From Honeybees to Robots and Back: Division of Labor based on Partitioning Social Inhibition

Payam Zahadat*, Sibylle Hahshold, Ronald Thenius,
Karl Crailsheim, Thomas Schmickl
Artificial Life Lab
of the Department of Zoology,
Universitätsplatz 2,
Karl-Franzens University Graz,
8010 Graz, Austria
Phone: +43 316 380 3982
{payam.zahadat, thomas.schmickl}@uni-graz.at

Abstract

In this paper, a distributed adaptive partitioning algorithm inspired by division of labor in honeybees is investigated for its applicability in a swarm of underwater robots in one hand and is qualitatively compared with the behaviour of honeybee colonies on the other hand. The algorithm, Partitioning Social Inhibition (PSI), is based on local interactions and uses a simple logic inspired from age-polyethism and task allocation in honeybee colonies. The algorithm is analyzed in simulation and is successfully applied here to partition a swarm of underwater robots into groups demonstrating its adaptivity to changes and applicability in real world systems. In a turn towards the inspiration origins of the algorithm, three honeybee colonies are then studied for age-polyethism behaviours and the results are contrasted with a simulated swarm running the PSI algorithm. Similar effects are detected in both the biological and simulated swarms suggesting biological plausibility of the mechanisms employed by the artificial system.

1 Introduction

Social insects are promising sources of inspiration in the field of swarm intelligence and collective robotics. Such insects are highly capable of self-organization and self-regulation of their colonies and adaptation to changes in the internal and environmental conditions. One of the prominent characteristics of social insects is division of labor happening in, e.g., honeybee colonies [34, 14], wasps [37], termites [8] and ants [13, 20].

*corresponding author

These insects maintain plasticity in their colony structure meaning that the partitioning of the colonies into task-groups are adaptable to internal status such as the age of the workers or the amount of brood as well as the environmental status such as availability of food. The adaptivity and quick response to the changes in the status of colony and maintaining specialization for the workers are interesting features of the methods driving behaviours of the colonies of social insects.

Different models have been proposed to explain the mechanisms of regulation of division of labor in social insects, e.g. foraging-for-work [36], response-threshold reinforcement [5, 35], and common-stomach models [22] (for a review of such models see [3]). A set of models were also proposed based on the concept of social inhibition [14, 16]. In the social inhibition models, inhibitory effects on behavioral development in social insects are taken into account.

In the field of swarm robotics where a large number of relatively simple robots self-coordinate to perform tasks, social insects are an important source of inspiration [32, 29]. For example, in [12] and [23], models of ants-foraging behaviour are implemented for maintaining division of labor in a group of robots performing an object retrieval task. Response-threshold reinforcement model [5, 35] which is inspired by wasps is applied in robot swarms, e.g. in [19, 38, 42]. In [33] a trophallaxis-inspired strategy which is inspired by food exchange of honeybees and ants is applied to a swarm of robots in simulation.

In this work, we consider an algorithm of division of labor called Partitioning Social Inhibition (PSI) previously proposed in [43]. In designing the algorithm we were interested in a number of features. The main feature was the general ability of the decentralized algorithm to split a swarm of agents into subgroups where each subgroup is assigned to a task and the size of the subgroups are set according to the relative demands for the tasks. The size of the subgroups should be adaptive to the task-demands and available workforce during run-time. In addition, specialization should be maintained in the system, i.e. unnecessary switching between different subgroups should be limited due to practical costs (e.g., a robot may need to spend some energy and time to move to another region when it switches to a different task; or a honeybee needs a number of physiological and morphological changes in order to perform a different task [17, 25]).

The algorithm is inspired from the behaviour of honeybees and it works based on local interactions between agents. It was previously [43] used to regulate the number of agents that undertake different tasks in a swarm in abstract simulation. The adaptivity of the algorithm against internal and environmental changes in the swarm and the low number of switching of agents between different tasks, which enables specialization, makes it promising for real robot applications.

The algorithm is implemented here in a swarm of Autonomous Underwater Vehicle (AUV) robots with

local communication abilities. It exhibits a successful partitioning of the swarm into different task-groups and a high adaptivity of the swarm to the changes in the number of robots in presence of real-world communication noise. The robotic experiment is then implemented in simulation with different number of robots and the internal dynamics of the algorithm variables are depicted. The algorithm is then analyzed for different parameter settings.

With a turn towards the biological origins of the algorithm, we also report results from experiments with a number of real honeybee colonies and simulated swarms of agents running the PSI algorithm. The colonies were faced a special human-induced situation, namely human-induced swarming event, which dramatically changes the age demography of the colony. Behaviours of the simulated swarm of agents that run the algorithm and the real colonies are then compared after the swarming event and indicates qualitatively similar effects of social inhibition in both cases.

2 Social Inhibition in Honeybees

A worker honeybee undertakes different tasks during its life-time (see Figure 1 for a brief illustration). The honeybee develops morphologically and physiologically according to the task it undertakes [10, 39, 26]. In a normal situation there is a correlation between the age of the honeybee and the task it performs [21, 30, 41, 4]. This mechanism of behavioral development is called *age-polyethism* (or *temporal polyethism*). Earlier in the adult life, a honeybee performs nursing in a normal colony. Then it physiologically grows to undertake other tasks inside the hive. Only in its final weeks, the honeybee gets physiologically old enough to leave the hive for foraging [39, 31, 18]. The behavioural development can be delayed, accelerated, or even reversed in response to changes in colony or environmental conditions.

It has been shown [15] that honeybee colonies are adaptive to age distribution of the colony and also to the changes in the task demands. For example, in a colony of young honeybees, the age in which a bee starts foraging (an outside task) is lower than in a normal colony. It means the behavioral development in such a colony is accelerated. On the other hand, presence of older bees delays or inhibits the development of the physiological age of other bees in the colony. Another example is the behaviour of colonies when the hive workers are removed. In this case, the development of the physiological age decreases and inverts resulting in transformation of out-of-hive workers into inside-hive workers.

It has been proposed [14] that the interactions between workers drive mechanisms of hormonal regulation in honeybees and result in a social inhibition that explains age polyethism and adaptability of the colony to

different age distributions.

This concept is used in different researches toward developing models of social inhibition, e.g., [2, 27, 11].

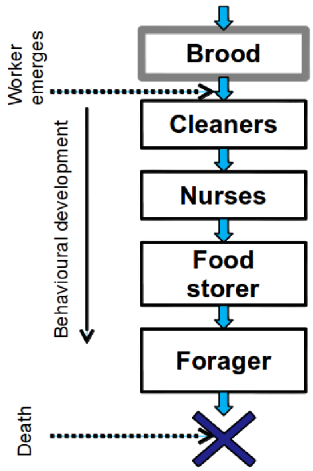


Figure 1: Behavioural development in brief.

3 Partitioning Social Inhibition

In the Partitioning Social Inhibition (PSI) algorithm, tasks are considered to be ordered in a sequence such that an agent is only allowed to switch to the previous or the next task in the sequence. Every agent in the swarm, contains a state variable x. This state variable is inspired from physiological age of honeybees. Since in honeybees, there is a correlation between the physiological age and the task the honeybee chooses to perform, here the agents choose their tasks based on their x value. The x variable can have a value in the range of $[x_{min}, x_{max}]$. If the x value of an agent reaches a threshold value, the agent switches to the next task. In fact, there are a number of thresholds defined in the range of $[x_{min}, x_{max}]$ that partition the range into a number of segments. Each segment is associated with a task in the task sequence (see Figure 2).

The thresholds are augmented with lower and upper margins (l_b and l_u respectively). That means, an agent in $task_i$ switches to $task_{i+1}$ if its x value exceeds $th_{i:i+1} + l_u$. The agent switches to $task_{i-1}$, if its x value becomes lower than $th_{i-1:i} - l_b$. The lower and upper margins prevent the agents from instant back and forth switches between two consecutive tasks due to noise.

The idea is to decentrally distribute the x values over the x-range while the agents keep the distance between their x values such that the number of xs in every task's segment is proportional to the demand for the associated task.

If the x values are uniformly distributed over the range and we set the thresholds such that the range is split into segments proportional to the task demands, the number of agents in each task will be proportional to the demands (Figure 2). The method is independent of the swarm size and therefore it is theoretically adaptive to any change in the workforce, e.g. adding or removing physiologically young (small x) or old (large x) agents to the swarm.

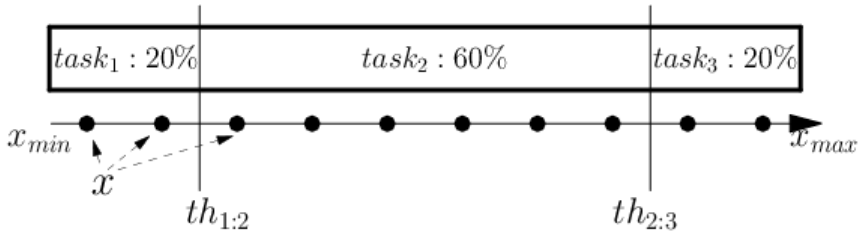


Figure 2: Uniform distribution of x over the range and proper thresholds that split the range relative to the task-demands. In this example, ten agents are divided into three tasks. The relative demands for the three tasks and therefore the relative number of the agents undertaking the tasks are 20%,60%, and 20% respectively.

The distribution of the x values over the x-range has to be done in a decentralized way via local communication between the agents. In order to achieve that, every agent contains two more variables, d^{lower} and d^{higher}, in addition to x. From an agent's point of view, d^{lower} and d^{higher} are estimations of the actual distances between its x value and the closest lower and higher x values among the other agents. When an agent meets another one, it receives the x value of the other agent. Based on this information, d^{lower}, d^{higher}, and consequently x of the agent might get updated. In order to get useful information for updating d values, communication is necessary between agents in the same task or the neighboring tasks in the sequence but it has no effect when the communicating agents are in tasks farther apart in the sequence.
If agent i that performs task t meets agent j, d_i^{lower} and d_i^{higher} are updated based on the distance between the x values of the two agents:

$$d_i^{lower} = \begin{cases} x_i - x_j & \text{if } (x_i - d_i^{lower}) < x_j \leq x_i \\ d_i^{lower} & \text{otherwise} \end{cases} \tag{1}$$

$$d_i^{higher} = \begin{cases} x_j - x_i & \text{if } x_i \leq x_j < (x_i + d_i^{higher}) \\ d_i^{higher} & \text{otherwise} \end{cases} \tag{2}$$

Additionally, d_i^{lower} and d_i^{higher} slowly increase in every step in order to be adaptable to changes in x values

of other agents as well as changes in the environmental conditions, i.e. changes in the workforce or task-demands:

$$d_i^{lower} = \begin{cases} d_i^{lower} + \kappa & \text{if } (d_i^{lower} + \kappa) \leq (x_i - x_{min}) \\ 2(x_i - x_{min}) & \text{otherwise} \end{cases} \tag{3}$$

$$d_i^{higher} = \begin{cases} d_i^{higher} + \kappa & \text{if } (d_i^{higher} + \kappa) \leq (x_{max} - x_i) \\ 2(x_{max} - x_i) & \text{otherwise} \end{cases}$$

where κ is a small value in terms of segments sizes. The d values at the boundaries are updated such that in a uniform distribution, there is no difference between the boundaries (x_{min} and x_{max}) and the threshold values th. For example in Figure 2, note the equality of the distance between the minimum x and x_{min} with the distance between $th_{1:2}$ and either of the x values next to it.

The gradual drift defined by Eq. 3 occurs before Eq. 1 and Eq. 2. This way, the window range $[x_i - d_i^{lower}, x_i + d_i^{heigher}]$ is first widened by using Eq. 3 and then x_j is compared against it by Eq. 1 and Eq. 2.

Every agent tends to keep its x value equally distanced from the closest higher and lower x values of the others. Therefore, with every update of the d^{lower} and d^{higher}, the x value is also updated, as follows:

$$x = \begin{cases} x + \delta & \text{if } d^{lower} < d^{higher} \\ x - \delta & \text{if } d^{lower} > d^{higher} \\ x \pm X & \text{otherwise} \end{cases} \tag{4}$$

where δ is *step-size* which is a constant parameter with a small value in terms of the size of task segments. In the current implementation $X \sim \delta \times U(0, 1)$.

After every update of x, an agent considers switching to the previous or next tasks in the task sequence. For an agent in $task_t$, new_task is chosen as follows:

$$new\_task = \begin{cases} task_{t+1} & \text{if } x > th_{t:t+1} + l_u \\ task_{t-1} & \text{if } x < th_{t-1:t} - l_b \\ task_t & \text{otherwise} \end{cases} \tag{5}$$

where $th_{t-1:t}$ and $th_{t:t+1}$ represent the threshold values between $task_{t-1}$ and $task_t$, and $task_t$ and $task_{t+1}$

respectively. l_u and l_b are the upper and lower margins for the thresholds.

Summarizing the PSI algorithm The following actions (Algorithm 1) are performed by any agent i that receives information from another agent j:

Algorithm 1 summarized algorithm

update d^{lower} and d^{higher} using Eq. 3.
update d^{lower} and d^{higher} using Eq. 1 and Eq. 2.
update x using Eq. 4.
update the assigned task using Eq. 5.

In the experiments reported here we applied a heuristic to regulate κ based on the number of interactions for every agent. The value of κ is updated for an agent every time it interacts with others (every time it receives information from another agents):

$$\begin{aligned} \kappa &= \kappa_b/(T+1) \\ T &= (1-\alpha) \times T + \alpha * T_{last} \end{aligned} \tag{6}$$

where $\alpha = 0.1$, and T_{last} is the time (number of time-steps) since the last interaction. κ_b is a constant value.

4 Is the PSI algorithm usable for division of labor in real-world robots?

A swarm of "Lily" underwater AUV robots (Figure 3) [9] developed in the project CoCoRo [7] was used for the first tests of PSI algorithm in a real world platform in presence of noise. The task was to distributively split the swarm into two task-groups with equal number of robots. The two groups stayed in different depth levels forming two separate layers of robots.

The maximum number of robots available for the experiment was eight. The robots have a size of about $15cm$. They use Radio Frequency (RF) for local communication under water. The RF range is about $50cm$ transmitting integer values between 0 and 512. The system is prone to noise. The robots were physically confined in an area with the surface size of $1 \times 2m^2$, and depth of $1m$. The depth of the two layers are $15cm$ and $35cm$ and the cycle length of the robots are $150msec$.

The algorithmic settings are presented in Table 1. All the robots were initialized with an identical physiological-age (x). The operational loop of a robot is summarized in Algorithm 2.

In order to test the adaptivity of the swarm to potential changes in the workforce (number of available robots), we changed the number of robots during the experiment and demonstrated the adaptability of the

Figure 3: A snapshot of the experiment with the robots. The swarm is divided into two layers. The robots in the top layer turn on their blue-lights for illustration purposes

Algorithm 2 robots' operational loop

```
loop
    broadcast physiological age (x)
    if data available from another robot * then
        run PSI
    end if
end loop
```

**at most, one set of data can be accepted in every cycle due to limitations in the robot's operationability.*

system to the changes.

Figure 4 shows the desired and actual number of robots in the course of the experiment. In the beginning, four robots are put into the water one after the other every 15 sec approximately. The 15 sec is the time that the experimenter needed in practice to pick a robot, starts it, let its pumps out since they are AUV robots and put them in the water. The experimenter then keeps the swarm size unchanged for about one minute (67 sec) and then starts adding more robots to the water until all the eight robots are inside. The swarm

size stays unchanged for about 400 sec. In second 600, all the four robots from the top layer are removed. Removing the robots takes about 30 sec which is the time the experimenter needs to catch the robots from the water and turn them off. The swarm reacts quickly to the change by rearranging the robots in the two layers such that each layer contains two robots. The swarm size is kept unchanged for about 200 sec and then the two robots on the top layer are removed from the water. The swarm reacts by rearranging the robots in the two layers such that each layer contains one robot. In second 914, the experimenter returns all the six robots to the water and the swarm reacts by rearranging the robots in the layers such that each layer contains four robots. After 173 sec, four robots on the top layer are removed from the water again and the swarm size is kept unchanged until the end of the experiment.

Figure 4a shows the instant desired number of robots in the top layer and the actual number in every observation step (every 1 sec.). Figure 4b represents the histogram of the error, which is the difference between the expected number and the actual number of robots in the layers during the observation period (20 min.). The figure indicates no error for 73% of the run-time and the maximum error of one robot in a wrong layer for the rest.

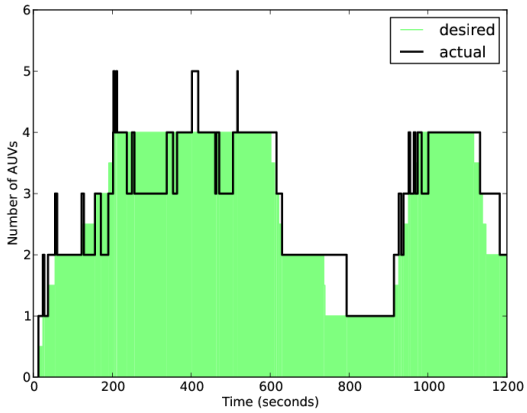
Table 1: Experimental settings of the algorithm in the robotic scenario

Parameter	value
x_{min}	0
x_{max}	512
δ	3
κ_b	6
l_u, l_b	6
initial x	195
initial d_{lower}, d_{higher}	10

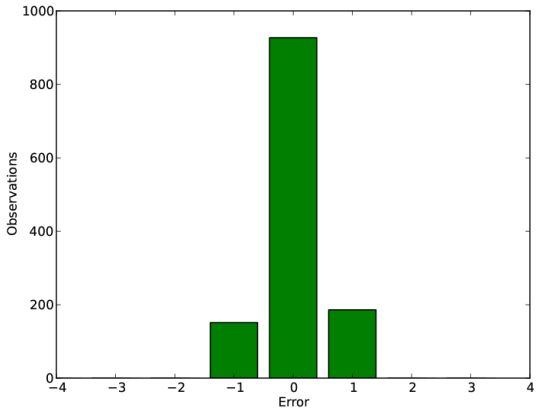
Due to the limitation in the number of available robots and their capability in reporting their internal variables, in the following subsections, the robotic scenario is simulated. The simulated scenario is repeated for several times with the same and higher number of robots and the internal variables are studied.

4.1 Simulated robotic scenario

The robotic scenario is simulated here by running the algorithm on eight robots distributed equally in two layers. The robots are added and removed from the swarm in the same order and timing as the real robot experiment. The robots are simulated with a probability of operational freeze. By the operational freeze we mean the state that the robot stops any action or reaction due to temporary hardware/bus problems. That



(a) The instant desired number of robots (filled light green) in layer 1 and the actual number (black line) in every observation step (1 second) for equal demand for the two layers. The desired number changes due to removing/adding robots from/to the swarm.



(b) The histogram of the amount of error collected during the observation time (20 minutes) representing that most of the time (73%) the number of robots in the layers are equal to the desired value ($error = 0$) and the maximum error is one robot in a wrong layer.

Figure 4: The results of equally partitioning a swarm of AUV robots. By adding new robots into the water or removing robots, the experimenter gradually increases the swarm size to eight, then gradually decreases to two, and again increases back to eight and decreases to four where the experiment ends.

is the case in our AUV robots when RF communication system was running. The probability of entering a freezing state is set to 0.05% and the probability of exiting the state is 0.5% in every cycle.

Figure 5a represents an example run of the simulated robotic swarm. Figure 5b shows the results for 25 independent runs with the simulated robotic swarm.

Since the number of our available robots for was limited to eight in reality, we simulate a robotic scenario with 16 robots in order to get an impression of the performance of the algorithm with higher number of robots. The parameter settings of the algorithm stayed the same as in the eight robot case (Table 1). Figure 6a represents the behavior of the swarm in an example run. Figure 6b represents the results for 25 independent runs of the system. The results show that the algorithm still performs well without tuning the parameters of the algorithm for this higher number of robots.

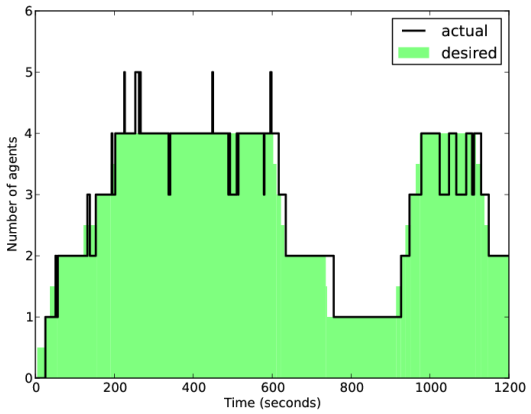
4.2 A closer look at the dynamics

In order to get a deeper understanding of the algorithm, we depicted the dynamics of the internal variables of the agents in the swarm during runtime. For that, we accumulated the values of x and d variables from all the agents in the swarm from 25 independent runs of the simulated robotic scenario with eight robots.

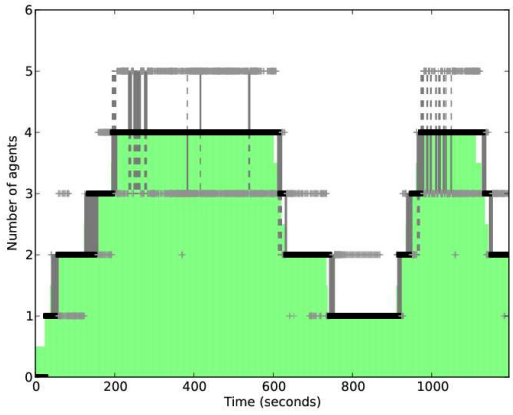
Figure 7a depicts the x values of the robots over time. The darker a patch is, the more data points are in that patch. The figure shows how the distribution of the x values changes by changing the number of agents in the swarm.

As it is represented in Figure 7a, the distance between the x values in each time-step, tends to be uniform over the whole experiment independent of the number of agents. When the number of agents is lower, the uniformity is more precise. When the number of agents is higher, there are more fluctuations in the x values and the distribution of the distances is less uniform. Nevertheless, success of the algorithm depends also on the range of the x values, the thresholds (number of subgroups) that split the swarm into tasks, and the parameters of the algorithm (section 4.3 investigates the parameter settings). In the experiment here, the agents are supposed to split into two tasks with equal demands meaning that a threshold value is set at the middle of the x-range and the robots with an x at each side of the threshold pick the corresponding task to perform. When the number of robots is small, the differentiation between the tasks are very clear for all the robots. When the number gets bigger, the differentiation becomes less clear for the robots with x values around the middle of the range where the threshold locates (note that due to the limitations in the communication bandwidth of the robots in our experiment, x is an integer value in the range of $[0, 512]$).

Figure 7b depicts the values of d_{lower} over time for all the robots in 25 runs. The diagram is similar for

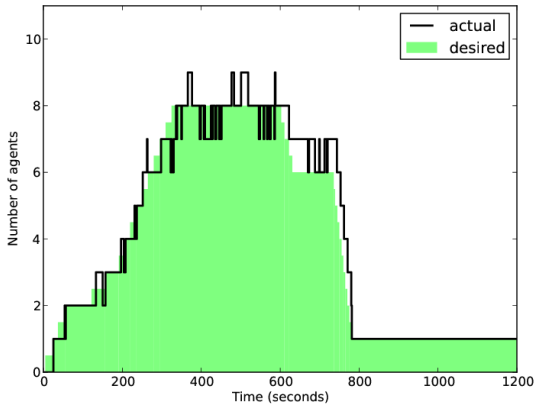


(a) An example run of the simulated robotic swarm. The instant desired number of agents in layer 1 is depicted in light green (filled) and the actual number of robots is depicted by black lines. The desired number changes due to removing/adding agent from/to the swarm.

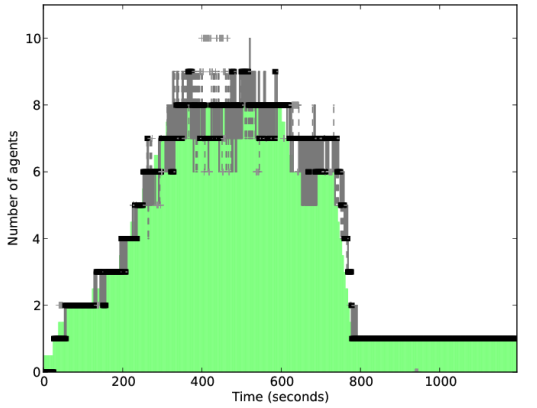


(b) The instant desired number of agents (filled light green) in layer 1 and the actual number (black) in every observation step for equal demand for the two layers. The desired number changes due to removing/adding agent from/to the swarm. The results are pulled from 25 independent runs. Box-plots (bold gray) indicate median and quartiles. + signs indicate outliers.

Figure 5: The simulation results of equally partitioning a swarm of agents into two layers (similar setup as the experiment with the real AUV swarm).



(a) An example run of the simulated robotic swarm. The instant desired number of agents in layer 1 is depicted in light green (filled) and the actual number of robots is depicted in back lines. The desired number changes due to removing/adding agent from/to the swarm.



(b) The instant desired number of agents (filled light green) in layer 1 and the actual number (black) in every observation step for equal demand for the two layers. The desired number changes due to removing/adding agent from/to the swarm. The results are pulled from 25 independent runs. Box-plots (bold gray) indicate median and quartiles. + signs indicate outliers.

Figure 6: The simulation results of equally partitioning a swarm of 16 robots into two layers.

d_{higher} and therefore is not shown. In the case of a change in the swarm size, the distance between the x values changes and consequently the expected value of d which is the local estimation of the agent for the distance also changes.

4.3 Investigating parameter settings

In Figure 7a, a number of local optima of the distribution of x in every time-step is detectable. These local optima are the expected values for the x according to the swarm size. The separation around optima is very sharp for low number of robots and is less clear when the number of robots is eight. In the case of eight robots, eight bands of distribution around optima are still detectable but the separation is very fuzzy. The degree of fuzziness and the speed of changes can be dependent on the parameters of the algorithm such as σ and κ.

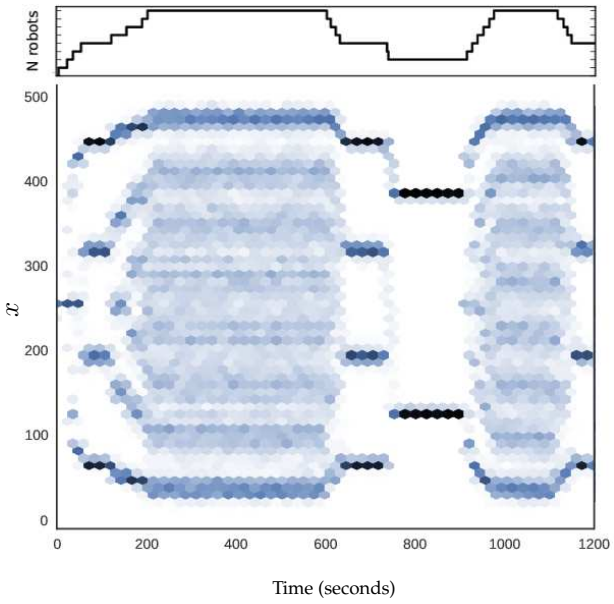
In order to investigate the effect of different parameterization in the performance of the algorithm, we run several experiments with a swarm of size eight for different values of σ and κ. Each experiment is repeated for 25 independent runs. An experiment starts with eight agents and keeps the swarm size fixed for the whole period (other parameters identical to the previous case). Figure 8 shows three example settings of σ and κ. As it can be seen in the figure, the separation of the distribution bands around the eight optima gets less clear as the κ increases. Another interesting feature to note is the difference between the speed of the spreading of x values along the range. In the figure, the dispersion is faster when the value of κ is larger.

We introduce two measures called *dispersion* and *convergence-time* that reflect the two above mentioned features.

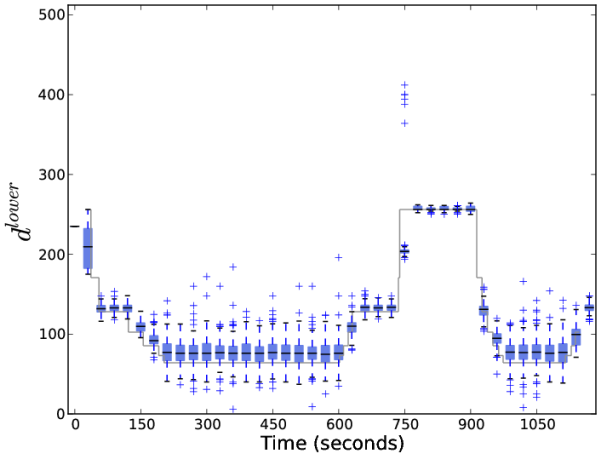
In order to compute *dispersion*, we split the range of x into eight equal sub-ranges. The interquartile range (IQR[1]) of the data distributed in every sub-range is computed and the maximum of the eight values is used as the *dispersion* value (see Figure 8a-c).

In order to compute *convergence-time*, the number of agents with their x values in each of the eight subranges is counted. The difference between the maximum and the minimum of the values are computed for every time step. That is in fact the difference between the number of x values locating in the most crowded subrange and the ones in the emptiest subrange. At the beginning, since all the agents start with an identical initial x, the difference is eight. The difference decreases over time when the x values are spreading over the range. *convergence-time* is defined as the time to reach to 0.5 as the difference averaged over 25 runs (see Figure 8d).

[1]IQR is the difference between the upper and lower quartiles of the data $IQR = Q_3 - Q_1$



(a) The bottom graph represents the x values of the robots over time accumulated from 25 runs. The darker a patch is, the more data points are in that patch. The top graph represents the number of robots in the swarm over time.



(b) The value of d_{lower} over time for all the robots collected from 25 runs. Box-plots indicate median and quartiles, whiskers indicate the most extreme data points which are no more than 1.5 times the length of the box away from the box, $+$ signs indicate outliers. the solid gray line indicates the expected value of d_{lower} in the case of a perfect distribution of x values.

Figure 7: Internal dynamics of the variables over time in simulated robotic scenario.

Both features, *dispersion* and *convergence-time*, are computed for several different values of σ and κ. Figure 9 represents the values of the features. In the figure, the darker a cell of the grid is, the higher is the feature value. In Figure 9a, brighter cells of the grid represent lower dispersion meaning a better separation between the x values of agents. The brighter cells of the *convergence-time* in Figure 9b represent faster convergence. A good choice of the parameter setup is a trade off between the two features and depends on the practical requirements and constraints of the given task.

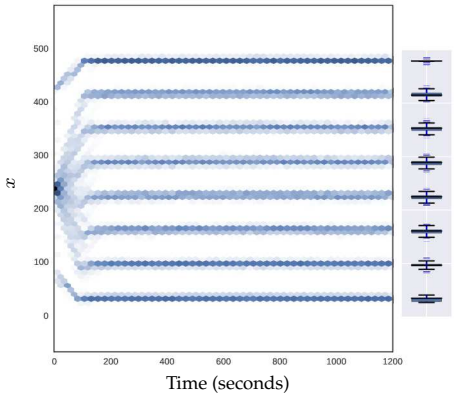
In addition to *dispersion* and *convergence-time*, the number of switchings between the tasks in the swarm is also interesting in practice. Figure 9c shows the total number of switchings between the tasks in the swarm over the period of 400 sec to 1200 sec for different values of σ and κ. The represented values are the median of the 25 runs. The brighter a cell is, the lower is the number of switchings over the period.

5 Is the behaviour of PSI comparable with the behaviour of real honeybee colonies?

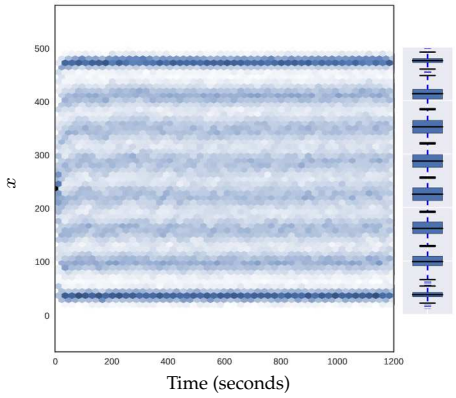
With a turn back towards the inspiration origins of PSI algorithm, here we report the results of experiments with honeybees (*Apis mellifera*). Along with that, the results of experiments with simulated swarm of agents running PSI algorithm are reported. The changes in conditions in the simulated swarm are set such that they emulate the changes induced in our real honeybee colonies. The behaviour of the simulated swarm and the honeybee colonies are then compared representing similar effects in both cases.

5.1 Experimenting with real honeybees

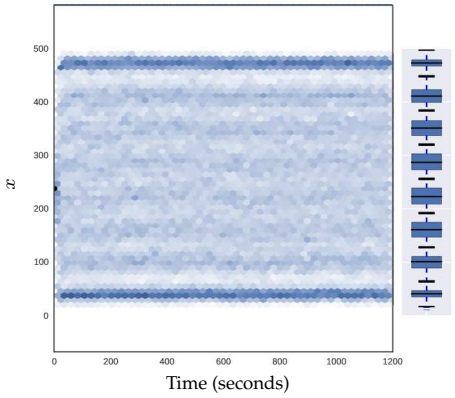
In a set of experiments with honeybees, we tested whether the rate of behavioral development is changed by a human-induced swarming of honeybee colonies. By *swarming* we mean the process of forming a new colony when a queen bee and a large group of worker bees leave their colony to establish a new colony in a new place. In the experiments here, we induced a swarm-like event by collecting bees and putting them in a fresh empty hive (as explained in the following). Swarming event is a big demographic change that a colony faces in a normal annual cycle [28]. The bees in this situation are confronted with a lot of difficulties, e.g., they must build combs and have no food stores and no very young workers who can take over some of the work. Since there are no freshly emerging bees for at least three weeks after the swarming event, all the tasks in the colony after swarming have to be done by the decreasing number of aging workers.



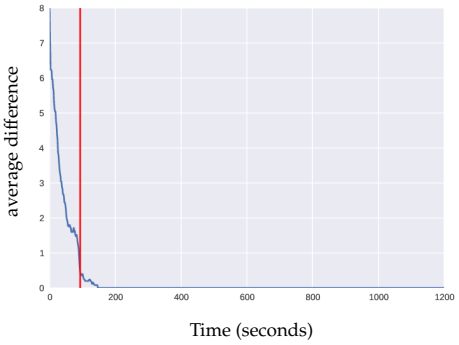
(a) $\delta = 3$ and $\kappa = 1$, *dispersion*$= 6$



(b) $\delta = 3$ and $\kappa = 6$, *dispersion*$= 24$

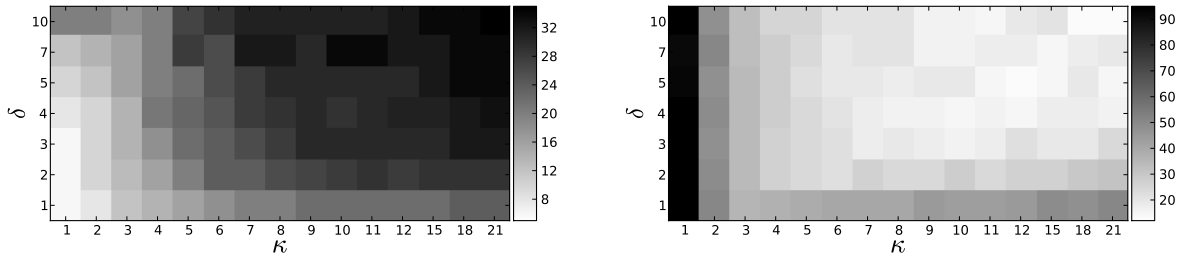


(c) $\delta = 3$ and $\kappa = 10$, *dispersion*$= 30$

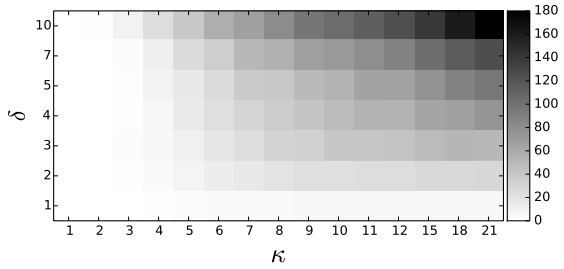


(d) $\delta = 3$ and $\kappa = 1$, *convergence time*$= 96sec$

Figure 8: a-c) Distribution of x values over time for a swarm of eight agents for different parameter sets (δ, κ). The agents start with identical x value. Data is accumulated from 25 independent runs. The box-plots represent median and quartiles for each sub-range. The *dispersion* value is the maximum of all the inter quartile ranges ($IQR = Q_3 - Q_1$). d) The difference between the number of x values located in the most crowded and the emptiest sub-range for 25 runs. The vertical red line indicates the *convergence-time* for this setup.



(a) *dispersion* (b) *convergence-time*



(c) *total number of switchings*

Figure 9: *dispersion*, *convergence-time*, and *total number of switchings* for different values of δ and κ. The lower the value is (less dispersed, faster convergence, less switchings), the brighter it is depicted in the grid.

The behavioral development of worker honeybees is associated with a number of morphological and physiological changes [10, 39, 26]. It has been demonstrated in previous studies [24] that the lack of young workers extends brood rearing and comb construction to older chronological ages. In addition, the age of workers in the colonies seem to shift upward until the first new bees emerge which represents an adaptation in longevity of the honeybees [40].

Table 2: Experimental settings in the bee experiments

	colony1	colony 2	colony 3
number of bees marked per day	203	213	213
number of days of marking	42	42	43
marking started	18 April	21 June	15 April (2002)
marking stopped	30 May	2 August	28 May (2002)
swarm-event	5 June	7 August	1 June (2003)

5.1.1 Experimental setup

For the experiments with the honeybees, three trials were performed with three separate colonies of *Apis mellifera carnica* (see Table 2). The colonies were housed indoors in an 8-frame observation hive on the

second floor in a building in the university (47° 04′ North, 15° 27′ East). The worker bees were given access to the outdoors via a transparent tube. All colonies were containing bees of all age cohorts and a mated queen. The brood combs were taken from different colonies, as it is almost impossible to get a sufficient number of young bees from only one colony over an extended period of several weeks. Every day we marked a number of 203-213 freshly emerged bees. To obtain enough different age classes, the bees were marked over a period of at least 42 days. The workers were defined as being 1-day-old at the day of marking, if their age is between 0 to 24 hours.

Since no freshly emerged bees are found in colonies after natural swarming event [6], we stopped marking the bees a few days (5-7 days) before swarming to exclude very young bees at the time of the swarm-like event. Workers that are marked on the same day are defined as one age-cohort. A few days before the swarming event we trained the bees to a feeder provided with sugar water near the hive to guarantee a constant food supply for each colony before and after the induced swarm-like event. As a control phase, observation started 4 days before inducing the swarm-like event. To induce the event, all bees were shaken from the old combs, then they were removed and replaced by empty wax foundation. The queen was caged inside the hive to prevent her loss, and the opening of the cage was sealed with sugar dough. The hive was closed when all bees were situated inside the observation hive (the place for establishing the new colony).

Two persons recorded the numbers and colors of marked workers showing nursing and foraging behaviour to analyze age demography of the respective group. Both observers collected data on each group for approximately one hour, trying to get similar sample sizes. This was done either daily or every second day. Nurses were defined as bees putting their heads for more than 3 seconds into unsealed brood cells containing larvae. Foragers were defined as bees coming in with pollen on their legs or those not carrying pollen were assumed to be carrying nectar if they showed waggling dances or round dances. Dance followers, trembling dancers and bees at the feeder were also defined as foragers.

The age demography of the nurses and foragers are depicted in Figure 10 (a-c) and Figure 11 (a-c) for the three investigated colonies.

5.2 Experimenting with simulated swarm running PSI

Swarms of simulated agents running PSI algorithm were investigated for their behaviour in response to the changes similar to the situation in the bee colonies with the induced swarming event. Four tasks, namely cleaning, nursing, food-storing, and foraging, were introduced and the algorithm was supposed to partition the swarm into the four tasks. The random interactions between the agents were implemented as local and

reciprocal exchange of physiological age (x values) between the agents in the same task or neighbouring tasks.

A simulated swarm starts with 50 agents with a fixed regime of birth and death (Table 3) and continued for 30,000 simulation steps when the swarm size was stabilized at about 2200 agents. At that step, the simulated procedure of induced swarming event started. All the agents which were presented up to this point, were considered as marked agents. Only the marked agents were observed in the next period of simulation. Every 200 simulation steps were considered a day for the agents. For a period of 4 days (control phase), the colony continued with the birth of new agents, and then the induced swarming event occurred. For a period of 5 days, no new agent was born emulating the situation of no larvae in the hive after swarming event. After 26 days (five days for no larvae plus three more weeks for brood), new agents started to emerge. At day 30 observation stopped. The data was collected from 10 independent simulated swarms. Figure 10d and Figure 11d depict the observation results in the same way as the real bee colonies.

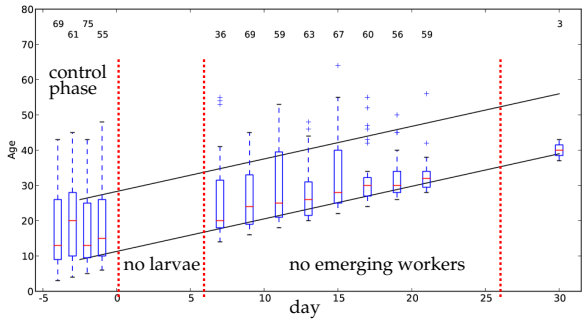
Table 3: Experimental settings in the simulated swarm scenario

birth rate	50 per day (100 attempts with probability of 0.5)
death rate	$(1 + exp(-0.15 * (age/200 - 60)))^{-1}$ or $age > 80$
ticks per day	200
x_{min}	0
x_{max}	16000
l_u, l_b	1
δ	3
κ_b	6
segment size of $Cleaning$	$0.5 \times (x_{max} - x_{min})$
segment size of $Nursing$	$2 \times (x_{max} - x_{min})$
segment size of $Food - storing$	$2 \times (x_{max} - x_{min})$
segment size of $Foraging$	$8 \times (x_{max} - x_{min})$

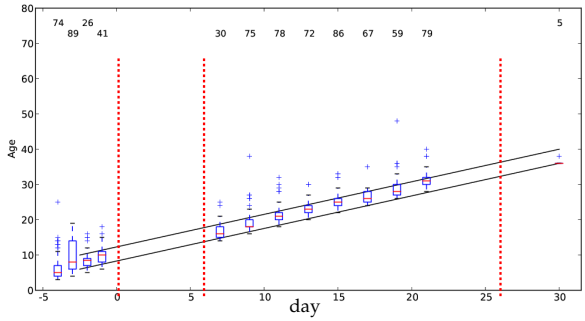
5.3 Results from honeybees and simulated swarms

Age demographics of the nurses and foragers are presented as Box-and-Whisker-Plots in Figure 10(a-c) and Figure 11(a-c). The two inclined lines emanating from the control phase on day -2.5, depict a prognosis. The lines depict the age of the bees that during the control phase perform the respective tasks (nursing or foraging). Since a bee gets older every day, the inclined lines increase linearly (one day every day). In order to get the starting points of the inclined lines, the respective quartiles of the four control days were averaged.

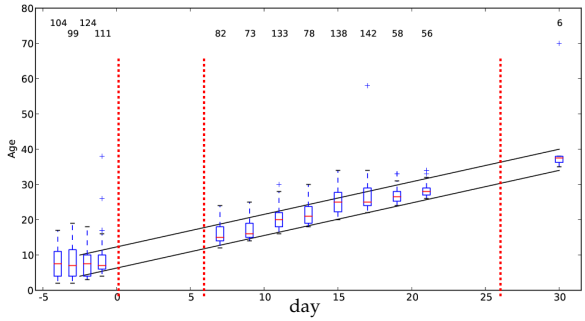
Recall that in a normal situation of a colony, bees follow a developmental behaviour and switch from a task



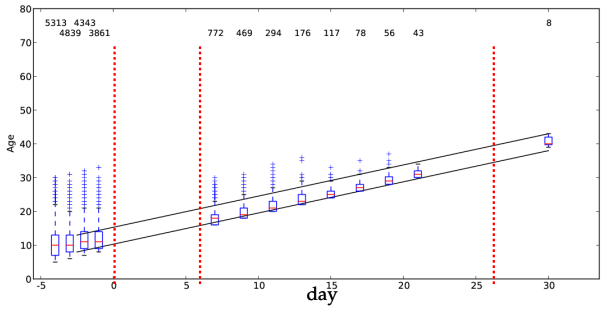
(a) Colony 1



(b) Colony 2

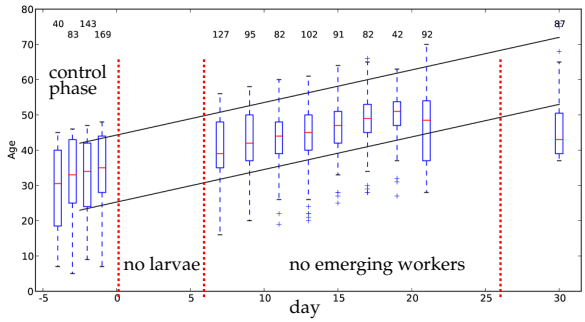


(c) Colony 3

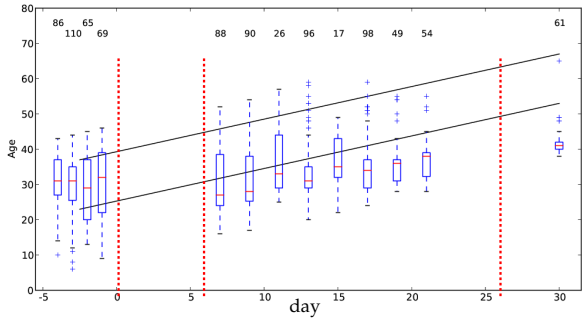


(d) Simulated swarm

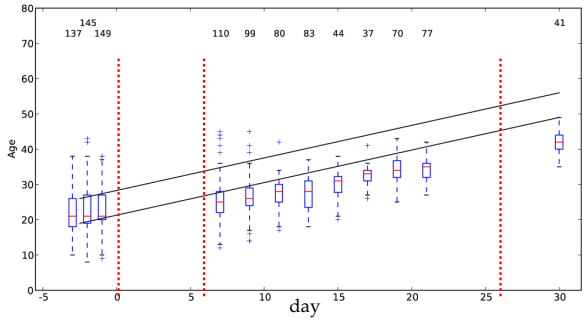
Figure 10: Age demographics of the nurses during the observation period for real bee colonies (a-c) and the simulated swarm (d). The two parallel inclined lines indicate the chronological age of the bees presented in this task in the control phase. If the same bees stay in this task, the age-values of the observed bees during the experiment should stay between these two lines. Box-plots indicate median and quartiles, whiskers indicate the most extreme data points which are no more than 1.5 times the length of the box away from the box, + signs indicate outliers. The numbers at the top of the figures represent the number of bees observed in that day (experimental observation in real bee and the actual number of bees in the simulated swarm)



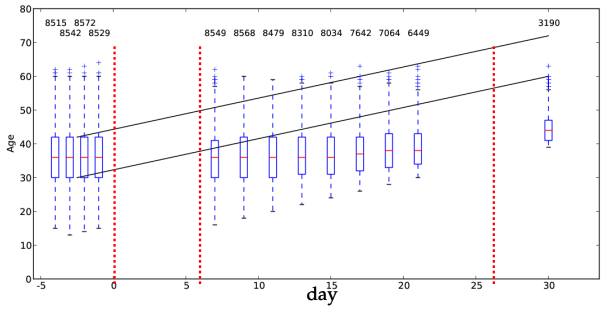
(a) Colony 1



(b) Colony 2



(c) Colony 3



(d) Simulated swarm

Figure 11: Age demographics of the foragers during the observation period for real bee colonies (a-c) and the simulated swarm (d). The two parallel inclined lines indicate the chronological age of the bees presented in this task in the control phase. If the same bees stay in this task, the age-values of the observed bees during the experiment should stay between these two lines. Box-plots indicate median and quartiles, whiskers indicate the most extreme data points which are no more than 1.5 times the length of the box away from the box, + signs indicate outliers. The numbers at the top of the figures represent the number of bees observed in that day (experimental observation in real bee and the actual number of bees in the simulated swarm)

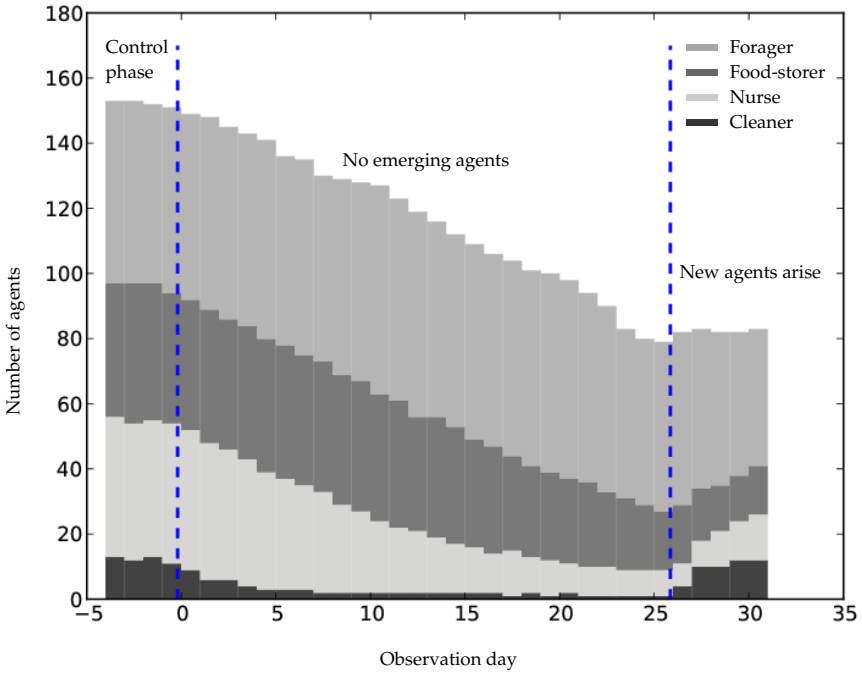


Figure 12: A typical simulated swarm with the swarming event

to the next one when they get old enough. Since the control phase (which is before the swarm-like event) represents the normal situation of a colony, the age cohorts of the bees in the control phase, represent the normal age cohorts for the tasks in that colony. This means, if the behavioral developments of the bees stay intact, the age cohorts for the task do not change over the time of observation. Nevertheless, we induced the swarm-like event which drastically disturbs the normal situation of the colony.

If the behavioral development of a colony is disturbed in a way that the same cohorts of bees are engaged in a task for long periods of observation, the age distribution for the task stays between the inclined lines. Figure 10 (a-c) shows the age demography of nurses. As the figure shows, after the swarm-like event, the ages of the observed nurses in the three colonies stay between the inclined lines. That means the age cohorts of nurses after the swarm event remain almost the same as the age cohorts of the nurses originated in the control period. On the other hand, since no young bees hatched in the newly settled colonies up to at least three weeks after swarming, there is no freshly emerged bee (unmarked bee) that can take over the nursing task in this period. Based on these two facts, we suspect that the same bees that were performing the nursing during the control phase are the nurses observed after the swarm-like event.

As represented in Figure 10 (a-c), in colony 1 the distance between the two inclined lines is bigger than the distance in the other two colonies, meaning that the nurses in the control phase of this colony were from a

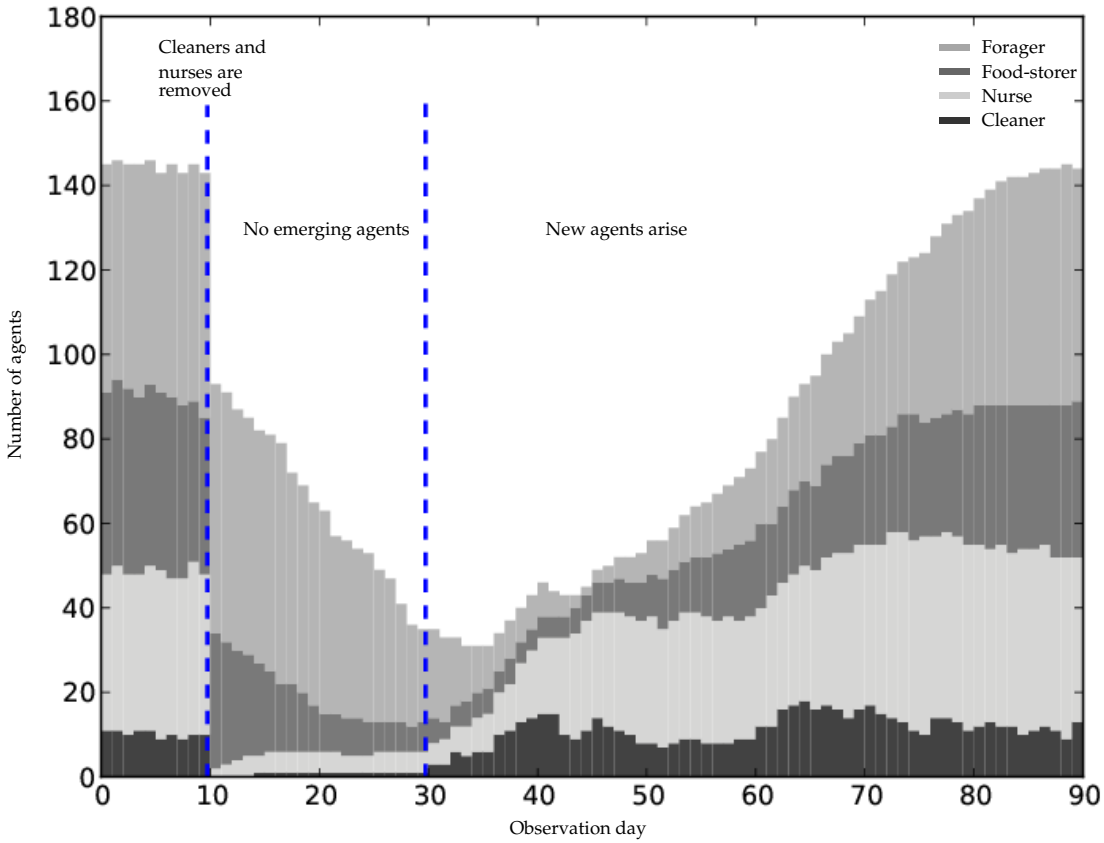


Figure 13: A typical simulated swarm where all the nurses and cleaners are suddenly removed from the colony at day 10 and no new bees are emerged until day 30. Immediately after that, some of the bees switch from food-storing to nursing and at day 15, at least one bee performs cleaning. It shows a reverse change in the behavioral development which is comparable with real bees. At about day 20, the relative number of bees undertaking the four tasks stays almost unchanged even though the swarm size keeps decreasing due to the death rate which hits older bees the most, meaning the adaptivity of the colony in keeping the balance of the task performances. At day 30, new bees emerge with a fixed rate. The swarm size increases gradually while the relative number of bees stays similar with some fluctuations.

broader range up to higher ages.

Figure 11 (a-c) shows the number of detected foragers. The median age of the foragers is significantly higher than the median age of the nurses in all the phases until the end of the experiments (p¡0.05, Mann-Whitney U-test). Until the day when the first new bees emerged, some of the oldest former foragers could not be observed at foraging any more, and had most probably died.

The inclined lines in Figure 11 represent the age cohorts of the foragers which were present in the control phase. The deviations from the lines represent a tendency against aging in the detected foragers after the swarm-like event. This is more clear in colony 2 and 3. A reason for this decrease in the age (in comparison with the inclined lines) is due to the death of the old bees as well as the presence of many bees in the previous tasks that might be able to take over the foraging when the older foragers die.

Figure 10d and Figure 11d depict the results from the simulated swarms representing behaviours similar to the real bee colonies for both nurses and foragers. This similarity suggests that the mechanisms defined in the PSI algorithm can be comparable with the mechanisms supporting the behaviours of real bee colonies.

5.4 Potential insights from the simulated swarm

Unlike the real bee experiments where we only have access to the data recorded for the detected bees, here we can look into more details of the behaviour of the swarm driven by the PSI algorithm. Figure 12 represents the number of agents in every simulated day of the experiment. As expected, the figure indicates a decrease in the swarm size causing by the death in absence of birth until day 26[2] when the new bees emerge. Note that the death rate is higher for the older agents (Table 3) and the birth rate (starting from day 26) adds new agents to the cleaning task. The balance between the relative number of agents in different tasks is noticeable. It represents the effectiveness of the system in keeping the relative numbers fixed although the swarm size changes by removing from one side and adding to the other side of the task sequence.

Figure 13 represents the behaviour of a typical simulated swarm where all the agents undertaking the cleaning and nursing and also the brood are removed at a point in time and no new agents emerge. As indicated in the figure, number of agents reverse their behavioral development (get physiologically younger) and undertake the cleaning and nursing. In addition, the relative numbers of agents in each task tend to stay fixed while some fluctuations still exist. These effects indicate adaptivity of the simulated swarms running the PSI algorithm and is comparable with the behaviour of colonies of real bees reported in the literature (e.g. [15]).

[2] 5 days of no larvae and eggs plus 21 days for the new bees to emerge from new eggs

6 Conclusion

In this paper, a distributed algorithm for adaptive partitioning for division of labor inspired from age-polyethism in honeybees is investigated. The algorithm is called Partitioning Social Inhibition (PSI) and employs the concept of physiological age of agents (represented by an internal variable x). By distributing the x value of the agents over the range and defining a number of thresholds (th) relative to the task demands, the swarm is split into corresponding task groups.

The paper is twofold: first robotic application of PSI algorithm and analysis of the parameters, and similarity of behaviour in honeybee colonies and simulated swarms running PSI algorithm.

We successfully implemented the algorithm for adaptive partitioning of a swarm of underwater robots in presence of noise. Figure 4 demonstrates the high performance of the algorithm for partitioning a robot swarm into two tasks-groups with adaptability to changes in the swarm. We then implemented the algorithm in a simulation of the robotic experiment with the same (Figure 5) and with a bigger number of robots (Figure 6) in the swarm. The simulation is then used to get insights of internal dynamics of the variables during run-time (Figure 7). Moreover, two features indicating the quality of dispersion and the convergence time of the x values are defined and different parameter sets of the algorithm are investigated for their influence in the features (Figure 9). The investigation indicates that a good choice of the parameter setup is a trade off between the two features depending on the practical requirements and constraints of the given problem.

In addition to the robotic application, we also investigated the algorithm in a simulated swarm and compared its behaviour to a set of experiments that we carried out with honeybees. For that, three colonies of honeybees undergoing a human-induced swarming event were studied and the age demographics of the nurse bees and forager bees were depicted. Figure 10 and Figure 11 represent the age demographics for the three investigated honeybee colonies and simulated swarms. The results from both the real bees and the simulated agents represent that the behavioral development slows down after the swarm-like event by prolongation of the phase of nursing. The similarity of the effects suggests that the mechanisms employed in the PSI algorithm might be comparable with the mechanisms supporting the behaviour in the real bee colonies.

The PSI algorithm has some similarities but also some basic differences with other algorithms in the field, such as foraging-for-work (FFW) [36]. For example, in FFW, tasks are spatially arranged in a sequence and agents wander around and seek a task to perform. The spatial presence and physical movement of

the agents in the task areas lead to the distribution of the agents relative to the task demands. On the other hand, in PSI, it is the physiological age (x) of the agents that is virtually distributed relative to the task demands and leads to physical distribution of the agents in different areas. From a biological point of view, FFW does not consider physiological differences between bees to drive them to take different tasks to perform. On the other hand in PSI, a physiological age (x) is defined for the agents. This physiological age is considered a representative of physiological and morphological characteristics of the agents enabling it to perform particular tasks. There are also other methods such as stochastic policies as discussed in [1] which do not assume sequential ordering of tasks but uses a directed graph that defines interconnection topology of possible switchings between the tasks. Unlike PSI that aims for low switchings between the tasks, the stochastic policy method works based on persistent switching between the tasks while the transition rates are computed according to the demand for different tasks and determine the probability of the transitions between the tasks. PSI algorithm on the other hand aims not only for maintaining the desired ratio of the agents (robots) according to the demands but also keeps the switching rate low since switching between different tasks may cost time and energy and prevents specialization of the agents in the tasks. Another difference is that in stochastic policy method the transition rates are subject to optimization techniques. In PSI there is no optimization technique used since the thresholds are set directly based on the demands such that every segment between two adjacent thresholds is proportional to the relative demand of the related task.

In the current implementation of the PSI algorithm, the thresholds are set based on the relative demands for the tasks. If the demands change, the thresholds have to be set to the new values for all the agents. In order to get an immediate rearrangement of the agents, the new thresholds can be broadcasted to all the agents as global information. Otherwise the speed of the reaction would depend on the implementation of the propagation of the information about the new demands (which leads to new values for the thresholds). In the future, we will improve the algorithm such that there is no need to any central calculation of the thresholds. We will also use the PSI algorithm in other real applications and more complicated scenarios. Moreover, potential predictions that can be generated by using the algorithm in simulated swarms (e.g., in Figure 13) can be validated by performing experiments with honeybee colonies.

7 Acknowledgements

This work was supported by: EU-ICT project 'CoCoRo', no. 270382; EU-H2020 project 'subCULTron', no. 640967. Some of the graphics are generated by Seaborn package (http://dx.doi.org/10.5281/zenodo.12710).

References

[1] Spring Berman, Adam Halasz, M Ani Hsieh, and Vijay Kumar. Optimized stochastic policies for task allocation in swarms of robots. *Robotics, IEEE Transactions on*, 25(4):927–937, 2009.

[2] S.N. Beshers. Social inhibition and the regulation of temporal polyethism in honey bees. *Journal of Theoretical Biology*, 213(3):461–479, 2001.

[3] S.N. Beshers and J.H. Fewell. Models of division of labor in social insects. *Annu. Rev. Entomol*, 46:413–440, 2001.

[4] S.N. Beshers, G Robinson, and J Mittenthal. Response thresholds and division of labor in social insects. In Claire Detrain, JeanLouis Deneubourg, and JacquesM. Pasteels, editors, *Information Processing in Social Insects*, pages 115–139. Birkhauser Basel, 1999.

[5] Eric Bonabeau, Andrej Sobkowski, Guy Theraulaz, and Jean-Louis Deneubourg. Adaptive task allocation inspired by a model of division of labor in social insects. In *Biocomputing and emergent computation: Proceedings of BCEC97*, pages 36–45. World Scientific Press, 1997.

[6] C.G. Butler. The ages of bees in a swarm. *Bee World*, 21:9–10, 1979.

[7] CoCoRo. Project website, 2013. http://cocoro.uni-graz.at.

[8] M. W. J. Crosland, S. X. Ren, and J. F. A. Traniello. Division of labour among workers in the termite, reticulitermes fukienensis (isoptera: Rhinotermitidae). *Ethology*, 104:57–67, 2010.

[9] Tobias Dipper. Einsatz von elektrischen Feldern zur Lokalisierung von autonomen Unterwasserschwarmrobotern. 2015. in: BV-Forschungsberichte.

[10] J. B. Free and Y. Spencer-Booth. The longevity of worker honeybees. *Proceedings of the Royal Entomological Society of London. Series A, General Entomology*, 34(10-12):141–150, 1959.

[11] Raghavendra Gadagkar. Division of labour and organization of work in the primitively eusocial wasp ropalidia marginata. In *Proceedings of the Indian National Science Academy - Part B: Biological Sciences*, volume 6 of *67*, pages 397–42, 2001.

[12] Roderich Gross, Shervin Nouyan, Michael Bonani, Francesco Mondada, and Marco Dorigo. Division of labour in self-organised groups. In *SAB*, pages 426–436, 2008.

[13] Bert Hölldobler and E.O. Wilson. *The Superorganism: The Beauty, Elegance, and Strangeness of Insect Societies*. W. W. Norton and Company, New York, 2008.

[14] Z Y Huang and G E Robinson. Honeybee colony integration: worker-worker interactions mediate hormonally regulated plasticity in division of labor. *Proc Natl Acad Sci U S A*, 89(24):11726–9, 1992.

[15] Z Y Huang and G E Robinson. Regulation of honey bee division of labor by colony age demography. *Behavioral Ecology and Sociobiology*, 39(3):147–158, 1996.

[16] Zhi-Yong Huang and GeneE. Robinson. Social control of division of labor in honey bee colonies. In Claire Detrain, JeanLouis Deneubourg, and JacquesM. Pasteels, editors, *Information Processing in Social Insects*, pages 165–186. Birkhauser Basel, 1999.

[17] Free JB. The allocation of duties among worker honeybees. In *Symp Zool Soc Lond*, volume 14, pages 39–59, 1965.

[18] Brian Johnson. Division of labor in honeybees: form, function, and proximate mechanisms. *Behavioral Ecology and Sociobiology*, 64(3):305–316, January 2010.

[19] Chris Jones and Maja J. Mataric. Adaptive division of labor in large-scale minimalist multi-robot systems. In *IROS*, pages 1969–1974. IEEE, 2003.

[20] G. E. Julian and Sara Cahan. Undertaking specialization in the desert leaf-cutter ant acromyrmex versicolor. *Anim Behav*, 58(2):437–442, 1999.

[21] N. Hrassnigg K. Crailsheim and A. Stabentheiner. Diurnal behavioural differences in forager and nurse honey bees (apis mellifera carnica pollm). *Apidologie*, 27(4):235–244, 1996.

[22] Istvan Karsai and Thomas Schmickl. Regulation of task partitioning by a "common stomach": a model of nest construction in social wasps. *Behavioral Ecology*, 22:819–830, 2011.

[23] Thomas H. Labella, Marco Dorigo, and Jean-Louis Deneubourg. Division of labor in a group of robots inspired by ants' foraging behavior. *ACM Trans. Auton. Adapt. Syst.*, 1(1):4–25, September 2006.

[24] Martin Lindauer. *Communication Among Social Bees*. Harvard University Press, 1961.

[25] B. Moritz and K. Crailsheim. Physiology of protein digestion in the midgut of the honeybee (apis-mellifera l). *J Insect Physiol*, 33(12):923–931, 1987.

[26] K. Crailsheim N. Hrassnigg. Differences in drone and worker physiology in honeybees (apis mellifera). *Apidologie*, 36(2):255 – 277, 2005.

[27] D. Naug and R. Gadagkar. Flexible division of labor mediated by social interactions in an insect colony a simulation model. *Journal of Theoretical Biology*, 197:123–133, 1999.

[28] K. Naumann and M.L. Winston. Effects of swarm type on temporal caste polyethism in the honey bee, apis mellifera l.(hymenoptera: Apidae). *Insectes Sociaux*, 37(1):58–72, 1990.

[29] Iñaki Navarro and Fernando Matía. An introduction to swarm robotics. *ISRN Robotics*, 2013:10, 2013.

[30] Amy L Toth; Sara Kantarovich; Adam F Meisel; Gene E Robinson. Nutritional status influences socially regulated foraging ontogeny in honey bees. *The Journal of experimental biology*, 208:4641–4649, 2005.

[31] G E Robinson. Regulation of division of labor in insect societies. *Annu Rev Entomol*, 37:637–65, 1992.

[32] E. Sahin. Swarm robotics: From sources of inspiration to domains of application. In *Lecture Notes in Computer Science*, volume 3342 of *LNCS*, pages 10–20, Berlin, Germany, 2005. Springer-Verlag.

[33] Thomas Schmickl, Christoph Möslinger, and Karl Crailsheim. Collective perception in a robot swarm. In Erol Şahin, William M. Spears, and Alan F. T. Winfield, editors, *Swarm Robotics - Second SAB 2006 International Workshop*, volume 4433 of *LNCS*, Heidelberg/Berlin, Germany, 2007. Springer-Verlag.

[34] Thomas D. Seeley. Adaptive significance of the age polyethism schedule in honeybee colonies. *Behavioral Ecology and Sociobiology*, 11:287–293, 1982.

[35] Guy Theraulaz, Eric Bonabeau, and Jean-Louis Deneubourg. Response threshold reinforcement and division of labour in insect societies. In *In Proc. Royal Society of London*, volume 265 of *5*, pages 327– 332, 1998.

[36] C. Tofts. Algorithms for task allocation in ants. *Bull. Math. Biol.*, 55:891–918, 1993.

[37] VO. Torres, TS. Montagna, J. Raizer, and WF. Antonialli-Junior. Division of labor in colonies of the eusocial wasp, mischocyttarus consimilis. *Journal of insect science*, 12:21, 2012.

[38] Tony White and James Helferty. Emergent Team Formation: Applying Division of Labour Principles to Robot Soccer. *Engineering Self-Organising Systems*, pages 180–194, 2005.

[39] Edward O. Wilson. *The insect societies*. Belknap Press of Harvard University Press, 1971.

[40] Mark L. Winston. Intra-colony demography+ and reproductive rate of the africanized honeybee in south america. *Behavioral Ecology and Sociobiology*, 4(3):279–292, 1979.

[41] M.L. Winston. *The biology of the honey bee*. Harvard University Press, 1987.

[42] Yongming Yang, Changjiu Zhou, and Yantao Tian. Swarm robots task allocation based on response threshold model. In Gourab Sen Gupta and Subhas Chandra Mukhopadhyay, editors, *ICARA*, pages 171–176. IEEE, 2009.

[43] Payam Zahadat, Karl Crailsheim, and Thomas Schmickl. Social inhibition manages division of labour in artificial swarm systems. In Pietro Lio, Orazio Miglino, Giuseppe Nicosia, Stefano Nolfi, and Mario Pavone, editors, *12th European Conference on Artificial Life (ECAL 2013)*, pages 609–616. MIT Press, 2013.