the environment, ρ , thus $t = f(x, \rho)$.

If there are *N* robots in the swarm, E_c is the energy consumed and E_r the energy retrieved, per second, by the swarm then

$$E_c = Ax + B(N - x) \tag{12.1}$$

12.4. Case study 1: Collective foraging for Energy 11

$$E_r = Cx/t = \frac{Cx}{f(x,\rho)}$$
(12.2)

The average energy income to the swarm, per second, is clearly the difference between the energy retrieved and the energy consumed,

$$E = E_r - E_c = (\frac{C}{f(x,\rho)} - (A - B))x - BN$$
(12.3)

Equation 12.3 shows that maximising the energy income to the swarm requires either increasing the number of foragers *x* or decreasing the average retrieval time $f(x, \rho)$. However, if we assume that the density of robots in the foraging area is high enough that interference between robots will occur then, for constant ρ , increasing *x* will increase $f(x, \rho)$. Therefore, for a given food density ρ there must be an optimal number of foragers x^* .

Krieger and Billeter adopt a threshold-based approach to the allocation of robots to either foraging or resting; in their scheme each robot is allocated a fixed but randomly chosen activation threshold [Krieger and Billeter (2000)]. While waiting in the nest each robot listens to a periodic radio broadcast indicating the nest-energy level E; when the nest-energy level falls below the robot's personal activation threshold then it leaves the nest and searches for food. It will continue to search until either its search is successful, or it runs out of energy and returns home; if its search is successful and it finds another food-item the robot will record its position (using odometry). On returning home the robot will radio its energy consumption thus allowing the nest to update its overall net energy. Krieger and Billeter show that team sizes of 3 or 6 robots perform better than 1 robot foraging alone, but larger teams of 9 or 12 robots perform less well. Additionally, they test a recruitment mechanism in which a robot signals to another robot waiting in the nest to follow it to the food source, in tandem. Krieger's approach is, strictly speaking, not fully distributed in that the nest is continuously tracking the average energy income *E*; the nest is — in effect — acting as a central coordinator.

Based upon the work of [Deneubourg *et al.* (1987)] on individual adaptation and division of labour in ants, Labella *et al* describe a fully distributed approach that allows the swarm to self-organise to automatically find the optimal value x^* [Labella *et al.* (2006)]. They propose a simple adaptive mechanism to change the ratio of foragers to resters by adjusting the probability of leaving home based upon successful retrieval of food. With reference to figure 12.6. the mechanism works as follows. Each robot will *leave home*, i.e. change state from resting to searching, with probability P_l . Each time the robot makes the *success* transition from deposit to resting, it increments its P_l value by a constant Δ multiplied by the number of consecutive successes, up to a maximum value P_{max} . Conversely, if the robot's searching time is up, the transition *failure* in figure 12.6., it will decrement its P_l by Δ times the number of consecutive failures, down to minimum P_{min} . Interestingly, trials with laboratory robots show that the same robots self-select as foragers or resters — the algorithm exploits minor mechanical differences that mean that some robots are better suited as foragers.

Recently Liu *et al* have extended this fully distributed approach by introducing two additional adaptation rules [Liu *et al.* (2007)]. As in the case of Labella



12 Collective Foraging: Cleaning, Energy Harvesting and Trophallaxis

Figure 12.7. Foraging environment showing 8 robots labelled A - H. The nest region is the grey circle with radius R_h at the centre. Robot A is shown with its arc of vision in which it can sense food items; robots C, E and F have grabbed food items and are in the process of returning to the nest to deposit these. Food items, shown as small squares, 'grow' in order to maintain uniform density within the annular region between circles with radius R_{inner} and R_{outer} .

et al individual robots use internal cues (successful object retrieval), but Liu adds environmental cues (collisions with team mates while searching), and social cues (team mate success in object retrieval), to dynamically vary the time spent foraging or resting. Furthermore, Liu investigates the performance of a number of different adaptation strategies based on combinations of these three cues. The three cues increment or decrement the searching time and resting time thresholds T_s and T_r as follows (note that adjusting T_r is equivalent to changing the probability of leaving the nest P_l):

- (1) Internal cues. If a robot successfully finds food it will reduce its own rest time T_r ; conversely if the robot fails to find food it will increase its own rest time T_r .
- (2) Environment cues. If a robot collides with another robot while searching, it will reduce its T_s and increase its T_r times.
- (3) Social cues. When a robot returns to the nest it will communicate its food retrieval success or failure to the other robots in the nest. A successful retrieval will cause the other robots in the nest to increase their T_s and reduce their T_r times. Conversely failure will cause the other robots in the nest to reduce their T_s and increase their T_r times.

In order to evaluate the relative effect of these cues three different strategies are tested, against a baseline strategy of no cooperation. The strategy/cue combina-

12.4. Case study 1: Collective foraging for Energy 13

tions are detailed in table 12.1.

	internal cues	social cues	environment cues
S_1 (baseline)	×	×	×
S_2	\checkmark	×	×
S_3	\checkmark	\checkmark	×
S_4	\checkmark	\checkmark	\checkmark



Figure 12.8. Number of foraging robots *x* in a foraging swarm of N = 8 robots with self-organised division of labour. *S*₁ is the baseline (no cooperation strategy); *S*₂, *S*₃ and *S*₄ are three different cooperation strategies (see table 12.1). Food density changes from 0.03 (medium) to 0.015 (poor) at t = 5000, then from 0.015 (poor) to 0.045 (rich) at t = 10000. Each plot is the average of 10 runs. Reprinted from [Liu *et al.* (2007)].

Fig. 12.7. shows the simulated foraging environment used to evaluate the collective foraging algorithm. Figures 12.8. and 12.9., from [Liu et al. (2007)], show the number of active foragers and the instantaneous net swarm energy, respectively, for a swarm of eight robots. In both plots the food density in the environment is changed at time t = 5000 and again at time t = 10000 seconds. Figure 12.8. shows the swarm's ability to automatically adapt the number of active foragers in response to each of the step changes in food density. The baseline strategy S_1 shows of course that all eight robots are actively foraging continuously; $S_2 - S_4$ however require fewer active foragers and strategies with social and environmental cues, S_3 and S_4 , clearly show the best performance. Notice, firstly that the additional of social cues - communication between robots - significantly improves the rate at which the system can adapt the ratio of foragers to resters and, secondly, that the addition of environmental cues — collisions with other robots — brings only a marginal improvement. The rates of change of net swarm energy in figure 12.9. tell a similar story. Interestingly, however, we see very similar gradients for $S_2 - S_4$ when the food density is high (on the RHS of the plot), but when the food density



14 *Collective Foraging: Cleaning, Energy Harvesting and Trophallaxis*

Figure 12.9. Instantaneous net energy *E* of a foraging swarm with self-organised division of labour. S_1 is the baseline (no cooperation strategy); S_2 , S_3 and S_4 are three different cooperation strategies (see table 12.1). Food density changes from 0.03 (medium) to 0.015 (poor) at t = 5000, then from 0.015 (poor) to 0.045 (rich) at t = 10000. Each plot is the average of 10 runs. Reprinted from [Liu *et al.* (2007)].

is medium or poor the rate of increase in net energy of strategies S_3 and S_4 is significantly better than S_2 . This result suggests that foraging robots benefit more from cooperation when food is scarce, than when food is plentiful.

12.5 Case study 2: From Information-sharing Trophallaxis to Energysharing Trophallaxis

In swarm robotics [Beni and Wang (2005); Beni (2005); Sahin (2005); Sahin and Winfield (2008); Dorigo and Şahin (2004)], which is the physical manifestation of swarm intelligent algorithms [Bonabeau et al. (1999b); Kennedy and Eberhart (2001); Millonas (1994)], finding a near-optimal balance between exploration and exploitation is a very important issue. In some sense, all swarm systems - regardless whether they are natural or engineered systems – have to find near-optimal solutions to problems posed by specific internal and environmental conditions. One typical benchmark for the ability of a swarm system to make such decisions dynamically and intelligently is foraging, which means that the swarm has to explore the environment and settle to a collective decision, that is to choose one or more foraging targets from a set of potential targets. After this decision is made, the swarm system has to exploit the chosen foraging targets efficiently and, after these targets are depleted, has to quickly choose new targets which were previously neglected. In addition to this iterative collective decision making, which is usually based on inter-agent communication, navigation is a critical task that has to be solved in such a foraging scenario that can be solved in several ways, leading to different mechanisms of collective foraging: Are the swarm members navigating individually, in groups, as (aligned) swarm/flock or is there

12.5. Case study 2: From Information-sharing Trophallaxis to Energy-sharing Trophallaxis 15

even mass-exploitation by trail formation to foraging sites? All these navigation patterns demand specific solutions concerning individual orientation and communication. Regardless of the chosen foraging pattern, some requirements are ubiquitous: Both, exploration and exploitation, demand for coordination of all swarm members, is easy to achieve in a bodiless abstract computer algorithm, for example in ACO [Dorigo and St/"utzle (2003); Blum (2005)] or PSO [Kennedy and Eberhart (2001)] algorithms. Coordination of agents gets tricky as soon as swarm members are physically embodied agents which are operating in a noisy world with imperfect actuators, sensors and communication devices. Thus robust coordination of swarm members is a crucial issue for robotic swarm performance: Biased sensor noise often affects exploration results. Noise on communication channels limits coordination commands. Centralized and supervised coordination quickly reaches its limits because of scaling issues in swarms of real-world situated embodied agents.

These problem sets pose characteristic challenges for swarm robotics, which can be tackled from several sides: decentralized communication principles, for example near-neighbour communication, can enhance the scalability of the swarm system significantly. Redundancy, of course, might compensate for errors in this nearest neighbour communication. However, a problem of redundant information is the 'validity problem' which plays an important role in such systems: If a single swarm member, who might be misinformed about something, is repeatedly communicating faulty information to other swarm members, this will significantly decrease the swarm's efficiency. Even if communicated information was true at some point in time, changes in the swarm structure or in the environment might have caused this information to be invalid in the meantime. Thus, if redundant and frequent repetition of communication - e.g., information spreading in a wave-like manner throughout the swarm – is used to organize a swarm, it is important that the communicated information has to be checked again (re-evaluated) by several swarm members. This way, wrong or outdated information has to be rejected by the swarm system as soon as possible, otherwise the swarm's performance will be harmed.

In this section, we will describe – as a case study – how a swarm robotic system can be engineered in a bio-inspired way that is able to solve a majority of the above-mentioned problem sets in a bio-inspired way. We will show that such a swarm system can balance exploitation and exploration, is scalable within specific bounds and is able to perform swarm-level cognition tasks like collective perception. At first we describe how a task that is collective foraging for materials is self-organized by a mechanism based on information exchange among nearest neighbour swarm members. Then we show how such a swarm strategy can be enhanced by allowing the members not only to exchange information but also energy, this way opening the door to novel methods of collective energy foraging and swarm-level energy homeostasis.

In natural systems, biological evolution through natural selection has created many swarm systems that are able to achieve the goal of swarm wide communi-



Figure 12.10. Trophallaxis in honeybees: Fluid food is exchanged from one adult bee to another adult bee. In this picture, the receiver bee is a male bee (drone). In the honeybee society, fluid food is frequently fed by worker bees to other workers, to drones and to the queen.

cation and regulation. Most prominently, eusocial insects (ants, bees, wasps and termites) have solved the problems imposed by mass-communication and massregulation in various ways [Wilson (1974); Hölldobler and Wilson (2008); Seeley (1995); Camazine et al. (2003a); Bonabeau et al. (1999b)]. On the one hand, these insects use direct communication, e.g., bee dances [von Frisch (1965)] and mouthto-mouth transfer of food ('trophallaxis', see [Camazine et al. (1998); Crailsheim (1998)]) to establish a robust and scalable regime of communication. In addition, these insect societies frequently exploit cue-based communication, where one single individual 'reads' just local cues in the environment [Seeley (1995)]. In contrast to the classical definition of communication [Shannon (1948)], where a welldefined message has to pass from a sender to a receiver, such cues are signals or artefacts that are not sent intentionally by any sender. Cue based communication is communication without message. It is self-regulation by simply observing what others do, or what others did. However, these cues were found to be valuable sources of information, which can inform one local worker quite well about the global status of the colony. For example, honeybees exploit queueing delays, which arise after foraging trips while waiting for unloading bees, to inform themselves about the current income status of the colony. This information is then used to decide whether to recruit new foragers with waggle-dancing or to recruit new nectar processors with tremble-dancing [Seeley (1989, 1992)]. The properties of this cue-based regulation system were studied intensively by individual based modelling [Anderson and Ratnieks (1999a,b); Ratnieks and Anderson (2000); Schmickl and Crailsheim (2008b,a); Thenius et al. (2008a,b)], demonstrating the communicational value of reading environmental cues. To give another example, some paperwasp foragers use the incoming flow of water, which is shared among wasps in the nest, to regulate their foraging activity for water and for wood [KARSAI and BALZSI (2002)]. In many cases, physical traces are left in the environment to act as local cues that inform other workers in a 'broadcast' scheme of communication.

12.5. Case study 2: From Information-sharing Trophallaxis to Energy-sharing Trophallaxis | 17

These communication principles are often called 'stigmergy' [Grasse (1959, 1967); Karsai and Penzes (1993); Karsai (1999); Karsai and Penzes (2000)]. For example, ants leave pheromone marks on the ground, when they recruit other ants to their food sources [Deneubourg et al. (1990); Goss et al. (1989); Dussutour et al. (2006)]. System theoretic analysis (modelling) of these communication networks revealed that it is a network of positive and negative feedback loops that allow these eusocial insect colonies to regulate their tasks in a rather homoeostatic way. These feedbacks act mostly localy and involve specific behavioural reactions of individual workers, which lead to a self-organization of the whole colony [Camazine et al. (2003b)]. Such behaviours that establish feedback loops can be as simple as: the more workers are at a specific target spot, the more cues they leave in the environment, the more other workers navigate to this place (positive feedback). And in contrast: the more workers are at this target spot, the longer it takes to reach that spot through overcrowding and the less un-recruited workers are available to be attracted by these cues (two negative feedback loops). Such simple behavioural patterns (attraction and saturation) lead to dynamic equilibria in task regulation or in foraging patterns, which represent classical building blocks in the regulation networks of social insects.

For swarm robots, it is difficult to leave physical (chemical) marks in the environment, which would allow them to perform real stigmergic approaches, although several attempts to achieve this goal have been made: In the study of [Russell (1997)], heat trails were used to achieve a 'stigmergic effect'. Other approaches used chemical substances, which were left behind by robots to mimic ant pheromone trails. This was done by using chemical markers [Russell (1999)] or by using a simple pen that paints on paper [J. Svennebring (2004)]. To mimic evaporation effects, which are also an important feature in ants' pheromone communication, some studies used 'disappearing ink' [Edelen (2003)] or glow paint [Blow (2005)]. Because real stigmegy is hard to achieve in robotics, some studies tried to mimic such environmental cues by using 'virtual pheromones', which are either projected from above onto the robot swarm [Sugawara *et al.* (2004); Garnier *et al.* (2007)] or which are communicated messages that are shared by local communication among the robots [Pearce (2003); D. Payton (2004)].

In contrast to the majority of published swarm robotic algorithms, which investigated cue-based communication by drawing inspiration from ant pheromone trails, we chose to model the usage of a different source of information in a robotic swarm: trophallaxis. The term trophallaxis refers to mouth-to-mouth transfer of fluid food among organisms. It is frequently found in birds that feed their siblings and also in some bats [Wilkinson (1984)], that exchange collected blood and establish a social 'security' network this way, keeping unsuccessful bats alive. In social insects, trophallaxis plays an important role to establish and to regulate a highly complex society [Camazine *et al.* (1998); Crailsheim (1998)]: ants, termites, wasps and bees frequently share food or water among nestmates. The study of [DeGrandi-Hoffman and Hagler (2000)] shows that a limited amount of nectar quickly reaches almost all colony members in a honeybee colony. Also in other eusocial insect colonies, the same fast distribution of material is assumed to be

a consequence of observed trophallactic events. Despite trophallaxis, also solid items of food (prey pieces, grass or leaf fragments) and building materials are often passed over from one nest mate to another, this way establishing a system of task partitioning [Ratnieks and Anderson (2000)]. In parallel, frequent body contacts (e.g., antennation, licking of other nest mates) lead to transfer of pheromones, which originate from the colony's queen in all insect societies. This way the whole worker population of a colony is constantly informed about the queen's current status. In all instances, that nearest-neighbour exchange of material leads to a diffusion-like distribution process in the society. If material is constantly added to the system at a specific place (foraging, queen) and if it is decaying over time (evaporation, degradation, consumption, usage), this process will automatically form a colony-wide gradient which will point uphill to the source of material. The swarm robotic algorithm that we present in this section is also based on a gradient that is established by near-neighbour communication among robots.

12.5.1 The swarm robotic task: A collective cleaning scenario

In recent studies, we examined several honeybee-derived algorithms for swarm robotics: The vector-based swarm algorithm draws inspiration from the dance language which is used by honeybees to communicate the location of feeding sites [Valdastri et al. (2006); Corradi et al. (2009)]. The BEECLUST algorithm is inspired by the self-organized aggregation behaviour of young honeybees in the hive [Schmickl et al. (2009); Kernbach et al. (2009a); Schmickl et al. (2009)]. The trohpallaxis-inspired algorithm is mimicking the frequent food exchange observable in honeybees [Schmickl and Crailsheim (2006); Schmickl et al. (2007a,c,b); Schmickl and Crailsheim (2008c)]. The algorithm that was analysed most intensively and which we present in the following sections is the called the "trophallaxis-inspired" robot algorithm. It is tested and analysed in a foragingcollectively-for-dirt scenario, which is closely related to the foraging-collectivelyfor-food task, which is prominently exhibited by in all eusocial insect colonies: in our focal scenario, a swarm of robots has to explore collectively the arena. If a robot finds one of those dirt particles, which are aggregated around some spot in the arena, it picks up that particle and transports it towards a designated dump area. All robots are able to perform only close-neighbour communication within a radius of 2-3 robot diameters. They are able to sense dirt particles and the dump area only directly below themselves on the ground, thus they have no far-range detection of dirt particles. The goal in the evaluated cleaning scenario was to control the swarm in a way that dirt collection is performed efficiently (fast) and robustly, which is indicated by a complete removal of dirt particles from the environment. Due to the limited sensory abilities of these robots, the only way to achieve this goal was by exploiting communication among neighbouring robots in a swarmintelligent way.

The focal robot for this scenario was the robot of the I-Swarm project [I-Swarm (2003-2007)], which is very small (millimetre-range) and which has very limited capabilities concerning precision of actuation, sensor reliability, communication bandwidth and computational speed. It is actuated by vibrating 3 legs, it can com-

12.5. Case study 2: From Information-sharing Trophallaxis to Energy-sharing Trophallaxis 19

municate with 4 local neighbours via light signals if they are in the line of sight, and it can transport small particles by using a small lever. Although the I-Swarm is made of small and rather limited robots, it was designed to be a powerful collective. The power of this swarm was sought to originate from the vast number of robots that would combine their efforts. In this sense, scalability of the used algorithm is also an important issue, as the I-Swarm was aimed at reaching population sizes of up to 1000 robots. This scalability requirement favours decentralized and self-organized algorithms over all conceivable centralized coordination principles. Figure 12.11. displays the basic spatial setup of the most simple experiment that we performed, as well as the final trail-following transportation scheme that resulted from executing the trophallaxis-inspired algorithm by the swarm robots. We performed our experiments with our simulation LaRoSim, which was parametrized in a way that it reflects the robot Jasmine [SWARMROBOT (2004-2010)], which is a spin-off product of the development of the I-Swarm robot platform. It mimics the communication principle of the I-Swarm robot and it is comparably imprecise in motion.



Figure 12.11. The tested swarm robotic scenario. (A) Starting condition. The dump in the upper right corner is still empty (yellow patches). All dirt particles are located at a spot in the lower left corner (blue patches). All robots are unloaded (red boxes) and explore the whole arena. (B) After some time the robots form a trail of loaded robots (blue boxes) which heads directly towards the dump. The unloaded robots continue to explore and to deliver information via the trophallaxis-inspired algorithm. Reprinted from [Schmickl and Crailsheim (2006)].

12.5.2 Deriving the trophallaxis-inspired algorithm from bio-inspiration

In our trophallaxis-inspired algorithm, all robots use two memory variables X_1^t and X_2^t , which represent their load of two distinct virtual substances at time step t. They ingest, store, consume and exchange these virtual substances under certain conditions, described as follows. We regard these substances to be two distinct kinds of 'virtual nectar', to link back to the honeybee domain. Each time step, the robots consume a fraction c_i of the substance X_i^t , thus decreasing their individual

values of the variable X_i^t . If an agent is located on a dirt particle, it adds an amount of a_1 to X_1^t . If an agent is located on the dump area, it adds an amount of a_2 to the variable X_2^t . Whenever two agents are within communication range, the one with the higher value of X_i^t donates a fraction D_i of the difference to the other agent, thus the two values of X_i^t get closer. Equation 12.4 gives the ODE that can describe the dynamics of the variables X_i^t in the focal robot j, which is located on the target (dirt area) and which communicates with a neighbouring robot k^1 . As robot j is located at the dirt area, it will have the 'addition values' set to $a_1 > 0$ and $a_2 = 0$. A robot located at the dump will have set these values to $a_1 = 0$ and $a_2 > 0$. At any other other location in the arena, all robots will use $a_1 = a_2 = 0$. In all circumstances, the 'consumption terms' and the 'diffusion terms' will be set to positive non-zero values: $D_i > 0$ and $c_i > 0$ with $i \in \{1, 2\}$. From these parametrisations, the following equation

$$\frac{X_{i,j}^t}{dt} = X_{i,j}^t (1 - c_i) + a_i + D_i (X_{i,j}^t - X_{i,k}^t)$$
(12.4)

establishes a swarm-wide gradient of X_1^t pointing uphill towards dirt particle locations and, in parallel, a swarm-wide gradient of X_2^t pointing uphill towards dump areas. With a frequency of $\frac{1}{\tau_1}$ time steps, an unloaded agent requests values of X_1^t from all visible neighbours, which indicates the the local gradient towards the dirt particles. This local information allows the agent then to turn into the direction of the neighbour with the highest value of X_1^t . With a frequency of $\frac{1}{\tau_2}$ time steps, a loaded agent collects the values of X_2^t from its neighbours, which indicates the the gradient towards the dump. This allows the agent to turn into the direction of the neighbour reporting the highest value of X_2^t . As $\tau_2 \ll \tau_1$, unloaded explorer robots do have some small bias towards the dirt area, but still cover huge parts of the arena for exploration. In contrast to that, loaded robots directly head on a trail-like route from the dirt particle areas to the dump areas, as shown in figure 12.11.b. Cumulative densities of robot locations in this scenario are depicted in figure 12.12.a,b. To further enhance the exploration capabilities of robots, we introduced a threshold parameter $\Theta_i^{aggregation}$. If $X_i^t < \Theta_i^{aggregation}$, robots move randomly, otherwise they move uphill in the local gradient. In addition, wider spreading of unloaded robots and trail-like motion of loaded robots was enhanced by introducing another threshold at which collision avoidance routines triggered an escape turn ($\Theta_{loaded}^{turn_away} > \Theta_{empty}^{turn_away}$). Figure 12.13. summarizes the algorithm described above graphically.

12.5.3 Collective perception within a robot swarm

The trophallaxis-inspired algorithm fuses the sensory inputs of robots (e.g. addition rates triggered by the presence of dump sites and dirt sites) in a swarm-wide manner by the diffusion of virtual nectar within the swarm collective. We were interested in whether or not this fact can lead to collective decision making based on

¹This ODE is solved by numerical forward integration in our simulation. Also on a micro-controller, this ODE will be solved by a time-discrete method.

12.5. Case study 2: From Information-sharing Trophallaxis to Energy-sharing Trophallaxis 21



Figure 12.12. *Cumulative paths of robots in the scenario depicted in figure 12.11.. (a) Cumulative density of unloaded robots. (b) Cumulative density of loaded robots. For these analyses we incre-mented localized counters in the arena, whenever a robot was located at this location. Finally, all counters were depicted as a map. The more colourful (and brighter) a patch is, the more often a robot was located at this patch. Reprinted from [Schmickl and Crailsheim (2008c)].*

environmental cues that are not recognizable for the individual swarm member. To achieve this, we generated two dirt spots, one left and one right of the arena. These spots were differing in size. In this experiment, the right spot was significantly bigger than the left spot. We altered the basic trophallaxis-inspired algorithm in a way that just one gradient was exploited by the robots, because we didn't allow them to pick up the dirt particles. In consequence, they just aggregated in high densities in the target areas. As is the case in all other experiments described here, the robots are able to locate a target patch – and later also the dump patches – only if they are located on it, thus there is no long-distance detection of these environmental features. As the bigger dirt spot on the right side of the arena can accommodate more robots, the total addition of a_1 into the swarm system is more frequent on the right side of the arena than on the left side. This leads to a steeper and higher gradient of X_1^r on the right side and to the attraction of more robots that aggregate at the bigger spot, as shown in figure 12.14.. We interpret the system in this way: the robots are a limited resource and the two target sites compete for these robots through positive feedback, which results from the emerging gradient and from the uphill movement. The bigger target spot on the right side of the arena imposes the stronger positive feedback, thus it finally wins the competition with the smaller target spot.

Both subfigures of figure 12.14. colour-code the intrinsic variable of X_1^t in the robots by shades of red. The brighter a robot is coloured, the higher the value of X_1^t is. With this experiment we showed that the swarm – as a total – is able to measure the areas of the two dirt patches and to make a collective decentralized decision, based on a global perception of the environment.

In our studies, we compared the trophallaxis-inspired algorithm to a simple algorithm, in which the robots just increment an 'hop-count' which allows them to measure how far the neighbouring robots are away from the aggrega-



Figure 12.13. Schematic representation of the trophallaxis-inspired algorithm. (a) Flowchart of the algorithm as it is executed by a robot i that is approaching a target. (b) Sensor model of the modelled Jasmine robot. (c) Schematic drawing how the gradient is built by inter-robot communication, omnipresent consumption and by locally restricted addition of virtual nectar X_t^i .

tion target. Such algorithms have been suggested before, sometimes called 'virtual pheromones' [Payton et al. (2001); ?]. Our experiments showed that swarms that use the hop-count algorithm are performing significantly worse in discriminating target zones by area size compared to the trophallaxis-inspired algorithm. The hop-count algorithm causes the swarm to break into two almost evenly sized fractions, regardless of target area size, while in the trophallaxis-inspired algorithm the vast majority of the swarm converged to the bigger target spot. In additional experiments, we varied the size differences between the two target spots and found (see figure 12.15.b) that the swarm that is using the trophallaxis-inspired algorithm is able to allocate fractions of the swarm population to the two target spots that reflect the ratio between sizes of the two target spots. In contrast, the 'hop-count' based algorithm failed to make such decisions. Swarms running this algorithm were not able to decide at all, as is shown by figure 12.15.a. Please note that we did count the aggregation result always in a circular area that was slightly bigger than the maximum-sized target spot and which was concentric with the target spots on each side. The area of this 'counting zone' was never changed, regardless how big the aggregation spot was.

Figure 12.16. shows a comparison of the emerging gradients produced by both





Figure 12.14. Collective perception favours collective decision making in swarms running the trophallaxis-inspired algorithm. (a) The experiment started with randomly distributed robots. After some time steps, the robots move in trail-like formations towards the two aggregation targets. The swarm collectively starts to select the right bigger aggregation site (blue area). Small red-white shaded boxes indicate robots. The brighter the colour of a robot the higher is its amount of stored virtual nectar. A single robot can only determine whether or not it is located on a blue target patch, no long-range detection of targets is possible. (b) After more time has passed, the swarm has made a clear collective decision and the majority of robots aggregated at the bigger target site. Reprinted from [Schmickl et al. (2007a)].



Figure 12.15. The experiments depicted in figure 12.14. were repeated with varying size-differences between the target spots. It was found that swarms were always able to reflect these environmental condition in its collective aggregation decision making if they used the trophallaxis-inspired algorithm. In contrast, the hop-count algorithm did not allow for this collective decision making. Reprinted from [Schmickl et al. (2007a)].

tested algorithms in an environment with a small aggregation site (r = 1 patch) at the left side of the arena and a big aggregation site (r = 5 patches) at the right side. It is shown that the hop-count algorithm does not change over time and is not able to depict any environmental features, as both differently sized target sites are barely reflected in the gradient. Please note that in the hop-count gradient, the





Figure 12.16. *Emerging gradient maps in both experiments shown in figure 12.15.. Top row: trophallaxis-inspired algorithm. Bottom row: hop-count based algorithm. left to right: Time evolution of the emerging gradients. Reprinted from [Schmickl* et al. (2007*a*)].

robots performed a gradient-descent, while they performed a gradient-ascent in the experiments with the trophallaxis-inspired algorithm. This is due to the fact that the hop-count algorithm provides a distance measurement to the target site (the closer to the target area, the smaller is the hop-count communicated by neighbours), while the trophallaxis-inspired algorithm establishes an 'emergent attraction field': the more virtual nectar a neighbour has, the more attractive it is to move in its direction. In comparison to the hop-count algorithm, the trophallaxisinspired algorithm shows a clear time evolution and the gradient map clearly reflects the given environmental situation (figure 12.16.). For more information on emergent gradient maps in the trophallaxis-inspired algorithm, refer to subsection 12.5.7. In additional experiments, we varied the level of $\Theta_i^{aggregation}$, which was found to control the final aggregation pattern in the following way: With low values of $\Theta_i^{aggregation}$ a minority of the swarm also aggregates at the smaller target site, while it totally neglects the smaller target site with higher values of $\Theta_i^{aggregation}$. Thus, the parameter $\Theta_i^{aggregation}$ can be used to parametrize how 'picky' the swarm behaves.

12.5.4 Preventing obstacles and shortest-path decision

Although the individual agents perform a simple gradient ascent in the emerging gradient field of the trophallaxis-inspired algorithm, it is still possible for them to circumvent obstacles. The 'intelligence' that is needed to achieve this relies not on individual decision-making in navigation, it originates from the 'clever' way in which the shared gradient map emerges within the swarm: if any obstacles block the way for the robots, it also blocks communication. Thus, a flux of 'virtual nec-

12.5. Case study 2: From Information-sharing Trophallaxis to Energy-sharing Trophallaxis 25

tar' exists due to the diffusion terms in equation 12.4, which automatically 'flows around' these obstacles. To test this, we constructed a more sophisticated arena setup, in which a diagonal wall blocks the way from the dirt area to the dump area. Only two 'gates' allow the robots to pass through this barrier (see figure 12.17.a). We observed that the robots establish trails through these gates (see figure 12.17.b), which is is an emergent property of the trophallaxis-inspired algorithm. In addition, we observed that such a swarm is able to collectively select the shortest path from dirt to dump, if the two gates offer pathways of differing length (see figure 12.17.c).



Figure 12.17. Gate experiment with the trophallaxis-inspired algorithm. (a) Arena setup with a diagonal wall. This wall can be passed only through two gates by the robots. The robots form trails that pass through these gates. (b) If both paths through these two gates are of identical length, the robots form two trails which are chosen by robots with approximately equal probability, as is shown by the cumulative density plots. (c) In an asymmetrical setting, robots automatically favour the shorter path. Reprinted from [Schmickl and Crailsheim (2008c)].

12.5.5 Quantitative analysis of swarm behaviours

We investigated the observed gate-choosing behaviour quantitatively by offering arena scenarios with differing degrees of asymmetry, as shown in figure 12.18.a. We observed in these experiments, that the collective behaviour of the swarm is a highly non-linear reaction to linear changes in the level of asymmetry. Another important aspect of swarm robotics is swarm density: if the habitat is populated too loosely, information cannot spread sufficiently through the swarm, thus it cannot perform in a 'swarm intelligent' manner. In contrast, if the habitat is populated too densely, then navigation of the robots is hindered and the swarm cannot perform efficiently.

Seeing this pragmatically, one could summarize the requirements for efficient swarm behaviour as follows: The swarm should be constituted in a way that local communication and collective information procession is maximized and that the triggering of collision avoidance routines is minimized. Such routines usually block the execution of the swarm algorithm, thus deviating the behaviour of the swarm away from the desired behaviour.

Our results depicted in figure 12.18.b clearly show that such an optimal swarm density does exist for our trophallxis-inspired algorithms. Those swarms that had a density of approximately 0.15, that is that 15% of the arena space is covered by

robots, showed the best collective behaviour in our case study cleaning task.



(a) Gate decisions with varying degree of (b) Swarm performance with varying swarm asymmetry in the environment. density.

Figure 12.18. *Quantitative analysis of swarm behaviour generated by the trophallaxis-inspired algorithm. Reprinted from [Schmickl and Crailsheim (2008c)].*

12.5.6 Individual adaptation enhances the swarm's abilities

In the basic version of the trophallaxis-inspired algorithms all individual metabolic rates (a_i , c_i , and D_i) are fixed throughout runtime. This restriction could be relaxed by allowing individuals to adapt one of these rates at runtime, for example if specific environmental cues are recognized. To investigate this, we enriched the environment with so called 'sticky patches' in which the actuators of the robots are assumed to slip or to stick, thus these areas slow down the robots. In such an environment, it can be better for the swarm, if loaded robots do not always choose the shortest path. Instead, they should choose the fastest path. In the trophallaxis-inspired algorithm, robots always follow the gradient map uphill. We expected it to be favourable, if such unfavourable areas are depicted as 'valleys' in the collective gradient map. To achieve this, we allow unloaded robots to increase the consumption rate (c_2) of the gradient that points uphill to the dump (X_2^t) with increasing stickyness of the local patch. Thus, we modified equation 12.4 by converting the former constant consumption rate c_i into a variable c_i^t

$$\frac{X_{i,j}^t}{dt} = X_{i,j}^t (1 - c_i^t) + a_i + D_i (X_{i,j}^t - X_{i,k}^t).$$
(12.5)

In this equation c_i^t is a time-dependent function of the current speed, as described by

$$c_{i}^{t} = c_{i}^{min} (1 + k \frac{v^{max} - v_{j}^{t}}{v^{max}}), \qquad (12.6)$$

12.5. Case study 2: From Information-sharing Trophallaxis to Energy-sharing Trophallaxis 27

where *k* is a scaling coefficient, c_i^{min} is the minimum consumption rate in the gradient *i*, v^{max} is the maximum robot speed and v_j^t is the actual motion speed of robot *j* in time step *t*.

Figure 12.19. shows a comparison between the algorithm without individual adaptation and with individual adaptation. The simple algorithm did not allow the loaded robots to circumvent the area of sticky patches, while the more elaborate version of the algorithm made such routing decisions possible.

Figure 12.19. Experiment with sticky patches. (a) Initial arena setup. red boxes: unloaded robots; blue patches: dirt particles; yellow patches: dump area; green patches: sticky area. (b) Cumulative density of loaded robots using the original trophallaxis-inspired algorithm without individual adaptation. (c) Cumulative density of loaded robots that adapt their consumption rate c_2 on sticky patches. The swarm clearly avoids those unfavourable habitats. Reprinted from [Schmickl et al. (2007b)].

Obviously, there is a continuous transition from a non-adaptive configuration k = 0 to a highly adaptive configuration k > 1. This allows the engineer to adjust the swarm to the given requirements in a real-world application. On the other hand, this variable k could be probably adaptive as well. In analyses that we performed so far, we kept the value of k a constant global swarm parameter. However, it was interesting to see if several values of k can lead to different adaptive swarm behaviours. Figure 12.20. shows that different non-zero values of k do indeed lead to different navigation decisions of the trail of loaded robots.

In addition, it is notable that sticky patches do not block communication of robots, which is in contrast to walls which block communication and movement simultaneously. It was interesting to investigate the interplay of walls and sticky patches in the same environment, leading to another environmental scenario: this time, the environment contained two larger areas of sticky patches. In a first run, there was an open passage between those two areas. The swarm of robots established a trail of loaded robots in a way that the robots navigated in an S-shaped path between the two sticky patch areas (see figure 12.21.b). In a second run, a wall was placed between those two sticky areas, thus blocking the passage between those areas. As a result, the swarm formed a trail which briefly crossed one of the sticky areas and circumvented the unfavourable habitats at the outer side of the arena (see figure 12.21.b).

Finally, we wanted to investigate how the trophallaxis-inspired algorithm allows a swarm to consider sticky areas in its collective decision making. Therefore,

Figure 12.20. Experiment with multiple sticky patches in the environment. (a) Initial arena setup. Blue patches: dirt particles; yellow patches: dump area; green patches: sticky areas. (b) Cumulative density of loaded robots using a moderate value of k = 10. (c) Cumulative density of loaded robots using a high value of k = 35. Reprinted from [Schmickl et al. (2007b)].

Figure 12.21. Experiment with multiple sticky patches and walls in the environment. (a) Arena setup. Blue patches: dirt particles; yellow patches: dump area; green patches: sticky areas; grey items: wall elements (b) Cumulative density of loaded robots when no wall was blocking the passage between the two sticky areas. (c) Cumulative density of loaded robots after a wall blocked this passage. Reprinted from [Schmickl et al. (2007b)].

we generated a scenario with two dump areas and one sticky area on the path between this dirt area and the dump area, and closer to the dirt (figure 12.22.a). With high 'stickyness' of these patches, that is robots were slowed down significantly, the robots collectively chose the other dump area as deposition site and created a trail that circumvented the sticky area (figure 12.22.b). If the robots were slowed down only slightly on the sticky area, the robots mainly crossed this area and collectively chose the closer dump site (figure 12.22.c). Thus we conclude from these experiments that swarms that run the trophallaxis-inspired algorithm are able to consider walls differently from other unfavourable features. Obviously, those swarms are able to react to continuous (non-discrete) environmental factors in an adaptive manner. 12.5. Case study 2: From Information-sharing Trophallaxis to Energy-sharing Trophallaxis 29

Figure 12.22. Swarm-level considerations about sticky patches in a complex environment. Experiment with sticky patches and multiple dump sites. (a) Arena setup. Blue patches: dirt particles; yellow patches: dump areas; green patches: sticky areas (b) Cumulative density of loaded robots when sticky areas decreased the robot speed significantly. (c) Cumulative density of loaded robots when sticky areas decreased the robot speed only slightly. Reprinted from [Schmickl et al. (2007b)].

12.5.7 Dynamic, complex and emergent maps of the environment

In the trophallaxis-inspired algorithm, obstacles and areas where robots adjust one of their intrinsic 'metabolic rates' (addition, consumption, diffusion/transfer-/communication) alter the swarm's behaviour significantly. Thus we assumed that these areas should be well depicted in the 'gradient map' that does not reside in single robots but within the whole swarm ('collective memory'). We created two additional scenarios to investigate the properties of the gradient map. In both scenarios, we used one dirt site and two equidistant dump sites. In the first scenario, two lengthy areas of sticky patches were placed on the robots' ideal routes from dirt to dump. One of these areas was rotated by 90 degrees, thus it required only a short passage through sticky terrain compared to the other arena side, in which the sticky area had to be traversed in full (see figure 12.23.a). The robots were able to make a two-fold collective decision. They collectively crossed the short path across the sticky terrain on the left arena side and mainly circumvented the sticky terrain on the right side of the arena (see figure 12.24.). Looking into the collective gradient map, we found that the sticky terrain was generating clearly visible valleys in the gradient, navigating locally the robots in a near-optimal way (see figures 12.23.b,c,d). Thus one connected swarm of robots was able to make two different decisions at the same time in different parts of the environment.

We repeated the same experimental setting in a second experiment, where we replaced the sticky terrain with walls that blocked not only movement but also communication between robots. In this case, the barriers were circumvented on both sides of the arena. As figure 12.25. shows, the collective gradient map represented the walls clearly as canyon-like, deep valleys, which differ significantly from the way sticky patches were represented. Thus we conclude that the swarm is able to represent environmental features qualitatively in a shared memory map, which allows the swarm to react in an adjusted manner to different environmental

30 Collective Foraging: Cleaning, Energy Harvesting and Trophallaxis

Figure 12.23. Environmental setup and maps of the emergent gradient maps in an experiment where two differently rotated areas of sticky patches were located between the dirt site and dump sites. (a) Arena with robots, dirt zones, dump zones and robots, as they look in the simulator. (b)Gradient map seen from above. (c)Gradient map seen from the left side. (d) Gradient map seen from the right side. For constructing the gradient map, we iterated over a grid of positions covering the arena space and recorded which values of X_2^t would be reported to a robot at this positions by its neighbours. Reprinted from [Schmickl et al. (2007c)].

Figure 12.24. *Cumulative robot density in the experiment depicted in figure 12.23.a. The swarm makes different decisions on either side of the arena, which are reflecting the different orientations of sticky areas in the arena. Reprinted from [Schmickl et al. (2007c)].*

qualities.

12.5.8 From 'information trophallaxis' to 'energy trophallaxis'

In the experiments described above, we used 'information trophallaxis' to control the efficient transportation of material. The agents moved according to the virtual gradient of X_1^t and X_2^t , and not according to an actual 'dirt gradient'. This is in contrast to social insects, where animals often do not communicate with ad-

12.5. Case study 2: From Information-sharing Trophallaxis to Energy-sharing Trophallaxis 31

Figure 12.25. Environmental setup and maps of the emergent gradient maps in an experiment where two differently rotated walls are blocking the direct way between the dirt site and dump sites. (a) Arena with robots, dirt zones, dump zones and robots, as they look in the simulator. (b)Gradient map seen from above. (c) Gradient map seen from the left side. (d) Gradient map seen from the right side. For constructing the gradient map, we iterated over a grid of positions covering the arena space and recorded which values of X_2^t would be reported to a robot at this positions by its neighbours. Reprinted from [Schmickl et al. (2007c)].

ditional signals or cues, instead they *reuse* existing cues for their communication efforts. This is efficient and thus expected to be favoured by natural selection. In our robotic scenario, it would be interesting if the robots' navigation would not be driven by a virtual gradient but by a gradient of dirt particles, as the robots collect these items anyway. However, we did not manage to construct such a pure 'stigmergic' approach so far.

For honeybees nectar is used to store and transfer energy, it is not primarily a substance used for communication purposes. Robots also rely substantially on energy supply, thus it would be interesting to transform the cleaning scenario into an energy harvesting scenario.

Imagine that the dirt areas are transformed into recharging stations, which are located somewhere in the environment. The dump area is transferred into a designated working spot, where robots have to perform a given task that consumes energy. Several options to configure such a swarm are plausible:

(1) The working robots perform their task until their energy reserves fall below a defined threshold Θ^{recharge}. At this point in time, they leave the working area and head towards a recharging place. In this swarm configuration, there is no division of labour in energy harvesting. However, the working robots could still be guided by a distributed network of cheaper guidance robots, which are spread in the environment. These robots could perform the trophallaxis-inspired algorithm and provide the gradient map that guides the expensive

working robots to the closest or richest recharging site. In this configuration, division of labour is present between those workers that harvest and spend energy, and those workers that act as pure communicating agents.

- (2) A different approach would be to establish more division of labour in the robot swarm. All dedicated worker robots can stay at the working site permanently, as a special group of energy-transfer robots establish a trail of robots that continuously oscillates between the recharging site and the working site. This scenario involves three-fold division of labour: Pure workers, pure energy harvesters and pure communicators, which guide the energy-harvesters by using the trophallaxis-inspired algorithm. Thus, this configuration allows the highest possible degree of hardware specialisation of robots.
- (3) However, there is another –and maybe more efficient– approach to this scenario: assume that the energy harvesting robots could transfer the energy also between each other in a trophallaxis-like manner. In this case there is no need to distinguish between energy harvesting and communication. Instead of virtual nectar that is communicated among the robots, the transferred energy is used as a communicational cue. In the following, we describe how such an 'energy-trophallaxis'-driven algorithm could operate.

Imagine a swarm that has to find energy sources (recharging stations) and in which energy can be passed from one robot to another. Thus, the virtual nectar concentration in a robot X_i^t becomes a real energy budget E_i^t in an accumulator set *i* inside this robot. Such robots will not need to calculate equation 12.4 at all. They will consume amounts of their energy per time step (equivalent to subtracting c_i every time step). They will gain new energy only at charging stations (equivalent to a_i). If we assume that robots will receive energy transfers only from robots of higher energy status compared to themselves, we have a real-world equivalent of $D_i(X_{i,i}^t - X_{i,k}^t)$. Automatically, a gradient of energy charging states will emerge within the robot swarm. To exploit this energy status for navigation, let us assume that robots blink their LEDs in a frequency which correlates with their energy status: The less energy a robot has, the slower it will move and the less frequently it will blink. We furthermore assume that robots, which fall below a defined energy threshold $E_i^t \geq \Theta_i^{recharge}$, move preferentially towards neighbours with the high LED blinking frequency, while robots with rich energy reserves $E_i^t \ge \Theta_i^{recharge}$ move preferentially towards neighbours with low blinking frequencies. As a consequence of these simple rules a turn-over behaviour of agents within the swarm will emerge, keeping the average energy level of the swarm around a homoeostatic set point. The level of this set point can be adjusted by shifting the value of $\Theta_i^{recharge}$. We predict that robots running such an algorithm will follow trails between recharging sites and sites of high energy consumption (working sites). As robots on the trail transfer energy from one to another, it is no longer necessary for individual robots to make the complete passage of these trails. Instead, it is the energy that will flow from the recharging site to the working areas, using the robots as a channel of energy flow and communication.

12.5. Case study 2: From Information-sharing Trophallaxis to Energy-sharing Trophallaxis 33

12.5.9 Future work and elaboration of the trophallaxis-inspired algorithm

By elaborating the system from information trophallaxis to energy trophallaxis, the swarm robotic algorithm moves closer to its biological source of inspiration. If the blinking signal, which communicates the energy status of a robot is strongly coupled to the energetic status of a robot, this signal becomes an 'honest signal', which is an important concept in interpreting animal behaviours and their evolution in biology. Our investigations showed that a weak point of the trophallaxis-inspired algorithm is that agents always have to trust other swarm members. The steady consumption of virtual nectar c_i establishes a steady decay of virtual nectar, which forces outdated, non-reinforced information to leave the system. When we introduced robots that had long-lasting faulty positives in target detection, they were overruled by other robots, which did not reinforce this information. Thus, singular faulty robots disturbed the system but did not prevent the swarm from working. However, if one robot increases its value of X_i^t to very high values, because it either performs very high addition acts a_i , because it performs no consumption for a long time $c_i = 0$ or because it communicates the value of X_i^t wrongly for other reasons, this could indeed harm the swarm's performance significantly. Thus, using an 'honest signal' in this situation would definitely increase the system's robustness. Such an 'honest signal' can be achieved by switching from 'information trophallaxis' to 'energy trophallaxis'. Another critical issue is the question of how to parametrize such a robotic algorithm. In the current system, we used evolutionary computation (evolution strategy) to optimize swarm-level parameters off-line in the simulator LaRoSim. We optimized those constants of the trophallaxis-inspired algorithm $(c_i, a_i, D_i, \Theta_i^{aggregation})$ on the one hand, and parameters such as swarm density on the other. As is shown by figure 12.18.b, such parameters are critical for the performance of swarms. After the transition from 'information trophallaxis' to 'energy trophallaxis', such an optimization could also be performed on-line on the running robotic hardware, as the supply with energy could be a good emerging 'fitness parameter' or 'cost function' for optimization. Such an on-line, on-board optimization is expected to achieve better results than any optimization achieved in simulation, as it bridges the reality gaps between models, simulation and reality. In addition, such optimization might be performing better on-board than in a simulator, as the optimization would be performed in parallel by many robots, the more robots there are in a swarm, the slower a sophisticated simulation gets, but the more computation power resides in the robotic hardware itself.

In conclusion, we explored information trophallaxis in robotic swarms intensively and found it to be extremely powerful and robust. By transforming these algorithms to exploit energy trophallaxis, the algorithm will become simpler and it is expected to achieve efficient, robust and flexible swarm behaviour by simple rules and by accounting for only locally available information. Thus, this family of swarm algorithms are clear examples of swarm intelligence in an engineering context.

12.6 Case study 3: A Kinetic model of swarm foraging

The previous sections of this chapter considered foraging models for discrete cases, see for instance the equation (12.3), where a chemical or electric energy is accumulated in the forms of batteries or energized particles. In this section we extend these models for case when robots forage electric energy by a direct recharging from docking stations. The recharging approach utilizes a kinetic relationship between behaviors of robots and an amount of energy required for these behaviors.

As demonstrated in chapter **??**, kinetic models make use of an analogy between the behavior of molecules in an ideal gas and the behavior of collective agents. This approach is useful for modeling interactions and information transfer in a swarm and for establishing relationships between number of agents N, swarm density D_{ws} , sensing and communication radii R_s , R_c , velocity of motion φ , consumed energy E^c and other values. The derived relationships can be tested in real robot swarms and verified in this way.

The topic of collective energy management based on kinetic models is discussed in Sec. ??, here we extend the topic to optimal energy foraging in swarms with both constant and variable densities. Fig. 12.26.(a) shows the Jasmine robot equipped with docking contacts, touch and energy sensors, and Fig. 12.26.(b) shows a typical setup for experiments in collective energy foraging. Without loss

Figure 12.26. (a) The micro-robot "Jasmine III" with docking contacts and touch sensors; (b) Experimental setup for collective energy foraging in a swarm of 50 micro-robots "Jasmine". The docking station, waiting zones and the working area are clearly visible.

of generality, we assume that robots can execute working tasks, look for docking stations, wait until recharging slots become free and then recharge. The behaviors in these tasks are denoted as "roles" R, so there are roles R_0 (working), R_1 (searching), R_2 (waiting) and R_3 (recharging). We denote further the number of robots that execute the role R_i as N_{R_i} and the duration of the role R_i on the robot j as $t_{R_{ij}}$ or in general t_{R_i} . The available individual energy E_i is estimated in ADC values of the corresponding voltage of the Li-Po accumulator. The efficiency Φ^j of the robot j can be calculated as

$$\Phi^{j} = \frac{t_{R_{0j}}}{t_{R_{0j}} + t_{R_{1j}} + t_{R_{2j}} + t_{R_{3j}}}.$$
(12.7)

Since charging and discharging currents in the Jasmine robot (i.e. time) are almost the same, thus $t_{R_{0j}} = t_{R_{3j}}$. When $t_{R_{1j}} = t_{R_{2j}} = 0$, the efficiency achieves $\Phi = \frac{1}{2} =$ 50%. When $t_{R_{3j}} = 0$, i.e. robot does not recharge and only works, its efficiency $\Phi = 100\%$. In this way the value of Φ is also useful for cases $t_{R_{0j}} \neq t_{R_{3j}}$ and expresses a general property of a robot's energetic balance.

Swarm efficiency Φ^s and the collective energy level of the swarm E^s are

$$\Phi^{s} = \frac{1}{N} \sum_{j}^{N} \Phi^{j}, \quad E^{s} = \frac{1}{N} \sum_{i}^{N} E_{i}, \quad (12.8)$$

where *N* is the number of robots and E_i is an individual energy level of a robot. The best swarm efficiency is achieved when $t_{R_1} = t_{R_2} = 0$; this is never achievable in real situations, i.e. $t_{R_1} \neq 0$ and $t_{R_2} \neq 0$ in real swarms. In the following two sections we estimate the swarm efficiency for different cases of R_1 and R_2 and formulate the requirements for a good collective foraging strategy.

12.6.1 Global energy homeostasis for a constant swarm density

Swarm density D^{sw} is defined as the relationship between the number of robots N and occupied by them area S_{sw} . Critical swarm density D_{crit}^{sw} can be derived from the assumption that robots cover the whole area S_{sw} (we can neglect here the area occupied by robots themselves), i.e. from $S_{sw} = N_{crit}^{max} \pi R_s^2$, where R_s is the sensing radius of a robot

$$N_{crit}^{max} = \frac{S_{sw}}{2\pi R_s^2}.$$
(12.9)

For Jasmine robot and for an arena $S_{sw} = 140 \times 115$ cm the critical maximal number of robots is 52. We can also estimate an optimal swarm density D_{opt}^{sw} from the assumption that for the best swarm reactivity (see [Kernbach *et al.* (2009b)]), robots should be within the communication radius R_c of each other

$$N_{opt} = \frac{S_{sw}}{2\pi R_c^2}.$$
 (12.10)

For the same Jasmine conditions, $N_{opt} = 23$. Maintaining $N \approx N_{opt}$ is of advantage because it provides several super-scalable parameters [Kernbach *et al.* (2009b)], see also chapter **??**. Therefore in this section we calculate the global energy homeostasis from the condition $N \approx N_{otp}$, i.e. with constant swarm density.

Let Ψ be the amount of energy, coming into the swarm from outside. The following inequality of the energy balance

$$E^s \le \Psi \tag{12.11}$$

says that the energy consumption should be less, or at least equal, to the energy input. The energy consumption E^c consists of parts spent for finding the energy source by N robots ($\omega t_{R_1}N$), for the waiting/docking by N robots ($\omega t_{R_2}N$) and finally for moving N/2 robots ($\omega t_{R_0}N/2$) until other N/2 robots are recharging ($\omega t_{R_3}N/2$). Setting for Jasmine robots $t_{R_0} + t_{R_3} = 2t_{R_0}$, we have

$$E^{s} = \omega N(t_{R_{0}} + t_{R_{1}} + mt_{R_{2}}), \qquad (12.12)$$

where the numeric coefficient m = 0, 1. In Sec. **??** we derived expression (12.12) and obtained

$$E^{s} = N\omega t_{R_{0}} + N^{2}\omega \frac{\pi R_{c}^{2} - S_{r}}{2R_{s}v} + N^{2}\omega m\lambda \leq \Psi, \qquad (12.13)$$

where m, λ are small coefficients. The linear term in (12.13) is in charge of doing some useful job, whereas the quadratic terms represent the energy required for supporting system-internal activities, such as collision avoidance. This means that collective systems have an optimal size at a constant swarm density. When a system grows further, i.e. it occupies more territory, the system-internal activities consume much more energy for internal activities than is required for useful output of the system.

12.6.2 Collective strategies for a variable swarm density

5

As demonstrated in the previous section, the constant swarm density case pays a high energetic price, expressed by the square terms in (12.13) denoting the system's internal activities. Therefore the main consideration for a good collective foraging strategy consists of reducing the impact of these terms. This can be achieved with a variable swarm density $N < N_{crit}^{max}$ and utilization of resulting effects. In particular, a higher swarm density allows reducing t_{R_1} by using the effect of overlapping trajectories.

As shown in the previous section, when the search area is S_{sw} , then the covering area S_{cov} of randomly moving robots can be estimated as a sum of non-overlapping local areas $S_l = vt2R_s$ (shown in Fig. 12.27.(a)) minus overlapping between S_l . There are two different reasons for overlapping. The first one is the swarm density, where one robot overlaps the trajectory of another robot (S_{ov1}). The second reason is the collision avoiding behavior of a robot itself, where it overlaps its own trajectory (S_{ov2}), i.e.

$$S_{cov} = NS_l - S_{ov1} - S_{ov2}.$$
 (12.14)

The relation $\frac{\sum S_{cov}}{S_{sw}}$ defines the goodness of a covering strategy. In Fig. 12.27. we demonstrate two cases of 2 and 13 robots moving randomly in an arena sized 140 × 115cm. When *t* is large enough, it is assumed $\sum S_l \approx S_{sw}$, whereas for short $t: \sum S_l < S_{sw}$. During a random motion, the local areas S_l overlap so that the efficiency of coverage is decreased. Obviously, a good strategy has to minimize overlapping between S_l . The overlapping is difficult to calculate exactly, however it can be estimated by using the following ideas. The *N* moving robots can be represented as *Nn* stationary robots in order to calculate a "differential image" as shown in Fig. 12.27., where *n* is a number of snapshots. The value of *n* increases continuously, so that n = kt, where *k* is a coefficient of "how often do we take snapshots during t". We assume that a snapshot only makes sense when robots move more than $2R_s$, i.e. $k = \frac{v}{2R_s}$. Randomly moving robots behave very similarly to gas molecules, i.e. they are uniformly distributed in the covering area at $t \to \infty$. Statistically, the areas S_l are not overlapped, when $\frac{2R_sS_{sw}}{Nvt} > \pi R_s^2(\frac{Ntv}{2R_s})$. Therefore the value S_{ov1} can be estimated as

$$S_{ov1} = \frac{\pi R_s^2 (Nvt)^2 - 4R_s^2 S_{sw}}{2R_s Nvt}.$$
(12.15)

12.6. Case study 3: A Kinetic model of swarm foraging 37

Figure 12.27. The area coverage of **(a)** 2 and **(b)** 13 robots. The figure **(b)** shows 10 seconds of motion (10 images) as difference images extracted from the video sequences.

The value S_{ov1} is valid when $S_{ov1} \ge 0$.

The value of S_{ov2} depends on the collision avoiding behavior of the robot itself. The point is that at each collision a robot rotates so that it moves partially along its own old trajectory. The lost area is proportional to the number of robot-robot contacts. At each contact a robot loses the area of a triangle, shown in Fig. 12.27.(a), which can be calculated as $\frac{R_s^2}{tan(\alpha/2)}$ or $\frac{R_s^2}{0.268}$ (α is collision avoiding angle $\approx 30^\circ$). The number of contacts *C* is equal to the average number of robots $\frac{N}{S_w}$ in the area S_l (with Maxwell coefficient $\sqrt{2}$), i.e.

$$S_{ov2} = \frac{2\sqrt{2}vtR_s^3N}{0.268S_{sw}}.$$
 (12.16)

Finally, we obtain the following expression of the covering area

$$S_{cov} = Nvt2R_s - \frac{\pi R_s^2 (Nvt)^2 - 4R_s^2 S_{sw}}{2R_s Nvt} - \frac{2\sqrt{2}vtR_s^3 N}{0.268S_{sw}}.$$
 (12.17)

In Fig. 12.28. we plot (12.17) in relation to time, when $S_{sw} = 16100cm^2$, v = 30cm/sec, $R_c = 15cm$ for two cases N = 2 and N = 13, as shown in Fig. 12.27.. As shown by this graph, 2 robots can cover the area in about 1 min, 13 robots need for this about 10 sec. Both numbers correlate very well with experimental measurement.

Equation (12.17) allows us to estimate the time t_{R_1} needed to cover some area for variable swarm density. Setting $S_l = S$, solving about t and simplifying, we obtain

$$t_{R_{1}} = \pm \frac{\sqrt{2} \left(mS_{sw} + \sqrt{m \left(mS_{sw}^{2} - 16 Rc^{2}mS_{sw} + 4 R_{c}^{2}mS_{sw}\pi + 16 \sqrt{2}R_{c}^{4} \right)} \right) S_{sw}}{R_{c} \left(-4 \sqrt{2}mS_{sw} + \sqrt{2}mS_{sw}\pi + 8 R_{c}^{2} \right) vN}$$
(12.18)

where m = 0.268. From two solutions we have to choose the positive one.

As follows from (12.17), a variable swarm density can provide better coverage and so a minimization of system's internal activities in (12.12). We do not expect

38 Collective Foraging: Cleaning, Energy Harvesting and Trophallaxis

Figure 12.28. Plot of covering rate from the relation (12.17).

that this approach can eliminate the square terms in expression (12.13), but it can minimize their influence on the common energy distribution. In the next section we formalize the requirements on a foraging strategy from the kinetic point of view.

12.6.3 Requirements for a good swarm foraging strategy

It is obvious, that a good foraging strategy should minimize times t_{R_1} and t_{R_2} as well as the number of robots N_{R_1} and N_{R_2} . This can be done in several ways.

1. There are several mechanisms which force swarms to D_{opt}^{sw} . In this case, the energy balance should be considered not as $\Psi \approx t_{R_0}N$ but as

$$\Psi \approx t_{R_0} S \approx t_{R_0} N^2, \tag{12.19}$$

The equation (12.19) means that energy input should be proportional to S_{sw} and should be uniformly distributed in S_{sw} - this is an important consequence allowing us to improve a foraging strategy. A uniform distribution enables the swarm to minimize system-internal activities.

2. It seems that a constant swarm density does not provide efficient energetic performance. In turn, this means that for $D^{sw} > D^{sw}_{opt}$ some swarm members will die because of bottlenecks and insufficient energy input. Individual energetic death is a self-regulating mechanism, allowing maximal collective energetic performance. Therefore a robot swarm should allow some robots to be sacrificed (e.g. to switch them into stand-by mode) for a better energetic performance.

3. The equation (12.18) for collective search provides shorter time t_{R_1} . However the more robots are involved in collective search, the worse will be their collective efficiency. The foraging strategy should maximize Φ^s by varying N_{R_1} in a collective search.

4. A good foraging strategy should minimize N_{R_2} and t_{R_2} by managing the number of robots N_{R_1} which go to recharging. In the ideal case N_{R_1} should be equal to the number of free slots in the docking station. The "buffered" robots should not cover the docking station.

12.6. Case study 3: A Kinetic model of swarm foraging **39**

5. A good foraging strategy should adapt the "critical energy" and "hungry" thresholds S_c and S_h to the current energetic state of the swarm and in this way reduce N_{R_2} .

From these five points, the two last ones have the greatest impact on collective foraging. In the next section we describe several experiments, which demonstrated a good strategy for optimizing Φ^s by managing N_{R_1} , N_{R_2} and t_{R_1} , t_{R_2} .

12.6.4 Experiments

We have performed several simulation and real robot experiments, with the intension of proving the following assumptions:

1. Varying swarm density $D^{sw} > D^{sw}_{opt}$ will lead to increasing efficiency, but also to dead robots.

2. A combination of different behavioral strategies can minimize the number of robots N_{R_1} going to recharge. This will in turn minimize N_{R_2} and t_{R_2} and should increase swarm efficiency.

3. Individual thresholds for critical and hungry states S_c and S_h can be adapted by considering $Pr(Task)_t$ related to local sensor data and local communication with neighbors. This can lead, at a collective level, to reduction of waiting robots N_{R_2} and so to higher efficiency.

The foraging strategy is implemented in the following way. It is firstly assumed that all robots possess exact knowledge about their working area and can precisely navigate in this area (so-called "all-knowing" robots). Additionally each robot is aware of the current energy state of all teammates. This allows precise coordination between robots. In the "all-knowing" scenario we distinguish four sub-cases ("best", "average" and "worst"), where available global information is gradually reduced. Finally, in the "not-knowing" case, there is no information available about the position of the docking station at all; robots can find the docking station and, as far as it is possible, in real robot experiments.

All real and simulated robots possess the following internal homeostasis, see Fig. 12.29.. Firstly, in the critical state, robots should break the currently executed activity and start looking for the docking station. Secondly, robots have a priority of currently executed activity Pr(Task) and priority of looking for food $Pr(S_h)$ ("hunger feeling"). When, for instance, the priority of current activity is 60%, but hunger is 70%, a robot will look for the docking station. Finally, a robot can have a so-called "collective instinct", it can recharge only until the satisfied state S_s (it takes less time), and make a slot free for another robot (otherwise it recharges until S_f). The energy values E^i of each robot can vary from 120 up to 181. At $E^i = 120$ the robot is considered to be "dead", at $E^i = 140$ the robot is in its "critical energy" state and should recharge as soon as possible. When $E^i \ge 181$ the robot is fully recharged. For all "all-knowing" robots we introduced two energy thresholds Th: Th_{hungry} and $Th_{recharged}$. These parameters differ through the experiments and determine at which internal energy level the robots start to "feel hungry" or "feel satisfied". We set the initial energy value E^i of each robot so that it exceeds Th_{hungry}.

Energy constrained environment with variable swarm density. To prove the assumption about variable swarm density, simulations with different numbers of robots in the "best" case scenario were conducted. The "best" case scenario is also the "most suitable" scenario for such a study for two reasons: (a) the robots have the best possible capabilities, so that finding the energy source is faster and the time for search can be assumed to be $t_s \approx 0$; (b) the perfect coordination avoids competition among robots, i.e. no robot has to wait and $t_w = 0$. Thus, if an environment does not have sufficient energy sources in the "best" case

constrained for all other cases, where

scenario, it will definitely be energy Figure 12.29. Structure of energetic homeostasis.

the robots have less information. For the study, the number of robots and their parameters are varied. Table 12.2 gives an overview of the parameter set.

Table 12.2. Parameters and results in "all-knowing" strategies, $Th_{recharged}$ =181 and Th_{dead} = 120.

Scenario	Ν	Th _{hungry}	Th _{crit}	Efficiency Φ^s [%]	Deaths
best 10	10	171	140	50	0
best 15	15	171	140	36	0
best 20	20	171	140	28,4	6
best h165	20	165	140	26,7	5,8
best c150	20	171	150	24,7	2
average 20	20	171	140	28,52	5
worst 20	20	171	140	15,04	0
not-knowing 20	20	171	140	16,02	0

Since the searching time in the "best" scenario is very low, $N_r/2$ docking stations can deliver sufficient energy for N_r robots, i.e. the efficiency is 50%. Increasing the number of robots to 15 in the "best 15" simulation reduces the efficiency to 36%. The collective energy fell and stabilized at 150 and there are no dead robots. In the "best 20" simulation, the ratio between stations and robots is 1:4. As shown in Fig. 12.30.(a), the swarm stabilized its efficiency at 28,4%, collective energy level around 140 and 6 robots "died" during the simulation. Since "dead" robots indicate insufficient energy in the area, this scenario was defined as an energy constrained environment, where the ratio of stations to robots is 1:4.

Influence of Th_{hungry} **and** Th_{crit} . The hungry threshold Th_{hungry} and the critical threshold Th_{crit} are thresholds for a robot to decide its own current energetic state. Now we explore how the choice of Th_{hungry} and Th_{crit} influences the performance. The "best" scenario is also used to compare different thresholds. Two

12.6. Case study 3: A Kinetic model of swarm foraging 41

Figure 12.30. Collective energy of different scenarios of "all-knowing" robots ("best h171" and "best c140" are used for comparison and are the "best 20" scenario).

simulations "best h165" and "best c150" were run with 20 robots, where Th_{hungry} and Th_{crit} were varied as shown in Table 12.2 and their collective energy is shown in Fig. 12.30.(b). We can see that the collective energy level does not depend on the chosen Th_{hungry} . In both simulations, the swarm was able to maintain its energetic homeostasis around the value 142. This was only possible because 6 robots "survive". Small differences appeared in the swarm efficiency, "best h165" - 26,7%, "best 20" - 28,4%.

Collective energy level for simulation with different Th_{crit} is shown in Fig. 12.30.(c). Since the critical threshold is the energy value at which robots discontinue their work, switch to the recharge procedure and try to preserve energy, it greatly influences the collective energy of the swarm. As Fig. 12.30.(c) shows, the level at which the swarm is able to maintain its energy constant builds around the chosen Th_{crit} . In the "best c150" simulation, the critical threshold has a higher value, so the robots switch to recharge role sooner than in the "best 20" simulation. This early interruption of work execution leads to lower efficiency (24,7% for "best c150") because the robots spend more time waiting for a free slot. Obviously, in that case they do not get really "exhausted", so only 2 of the robots died in the "best c150" case. Since requirements of energy foraging postulate maximal efficiency at minimal constant collective energetic level, the value $Th_{crit} = 140$ was used for all further simulations.

Collective energy vs. swarm efficiency. The collective energy for all "allknowing" scenarios is shown in Fig. 12.30.(d), whereas the swarm efficiency and the number of energetically "dead" robots are presented in Table 12.2. As expected, the "best 10" offers the best possible performance. When limitations are imposed on the available energy, as in the "best 20" case, the collective energy level falls to the critical threshold. There it is stabilized only because some robots die and swarm size is reduced to 75% of its initial value. The achieved efficiency is around 28%. Energetic performance is comparable for both scenarios. Removing knowledge about the energy state of other robots in the "worst 20" simulation has a deep impact on the swarm performance. Collective energy stabilizes around the Th_{hungry} without any "dead" members, but the efficiency falls to approximately 15%.

The results obtained are logical consequences of the underlying behavioral model. Robots in the "best" and "average" case scenarios are very social - they coordinate their actions with the other team members and respect their needs. This leads to longer working times for the ones that are not the "hungriest" in the swarm. These robots get finally "exhausted" because they keep working until reaching critical energetic state. In this way, the whole group achieves very high efficiency. When there is not enough "food", some individuals die.

In contrast, robots in the "worst" and "not-knowing" scenarios are egoistic creatures. As soon as robots are "hungry" they start searching for "food". Since there are not enough "food places" for all, the successful robots are "eating", whereas all other wait or struggle to reach the "food". Finally, everyone is busy with its own survival and less collective work is done. Such egoistic behavior allows the swarm to maintain a constant collective energy but the efficiency achieved is very low.

Real experiments. Real robot experiments have been performed in several arenas: variable arena between $0, 2m^2 - 0, 5m^2$, see Fig. 12.31.(a), and two fixed setups: small $110cm \times 85cm = 0,935m^2$, see Fig. 12.31.(b), and large $140cm \times 115cm =$ $1,61m^2$, see Fig. 12.32.. The number of robots varies between 3 and 50, experiments in the large arena have been performed with 30 and 50 robots. These experimental conditions allow us to explore different swarm densities. Additionally, real experiments aim to investigate different behavioral strategies, which optimize N_{R_1} and N_{R_2} and to use local communication between robots to adapt Pr(Task).

There are several docking setups: without landmarks, see Fig. 12.31.(a) and with different landmarks (robots, which indicate the docking station), see Figs. 12.31.(b) and 12.32.. The priority of recharging $Pr(S_{hungry})$ and $Pr(S_s)$ are set up to a maximum, so that when a robot is hungry, it breaks its current activity and looks for the docking station. In the docking station we implemented the two-line approach, clearly visible in Figs. 12.31.(b) and 12.32.(c),(d): the robots which receive a direct signal from a docking station navigate along this signal for docking and then start recharging. During recharging these robots in turn send a secondary signal with the meaning "docking station is here, but it is currently busy". This secondary signal has a larger covering area than the direct signal from the docking station. When another robot receives the secondary signal from a robot, it reduces

12.6. Case study 3: A Kinetic model of swarm foraging 43

Figure 12.31. (a) Experimental setup for variable arena. Shown is the two-line docking approach of a few robots for recharging: the first line – recharging robots, the second line – robots waiting for recharging; (b) Experimental setup for the small arena.

its own velocity and slightly rotates. In this way the waiting robots perform a local search – when a docking slot becomes free, a robot has a higher chance to find it. These waiting robots build a "recharging buffer" which allows a self-regulation in the swarm.

In order to adapt Pr(Task), the robot has to acquire feedback from the swarm. Feedback is collected by counting the messages received from working or waiting teammates. For estimation of the robot's need to stay or leave its current state, three priorities were introduced as in the simulation: *prioWorkTask*, *prioSearchTask* and *prioRechargeTask*. During work, the robot increases its *prioWorkTask* stepwise until the "hungry" threshold is reached. Then, the robot starts to increase the *prioSearchTask*. The speed of increase depends on the inertness coefficient which will be determined experimentally. When *prioSearchTask* exceeds *prioWorkTask*, the robot switches to the recharge role. The same procedure is followed while the robot is recharging and its energy exceeds $Th_{recharged}$. Then *prioWorkTask* competes with *prioRechargeTask* and when it exceeds it, the robot switches back to the work role.

To optimize N_{R_1} and N_{R_2} , we used different behavioral strategies, see more in [Kancheva (2007)], [Häbe (2007)], [Prieto (2006)], [Attarzadeh (2006)], [Jebens (2006)] as well as in [Kernbach *et al.* (2010)]. Table 12.3 summarizes several results across these experiments.

To summarize, experiments in unconstrained environments (for example 3 docking stations around the arena, large copper strips, as in Fig. 12.32.) demonstrate the best efficiency between 47,8% and 36,58%, and collective energy between 177,8 and 157,10. Energy-constrained environments lead to a drop of efficiency between 29,56% and 13,4%, and collective energy between 144,14 and 132,06. Variable swarm density leads to "dead" robots, whose number varies between 1 and 19. "Not-knowing" robots (i.e. robots performing random search strategy) have an average performance around 16%, whereas mixed behavioral strategies in energy-restricted environments provide almost twice the efficiency – around 30%.

(a) Initial set-up. Marked robots are landmarks (b) The macro-state 2 with the phase of collective with a large covering area. decision making.

(c) The macro-state 3, where the first "hungry" robots are approaching the docking station.

(d) The robots are docked and start recharging.

Figure 12.32. Collective energy foraging in a swarm of 50 micro-robots "Jasmine".

Table 12.3. Parameters and results for several experiments with 3, 6, 10, 30 and 50 robots, "mixed" – combined behavioral strategy, see more in [Kernbach *et al.* (2010)], "unconstrained" – energy-rich environment for 50 robots.

		Collective	Efficiency Φ^s	
Experiments	Ν	energy	[%]	Deaths
not-knowing 3	3	174,47	36,58	0
not-knowing 6	6	157,10	16,23	0
not-knowing 10	10	144,14	15,57	3,4
mixed 10	10	140,14	29,56	4
not-knowing 30	30	136,3	13,4	14
mixed 30	30	136,5	23,04	15
not-knowing 50	50	139,9	16,3	19
not-knowing (unconstrained) 50	50	166,15	38,14	0

12.6.5 Concluding remarks on the kinetic foraging model

The kinetic models considered here are useful in optimizing the foraging efficiency of a robot swam. Experiments confirmed that a variable swarm density, $D^{sw}=3,2$ - 18,63 in robot experiments, leads to unscalable behavior (e.g. bottleneck around D_{crit}^{sw}), but utilizes energetic self-regulating through dead robots. This mechanism finally increases collective energetic efficiency. Both theoretical and experimental data indicated a necessity to optimize N_{R_1} , N_{R_2} and t_{R_2} . Using several bio-inspired

12.7. Conclusion 45

behavioral strategies leads to almost a doubling of energetic efficiency in robot swarms. It was also demonstrated that adaptation of individual thresholds for critical and hungry states S_c and S_h by changing priorities of tasks Pr(Task) (i.e. feedback mechanisms) leads to better collective performance. There are a few other remarks:

- Collective knowledge greatly influences the robotic energy foraging performance, whereas exact localization abilities have little impact over it.
- Social robots, as in the "best" and "average" case strategies achieved a very good swarm efficiency but a poor collective energy level. High tolerance towards the needs of the other robots led to self-destruction. The agents worked until exhaustion and did not try to preserve their own energy. Thus, many agents died energetically and collective homeostasis was achieved at a very low level.
- Egoistic robots, as in the "worst" and "not-knowing" case strategies maintained their collective energy homeostasis at a high level but the swarm efficiency was minimal. In this society, the agents try to work as little as possible.

12.7 Conclusion

In this chapter we have provided a comprehensive overview of collective foraging. We began, in section 12.2, by defining robot foraging and setting out an abstract model for generalised collective robot foraging based on a Finite State Machine representation. We then surveyed strategies for cooperation in collective foraging, including information sharing, physical cooperation and division of labour. We presented three case studies. Case study 1, in Sect. 12.4, described the special case in which robots are foraging for their own energy – which is distributed as discrete 'food' items in the environment. Case study 2 in Sect. 12.5 introduced an information-sharing trophallaxis approach to collective cleaning (i.e. foraging for dirt) and the case study concluded with a proposal for extending the approach to true energy trophallaxis. Sect. 12.6 then described our 3rd case study – a kinetic model of collective foraging – in which the robot collective must maintain energy homeostasis by foraging for energy from fixed charging stations.

Although the fundamentals of collective robot foraging are now well understood, the engineering realisation of those principles remains a research problem. Consider multi-robot cooperative robot foraging. Although separate aspects have been thoroughly researched and demonstrated there has, to date, been no demonstration which fully integrates self-organised cooperative search, object manipulation and transport in unknown or unstructured real-world environments. Such a demonstration would be a precursor to a number of compelling real-world applications including search and rescue, toxic waste cleanup or foraging for recycling of materials.

The future directions for collective foraging robots lie along two separate axes. One axis is the continuing investigation and discovery of foraging algorithms — especially those which seek to mimic biologically inspired principles of self-

organisation. The other axis is the real-world application of foraging robots and it is here that many key challenges and future directions are to be found. Foraging robot teams are complex systems and the key challenges are in *systems integration and engineering*, which would need to address:

- (1) Principled design and test methodologies for self-organised collective foraging robot systems.
- (2) Rigorous methodologies and tools for the specification, analysis and modelling of collective robot foraging systems.
- (3) Agreed metrics and quantitative benchmarks to allow comparative evaluation of different approaches and systems.
- (4) Tools and methodologies for provable collective robot foraging stability, safety and dependability.

Bibliography

- Anderson, C. and Ratnieks, F. L. W. (1999a). Task partitioning in insect societies. I. effect of colony size on queueing delay and colony ergonomic efficiency, *The American Naturalist* 154, pp. 521–535.
- Anderson, C. and Ratnieks, F. L. W. (1999b). Worker allocation in insect societies: Coordination of nectar foragers and nectar receivers in honey bee (apis mellifera) colonies, *Behavioral Ecology and Sociobiology* 46, pp. 73–81.
- Attarzadeh, A. (2006). *Development of advanced power management for autonomous micro-robots* (Master Thesis, University of Stuttgart, Germany).
- Balch, T. and Arkin, R. C. (1994). Communication in reactive multiagent robotic systems, Autonomous Robots 1, pp. 1–25.
- Beni, G. (2005). From swarm intelligence to swarm robotics, in E. Şahin and W. M. Spears (eds.), Swarm Robotics - SAB 2004 International Workshop, volume 3342 of LNCS (Springer-Verlag, Berlin, Germany), pp. 1–9.
- Beni, G. and Wang, J. (2005). Swarm intelligence in cellular robotic systems, in *Proceedings of the NATO Advanced Workshop on Robots and Biological Systems*.
- Blow, M. (2005). Stigmergy: Biologically-inspired robotic art, in *Proceedings of the Symposium* on Robotics, Mechatronics and Animatronics in the Creative and Entertainment Industries and Arts (The Society for the Study of Artificial Intelligence and the Simulation of Behaviour).
- Blum, C. (2005). Ant colony optimization: Introduction and recent trends, *Physics of Life Reviews* 2, 4, pp. 353–373.
- Bonabeau, E., Dorigo, M. and Theraulaz, G. (1999a). Swarm Intelligence From Natural to Artificial Systems (Oxford Univ. Press).
- Bonabeau, E., Dorigo, M. and Theraulaz, G. (1999b). *Swarm intelligence: from natural to artificial systems* (Oxford University Press, New York).
- Bonabeau, E., Theraulaz, C. and Deneubourg, J.-L. (1996). Quantitative study of the fixed threshold model for the regulation of division of labor in insect societies, *Proceedings* of the Royal Society of London, Series B Biological Sciences **263**, pp. 1565–1569.
- Bonabeau, E., Theraulaz, G. and Deneubourg, J.-L. (1998). Fixed response thresholds and the regulation of division of labour in insect societies, *Bulletin of Mathematical Biology* **60**, pp. 753–807.
- Camazine, S., Crailsheim, K., N., H., Robinson, G., Leonhard, B. and Kropiunigg, H. (1998). Protein trophallaxis and the regulation of pollen foraging by honey bees (apis mellifera l.), *Apidologie* **29**, p. 113126.

- Camazine, S., Deneubourg, J.-L., Franks, N., Sneyd, J., Theraulaz, G. and Bonabeau, E. (2003b). *Self-Organization in Biological Systems* (Princeton University Press, Princeton, NJ, USA).
- Corradi, P., Schmickl, T., Scholz, O., Menciassi, A. and Dario, P. (2009). Optical networking in a swarm of microrobots, in *Nano-Net*, *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Vol. 3 (Springer Verlag,

Camazine, S., Deneubourg, J.-L., Franks, N., Sneyd, J., Theraulaz, G. and Bonabeau, E. (2003a). Self-organization in biological systems, URL http://books.google.com/books?hl=en&lr=&id= zMgyNN6Ufj0C&oi=fnd&pg=PP9&dq=self-organization+large+scale+ systems&ots=MJEAj0HDbd&sig=7rSeICURPzQURC6KTTWwYqsCCWw.

52 Bibliography

Berlin, Germany), pp. 107-119.

- Crailsheim, K. (1998). Trophallactic interactions in the adult honeybee (apis mellifera l.), *Apidologie* **29**, p. 97112.
- D. Payton, M. H., R. Estkowski (2004). Pheromone robotics and the logic of virtual pheromones, **3342**, pp. 47–57.
- DeGrandi-Hoffman, G. and Hagler, J. (2000). The flow of incoming nectar through a honey bee (apis mellifera l) colony as revealed by a protein marker, *Insectes Sociaux* **47**, pp. 302–306.
- Deneubourg, J.-L., Aron, S., Goss, S. and Pasteels, J. M. (1990). The self-organizing exploratory pattern of the argentine ant, *Journal of Insect Behavior* **3**, pp. 159–168.
- Deneubourg, J.-L., Goss, S., Pasteels, J. M., Fresneau, D. and Lachaud, J.-P. (1987). Selforganization mechanisms in ant societies (ii): learning in foraging and division of labour, *Experientia Suppl* 54, pp. 177–196.
- Dorigo, M. and Birattari, M. (2007). Swarm intelligence, Scholarpedia , p. 22361.
- Dorigo, M. and Şahin, E. (2004). Guest editorial: Swarm robotics, *Autonomous Robots* **17**, 2-3, pp. 111–113.
- Dorigo, M. and St/"utzle, T. (2003). *Ant colony optimization* (MIT Press, Cambridge, MS., London, UK).
- Dorigo, M., Tuci, E., Groß, T., Trianni, V., Labella, T., Nouyan, S. and Ampatzis, C. (2005). The SWARM-BOT project, in E. Şahin and W. Spears (eds.), *Swarm Robotics Workshop: State-of-the-art Survey*, no. 3342 in Lecture Notes in Computer Science (Springer-Verlag, Berlin Heidelberg), pp. 31–44.
- Dussutour, A., Fourcassié, V., Helbing, D. and Deneubourg, J.-L. (2006). Optimal traffic organization in ants under crowded condition, *Nature* 428, pp. 70–73.
- Edelen, M. R. (2003). Swarm Intelligence and Stigmergy: Robotic Implementation of Foraging Behaviour, Master's thesis, University of Maryland.
- Garnier, S., Tache, 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 Swarm Intelligence Symposium*, 2007. SIS 2007. IEEE (Honolulu, HI) (IEEE Press, Los Alamitos, CA), ISBN 1-4244-0708-7, pp. 37 – 44, doi:10.1109/SIS.2007.368024.
- Goldberg, D. and Matarić, M. J. (1997). Interference as a tool for designing and evaluating multi-robot controllers, in *Proc. 14th National conference on Artificial Intelligence (AAAI-97)* (MIT Press), pp. 637–642, URL citeseer.ist.psu.edu/ goldberg97interference.html.
- Goss, S., Deneubourg, S. A. J. and Pasteels, J. (1989). Selforganized shortcuts in the argentine ant, *Naturwissenschaften* **76**, pp. 579–581.
- Grasse, P.-P. (1959). La reconstruction du nid et les coordinations interindividuelles chez bellicositermes natalensis et cubitermes sp. la theorie de la stigmergie: essai dinterpretation du comportement des termites constructeurs, *Insectes Sociaux* **6**, p. 4183.
- Grasse, P.-P. (1967). Nouvelles experiences sur le termite de mller (macrotermes mlleri) et considerations sur la theorie de la stigmergie, *Insectes Sociaux* 14, p. 73102.
- Groß, R., Tuci, E., Dorigo, M., Bonani, M. and Mondada, F. (2006). Object transport by modular robots that self-assemble, in *Proc. IEEE International Conference on Robotics and Automation*, pp. 2558–2564.
- Häbe, D. (2007). *Bio-inspired approach towards collective decision making in robotic swarms* (Master Thesis, University of Stuttgart, Germany).
- Hölldobler, B. and Wilson, E. (2008). The Superorganism: The Beauty, Elegance, and Strangeness of Insect Societies (W. W. Norton and Company), ISBN 978-0393067040.
- I-Swarm (2003-2007). Intelligent Small World Autonomous Robots for Micro-manipulation, 6th Framework Programme Project No FP6-2002-IST-1 (European Communities).
- Ijspeert, A., Martinoli, A., Billard, A. and Gambardella, L. (2001). Collaboration through the exploitation of local interactions in autonomous collective robotics: The stick pulling experiment, *Autonomous Robots* 11, 2, pp. 149–171.
- J. Svennebring, S. K. (2004). Building terrain-covering ant robots: A feasibility study, Au-

tonomous Robots **16**, pp. 313–332.

- Jebens, K. (2006). *Development of a docking approach for autonomous recharging system for microrobot 'Jasmine'* (Studienarbeit, University of Stuttgart, Germany).
- Kancheva, T. (2007). *Adaptive role dynamics in energy foraging behavior of a real micro-robotic swarm* (Master Thesis, University of Stuttgart, Germany).
- Karsai, I. (1999). Decentralized control of construction behavior in paper wasps: an overview of the stigmergy approach, *Artificial Life* **5**, 2, pp. 117–136.
- KARSAI, I. and BALZSI, G. (2002). Organization of work via a natural substance: Regulation of nest construction in social wasps, *Journal of Theoretical Biology* 218, 4, pp. 549 – 565, doi: DOI:10.1006/jtbi.2002.3099, URL http://www.sciencedirect.com/science/ article/B6WMD-46YHVSH-F/2/2f55f4941a03acb1f67538c717e2d13d.
- Karsai, I. and Penzes, Z. (1993). Comb building in social wasps: Self-organization and stigmergic script, *Journal of Theoretical Biology* **161**, pp. 505–525.
- Karsai, I. and Penzes, Z. (2000). Optimality of cell arrangements and rules of thumb in cell initiation in polistes dominulus: a modeling approach, *Behav Ecol* **11**, 4, pp. 387–395.
- Kennedy, J. and Eberhart, R. (2001). *Swarm Intelligence* (Morgan Kaufmann Publishers: San Francisco).
- Kernbach, S., Nepomnyashchikh, V., Kancheva, T. and Kernbach, O. (2010). Specialization and generalization of robotic behavior in swarm energy foraging, *under consideration in "Journal of Experimental and Theoretical Artificial Intelligence"*, *Taylor & Francis*.
- Kernbach, S., Thenius, R., Kernbach, O. and Schmickl, T. (2009a). Re-embodiment of honeybee aggregation behavior in an artificial micro-robotic system, *Adaptive Behavior* **17**, 3, pp. 237–259.
- Kernbach, S., Thenius, R., Kernbach, O. and Schmickl, T. (2009b). Re-embodiment of honeybee aggregation behavior in artificial micro-robotic system, *Adaptive Behavior* **17**, 3, pp. 237–259.
- Krieger, M. and Billeter, J.-B. (2000). The call of duty: Self-organised task allocation in a population of up to twelve mobile robots, *Jour. of Robotics & Autonomous Systems* 30, pp. 65–84.
- Labella, T. H., Dorigo, M. and Deneubourg, J.-L. (2006). Division of labour in a group of robots inspired by ants' foraging behaviour, *ACM Transactions on Autonomous and Adaptive Systems* **1**, 1, pp. 4–25.
- Lerman, K. and Galstyan, A. (2002). Mathematical model of foraging in a group of robots: Effect of interference. *Autonomous Robots* **13**, 2, pp. 127–141.
- Lerman, K., Jones, C., Galstyan, A. and Matarić, M. (2006). Analysis of dynamic task allocation in multi-robot systems, *International Journal of Robotics Research* 25, 3, pp. 225–242.
- Liu, W. and Winfield, A. F. T. (2010). Modelling and Optimisation of Adaptive Foraging in Swarm Robotic Systems, *International Journal of Robotics Research* doi:10.1177/ 0278364910375139.
- Liu, W., Winfield, A. F. T. and Sa, J. (2009). A macroscopic probabilistic model of adaptive foraging in swarm robotics systems, in *Proc. of 6th Vienna International Conference on Mathematical Modelling (Mathmod 2009): Special Session on Modelling the Swarm* (Vienna).
- Liu, W., Winfield, A. F. T., Sa, J., Chen, J. and Dou, L. (2007). Towards energy optimisation: Emergent task allocation in a swarm of foraging robots, *Adaptive Behaviour* 15, 3, pp. 289–305.
- Martinoli, A., Easton, K. and Agassounon, W. (2004). Modeling swarm robotic systems: A case study in collaborative distributed manipulation, *International Journal of Robotics Research* 23, 4, pp. 415–436.
- Matarić, M. J. and Marjanovic, M. J. (1993). Synthesizing complex behaviors by composing simple primitives, in *Proc. Self Organization and Life, From Simple Rules to Global Complexity*, European Conference on Artificial Life (ECAL-93), Brussels, pp. 698–707.
- Melhuish, C. (1999). Employing secondary swarming with small scale robots: a biologically

53

54 Bibliography

inspired collective approach, in *Proc. of the 2nd Int.Conf. on Climbing & Walking Robots CLAWAR*.

- Millonas, M. M. (1994). Swarms, phase transitions, and collective intelligence, in C. G. Langton (ed.), Artificial Life III (Addison-Wesley, Reading, MA.).
- Mondada, F., Gambardella, L. M., Floreano, D., Nolfi, S., Deneubourg, J.-L. and Dorigo, M. (2005). The cooperation of Swarm-bots: Physical interactions in collective robotics, *IEEE Robotics and Automation Magazine* **12**, 2, pp. 21–28.
- Parker, L. E. (1994). ALLIANCE: an architecture for fault tolerant, cooperative control of heterogeneous mobile robots, in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 776–783.
- Payton, D., Daily, M., Estkowski, R., Howard, M. and Lee, C. (2001). Pheromone robots, *Autonomous Robots* 11, pp. 319–324.
- Pearce, P. E. S. S. A. P. N., J. L. Rybski (2003). Dispersion behaviors for a team of multiple miniature robots, in *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, Vol. 1, pp. 1158–1163.
- Prieto, V. (2006). *Development of cooperative behavioural patterns for swarm robotic scenarios* (Master Thesis, University of Stuttgart, Germany).
- Ratnieks, F. and Anderson, C. (2000). Task partitioning in insect societies, *Insectes Sociaux* **46**, pp. 95–108.
- Russell, R. A. (1997). Heat trails as short-lived navigational markers for mobile robots, *Robotics and Automation* 4, pp. 3534–3597.
- Russell, R. A. (1999). Ant trails an example for robots to follow? in *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, Vol. 4, pp. 2698–2703.
- Sahin, E. (2005). Swarm robotics: From sources of inspiration to domains of application, in *International Workshop on Swarm Robotics*, pp. 10–20.
- Şahin, E. and Winfield, A. (2008). Special issues on swarm robotics, Swarm Intelligence 2, 2-4, pp. 69–72.
- Schmickl, T. and Crailsheim, K. (2006). Trophallaxis among swarm-robots: A biologically inspired strategy for swarm robotics, in *Proceedings of the 1st IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechanotronics*.
- Schmickl, T. and Crailsheim, K. (2008a). Analysing honeybees' division of labour in broodcare by a multi-agent model, in S. Bullock, J. Noble, R. Watson and M. A. Bedau (eds.), *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems* (MIT Press, Cambridge, MA), pp. 529–536.
- Schmickl, T. and Crailsheim, K. (2008b). Taskselsim: a model of the self-organization of the division of labour in honeybees, *Mathematical and Computer Modelling of Dynamical Systems* 14, pp. 101–125.
- Schmickl, T. and Crailsheim, K. (2008c). Trophallaxis within a robotic swarm: bio-inspired communication among robots in a swarm, *Autonomous Robots* **25**, 1-2, pp. 171–188.
- Schmickl, T., Möslinger, C. and Crailsheim, K. (2007a). Collective perception in a robot swarm, in E. Şahin, W. Spears and A. F. T. Winfield (eds.), *Swarm Robotics - Second SAB 2006 International Workshop*, *LNCS*, Vol. 4433.
- Schmickl, T., Möslinger, C., Thenius, R. and Crailsheim, K. (2007b). Bio-inspired navigation of autonomous robots in heterogenous environments, *International Journal of Factory Automation, Robotics and Soft Computing* 3, pp. 164–170.
- Schmickl, T., Möslinger, C., Thenius, R. and Crailsheim, K. (2007c). Individual adaptation allows collective path-finding in a robotic swarm, *International Journal of Factory Automation, Robotics and Soft Computing*, pp. 102–108.
- Schmickl, T., Thenius, R., Möslinger, C., Radspieler, G., Kernbach, S. and Crailsheim, M. S. K. (2009). Get in touch: cooperative decision making based on robot-to-robot collisions, *Autonomous Agents and Multi-Agent Systems* 18, 1, pp. 133–155.
- Seeley, T. D. (1989). Social foraging in honey bees: how nectar foragers assess their colonys nutritional status, *Behavioral Ecology and Sociobiology* **24**, pp. 181–199.
- Seeley, T. D. (1992). The tremble dance of the honey bee: message and meanings, Behavioral

Ecology and Sociobiology **31**, pp. 375–383.

- Seeley, T. D. (1995). *The wisdom of the hive: the social physiology of honey bee colonies* (Havard University Press, Cambridge, Massachusetts, London, England), ISBN 0674953762.
- Shannon, C. (1948). A mathematical theory of communication, *Bell System Technical Journal* **27**, pp. 79–423.
- Sugawara, K., Kazama, T. and Watanabe, T. (2004). Foraging behavior of interacting robots with virtual pheromone, in *Proc. of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE Press, Los Alamitos, CA).
- SWARMROBOT (2004-2010). SWARMROBOT Large-Scale Robotic Swarm Jasmine (University of Stuttgart).
- Thenius, R., Schmickl, T. and Crailsheim, K. (2008a). How to know without having been there? investigating communication channels in the nectar collecting system of a honeybee colony (abstract), in S. Bullock, J. Noble, R. Watson and M. A. Bedau (eds.), Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems (MIT Press, Cambridge, MA), p. 807.
- Thenius, R., Schmickl, T. and Crailsheim, K. (2008b). Optimisation of a honeybee-colony's energetics via social learning based on queuing delays, *Connection Science* **20**, 2, pp. 193–210.
- 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, p. 789804.
- von Frisch, K. (1965). *Tanzsprache und Orientierung der Bienen* (Springer Verlag, Berlin, Heidelberg, New York).
- Wilkinson, G. S. (1984). Reciprocal food sharing in the vampire bat, Nature 308, pp. 181–184.

Wilson, E. O. (1974). The Insect Societies (Harvard Paperbacks)

- (Belknap Press), ISBN 0674454952, URL http://www.amazon.ca/exec/obidos/ redirect?tag=citeulike09-20\&path=ASIN/0674454952.
- Winfield, A. F. T. (2009). Foraging robots, in R. A. Meyers (ed.), *Encyclopedia of Complexity* and System Science (Springer; New York), pp. 3682–3700.

55