

Social Inhibition Manages Division of Labour in Artificial Swarm Systems

Payam Zahadat¹, Karl Crailsheim¹ and Thomas Schmickl¹

¹Artificial Life Lab of the Department of Zoology, Universitätsplatz 2,
Karl-Franzens University Graz, 8010 Graz, Austria
payam.zahadat@uni-graz.at

Abstract

This paper presents a novel and bio-inspired algorithm for distributed division of labour in swarms of artificial agents (e.g., autonomous underwater vehicles). The algorithm is inspired by division of labour via local interactions in social insects. The algorithm is successfully implemented in virtual agents and simulated robot swarms and demonstrates a high adaptivity in response to changes in the workforce and task demands in the swarm level as well as a high specialization to tasks in the agents level.

Introduction

In the field of swarm intelligence and collective robotics, social insects are promising sources of inspiration due to their capabilities in self-organization and self-regulation in the colonies. Division of labour is one of the prominent characteristics of social insects such as honey bees (Seeley (1982); Huang and Robinson (1992)) and ants (Hölldobler and Wilson (2008); Julian and Cahan (1999)). The insect colonies maintain plasticity that adapts the division of labour to the status of the colony and, in parallel, also to environmental situations, e.g. to the number of workers or task-demands. The flexibility and quick response to the changes in the status of the colony and environment in one hand and specialization of workers for the tasks in the other hand are interesting properties of the methods driving behaviour in the colonies.

Several models have been proposed to explain the mechanisms of division of labour in social insects, e.g. foraging-for-work (Tofts (1993)), response-threshold reinforcement (Bonabeau et al. (1997); Theraulaz et al. (1998)), and common-stomach models (Karsai and Schmickl (2011)) (for a good review of such models see Beshers and Fewell (2001)). These models have been also used as inspiration for distributed algorithms in application areas such as swarm robotics. For example, in Gross et al. (2008) and Labella et al. (2006), models of ants-foraging behaviour have been implemented for maintaining division of labour in a group of robots performing an object retrieval task or response-threshold reinforcement model (Bonabeau et al.

(1997); Theraulaz et al. (1998)) that is inspired by wasps is applied for swarms of robots (e.g. in White and Helferty (2005); Yang et al. (2009)), or in Schmickl et al. (2007) a *trophallaxis-inspired* strategy which is inspired by food exchange of honeybees and ants is applied to a simulated robot swarm.

In this paper, we are interested in a mechanism of division of labour that is inspired by behaviours of honey bees (Huang and Robinson (1992, 1999)). A honey bee undertakes different tasks during its life-time in a process of behavioural development. In earlier weeks of its adult life, a honey bee performs nursing, then it performs other tasks inside the hive, and only in its final weeks it leaves the hive for foraging (Johnson (2010)). This behavioural development can be delayed, accelerated, or even reversed in response to changes in colony or environmental conditions. *Social inhibition* is proposed (Huang and Robinson (1992)) as a conceptual method for maintaining this adaptive behaviour of the colony. In this method, tasks are considered in an ordered sequence. The behavioural development of an individual that determines when the worker switches to the neighbouring task in the sequence is regulated via local interactions with other individuals.

In this paper, a distributed algorithm of division of labour based on local communication is inspired by social inhibition. The algorithm considers the spatiality of the task-regions that restrict the possible local interactions between the individuals as it is more realistic regarding many application areas and also the biological system. For example, in a honey bee colony, the workers of the tasks which are early in the sequence stay inside the hive while the workers of the tasks later in the sequence work outside the hive. The very early workers do not have much contact with the out-of-the-hive workers. In other words, the interactions are restricted to some extent to the individuals of the tasks which are next to each other in the sequence.

Despite the biological source of inspiration of the proposed algorithm, we aim for applications in swarm robotics. In particular, where there is a spatial clustering of the robots based on their tasks that limits the local communications

to the robots of the same or neighbouring tasks. The proposed algorithm is simple and easy-to-implement. It is implemented in swarms of virtual agents as well as swarms of simulated robots that perform several tasks. The behaviour of the swarm in response to the changes in the number of agents in each task and the task-demands are investigated representing adaptability that is achieved by the algorithm.

Social Inhibition

In honey bee colonies, division of labour is mainly based on the age of the workers. This mechanism is called *temporal polyethism*. In temporal polyethism, there is a correlation between the age of the workers and the tasks they perform; e.g. older workers perform tasks outside of the hive and younger workers perform tasks within the hive (Wilson (1971); Robinson (1992)). The behavioural development of the bees is associated with their physiological development such that the physiological age of a bee indicates the main task that it performs (Winston (1987); Beshers et al. (1999)).

As it is shown in different studies (e.g. Huang and Robinson (1996)), honey bee colonies are flexible to changes in age distribution of the colony and task demands. For example, in a colony of young honey bees, the age in which a bee starts foraging (an outside task) is lower than in a normal colony. It means the behavioural development in such colony is accelerated. On the other hand, presence of older bees delays or inhibits the development of physiological age of other bees in the colony. Another example is the behaviour of the colonies when the hive workers are removed. In this case, the development of physiological age decreases and inverts resulting in transformation of out-of-hive workers into inside-the-hive workers.

Huang and Robinson (1992) proposed that worker-worker interactions drive mechanisms of hormonal regulation in bees resulting in a social inhibition that explains temporal polyethism and adaptability of the colony to different age distributions. The concept is then used in other researches toward developing models of social inhibition, e.g., Beshers (2001); Naug and Gadagkar (1999); Gadagkar (2001). In this work for some reasons Naug and Gadagkar (1999); Gadagkar (2001) are not used as a source of inspiration.

Previously Suggested Algorithm: Evolution Maps

One of the models of social inhibition following Huang and Robinson (1992) is the model of *evolution maps* proposed by Beshers (2001). In this model, a “map” which is a set of curves describing changes in the physiological age of the individual is introduced. A state variable x represents the physiological age of every individual. In addition to that, the individuals also contain an auxiliary variable y . In every time-step (day for bees), an individual has a number of interactions with others. y is a weighted average of x values of all the interacted individuals. The weights are set based on the task the individual is performing. Every curve of the

“map” describes changes of x based on its current value and the value of y .

In the reported work (Beshers (2001)) two tasks are implemented. A threshold is set to indicate which task is chosen by the individual based on its x value. The threshold is augmented with higher and lower margins providing more stability for the system. The curves are derived based on experimental data from real bees.

Although the model might be extendible to more tasks, global-range interactions (interactions between individuals irrespective to the task they are performing) would be a necessary condition for the stability of the model. The reason is that if the interactions are restricted, e.g. to the neighbouring tasks in the task-sequence, the y value is no longer an estimation of the x in the whole system and will have instantaneous changes when an individual switches between two tasks leading to endless back and forth switchings.

Apart from the complexity of generating a proper map, the global-range interaction is usually not the case in both insects and robotic tasks, since workers of different tasks are usually separated spatially and interactions are restricted to individuals of the same or neighbouring tasks.

New Proposed Algorithm

One of the properties that we are interested in them are the ability of the decentralized algorithm to divide the swarm into groups relative to the task-demands while the system is flexible to changes in the demands and workforce. In addition, the number of switchings between different tasks should be limited due to practical costs (e.g., a robot may need to spend some energy to change its working area in order to perform a different task). Therefore, specialization of the individuals is also of our interest.

In the proposed model, every individual contains a state variable x as its physiological age. This variable is restricted to a defined range of (x_{min}, x_{max}) . There is a number of tasks with their associated demands. The tasks are ordered in a sequence such that an individual can only switch to the previous or the next task in the sequence.

An individual chooses a task based on its x and a set of defined thresholds that separate the tasks (see Figure 1). The thresholds are used together with lower and upper margins. For an individual that is performing $task_i$, in order to switch to $task_{i+1}$, its x value should exceed $th_{i:i+1} + upper_margin$. For an individual that is performing $task_i$, in order to switch to $task_{i-1}$, its x value should become lower than $th_{i-1:i} - lower_margin$. The lower and upper margins prevent the individuals from instant back and forth switches between two consecutive tasks due to noisy changes in x .

The main idea of the algorithm is to spread the x values of the whole swarm uniformly over the range of (x_{min}, x_{max}) . With such a uniform distribution of x throughout the swarm and setting the thresholds such that the range is split into segments relative to the task-demands, the required number

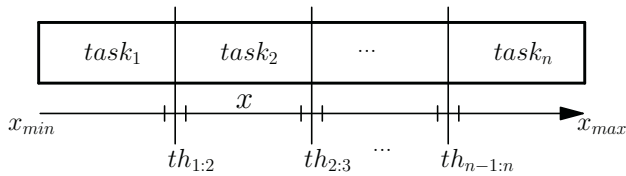


Figure 1: x changes with small steps in a range of (\min_x, \max_x) . A task is assigned to the individual when its x variable passes the respective thresholds.

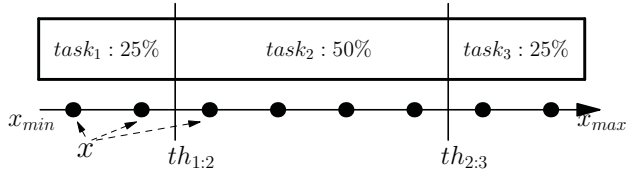


Figure 2: An example uniform distribution of x and proper thresholds that split the range relative to the task-demands. In this example, eight agents are divided into three tasks. The relative demands for the three tasks are 25%, 50%, and 25% respectively.

of agents will be assigned to each task (see Figure 2).

Regulation of x values in the swarm occurs through local interactions between the individuals. Every individual contains two variables that keep track of its experience in the swarm. In an ideal case, these two variables stand for the closest higher and lower x values belonged to other individuals in the swarm. In practice, these variables have to be estimated during interactions. Let these variables be x^{low} and x^{high} . In the ideal case their variables are as follows:

$$\begin{aligned} x^{low} &= \arg \min_{x_i} (|x - x_i|), \text{ where } x_i < x \\ x^{high} &= \arg \min_{x_i} (|x - x_i|), \text{ where } x_i > x \end{aligned} \quad (1)$$

The value of x is updated for each agent in the direction towards the average of its x^{low} and x^{high} , as follows:

$$x = \begin{cases} x + \delta & \text{if } x - x^{low} < x^{high} - x \\ x - \delta & \text{if } x - x^{low} > x^{high} - x \\ x \pm X & \text{otherwise} \end{cases} \quad (2)$$

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

Since there is no global information about the x values in the swarm, agents update their x^{low} and x^{high} values gradually during time and in every interaction with another agent.

Say agent i interacts with agent j . x^{low} and x^{high} are updated as follows:

$$x_i^{low} = \begin{cases} x_j & \text{if } x_i^{low} < x_j < x_i \\ x_i^{low} & \text{otherwise} \end{cases} \quad (3)$$

$$x_i^{high} = \begin{cases} x_j & \text{if } x_i < x_j < x_i^{high} \\ x_i^{high} & \text{otherwise} \end{cases} \quad (4)$$

x^{low} and x^{high} also slowly drift away from x in every time-step in order to be adaptable to changes in other agents' x values as well as the environment:

$$\begin{aligned} x_i^{low} &= x_i^{low} - \varphi \\ x_i^{high} &= x_i^{high} + \varphi \end{aligned} \quad (5)$$

where φ is a value smaller than δ in Eq. 2.

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

$$new_task = \begin{cases} task_{i+1} & \text{if } x > th_{i:i+1} + l_u \\ task_{i-1} & \text{if } x < th_{i-1:i} - l_b \\ task_i & \text{otherwise} \end{cases} \quad (6)$$

where $th_{i-1:i}$ and $th_{i:i+1}$ represent the threshold values between $task_{i-1}$ and $task_i$, and $task_i$ and $task_{i+1}$ respectively. l_u and l_b are the upper and lower margins for the thresholds.

The Algorithm The following actions are performed by any agent i in every time-step of running:

1. update x^{low} and x^{high} using Eq. 5.
2. if there is an interaction with agent j :
 - (a) update x^{low} and x^{high} using Eq. 3 and Eq. 4.
 - (b) update x using Eq. 2.
 - (c) update the assigned task using Eq. 6.

Experiments

A number of experiments are performed in order to investigate the performance of the proposed algorithm, its adaptivity to changes, and specialization of the agents to the tasks.

In the first set of experiments a swarm of virtual agents is simulated and the interaction between the agents performing the same task or in the adjacent tasks occur based on defined probabilities. The sensitivity of the algorithm to the chosen values for the step-size δ is also investigated.

In the second experiment, a simulated swarm of moving agents (robots) is investigated and the interactions are based on the location of the agents in the arena in every time-step of the simulation.

Virtual swarm experiments

The algorithm is first tested in a number of virtual swarms of agents.

In all of the following experiments a sequence of five different tasks is considered. Every experiment is repeated for

25 independent runs. Every run is simulated for 50000 time-steps while each agent runs the proposed algorithm. Interactions are possible between the agents of the same task or neighbouring tasks. In every time-step of the simulation, 30 interactions are sampled from a uniform distribution over all the possible interactions. All the agents are initialized by x_{min} for the state variable x . Experimental settings of the algorithm are represented in Table 1.

Investigation of the algorithm with fixed settings In the first experiment the demands for the five tasks are equal. The progress of the number of robots in each task over time is represented in Figure 3 (left). Since the agents start with the same initial value for x (x_{min}), they all start with *task1*. By occurring interactions during time the x values are spread in the range and the swarm is split for performing different tasks.

In the second experiment the demands for the five tasks are 10%, 40%, 10%, 30%, 10% respectively. Figure 3 (right) represents the results.

All the runs reached the stable desired status for both experiments. The experiments were also repeated while the possible interactions between the agents of neighbouring tasks were limited to a fraction of the agents in each task instead of the whole agents and similar results were achieved (data not shown).

Investigation of adaptivity to changes in task-demands

In order to investigate adaptability of the swarm to changes in the task-demands, another experiment is performed starting with equal demands for every task. The task demands are then changed at time-steps 20000 and 40000.

Figure 4 (left) represents the results for this experiment. As the figure demonstrates, the swarm immediately reacts to the changes in the demands. The reason is that the x values of the swarm are spread almost uniformly in the range (in the ideal situation they are spread uniformly). By changing the task-demand, the thresholds over the range of x are changed such that the range is split relative to the new settings. Therefore, proper fractions of agents are reassigned for different tasks while the x values do not need to change.

Investigation of adaptivity to changes in work-force

In the next experiment the adaptability of the swarm to the changes in the number of agents presented in each task is investigated. In order to do that, the experiment starts with a swarm of 30 agents. In time-step 15000, 20 agents including all the agents in *task2* and *task3* are removed from the swarm. Later on in time-step 30000, 10 more agents are introduced in the swarm in the *task1*.

Figure 4 (right) represents the behaviour of the swarm in response to these changes. As the figure demonstrates, the swarm reacts to these changes by switching the tasks of the proper number of agents. The mechanism behind this reaction is as follows: When a number of agents are removed

from the system (or new agents are introduced), the uniformity of the distribution of x in the swarm is violated. The agents with x values close to the low-density (or high density) regions in the distribution react to this situation by shifting their x value towards the region (or in opposite direction). The process continues until the distribution becomes uniform again.

Investigation of specialization In order to investigate specialization of the agents for the tasks in the swarm, the number of non-necessary task-switchings of every agent is evaluated during the run-time. The settings are the same as the first experiment: fixed equal demands for all the five tasks. Since all the agents start with *task1* due to initial value for x , a certain number of switchings from a task to the next one is necessary for a certain number of agents. Moreover, any switching to a previous task (switch-back) is not desirable. Figure 5 represents the frequency of switch-backs during 50000 time-steps for all runs and agents for different values of step-size (δ).

Investigation of the effects of step-size The step-size δ in Eq. 2 is a predefined parameter in the current implementation of the algorithm. Therefore, the sensitivity of the algorithm to this parameter is investigated by repeating the first experiment with different values for δ . Figure 6 represents a comparison for different values. The main figure demonstrates the median error of the task allocation (number of agents in wrong tasks) over time. The inline figure compares the time required for reaching a swarm-state stabilized in maximum of 5% error.

As it is represented in Figure 6, for very small values of δ , error decreases slowly. It is more quick for middle values. For high values of δ , the error decreases quickly regarding the main figure, but regarding the inline figure the time to reach the stable status with maximum of 5% error is high. In addition, the sizes of quartiles are big indicating that in different runs different values are calculated for the time-to-reach. The instability of the high values is also visible in Figure 5 that represents the frequency of switch-backs for different δ for all the runs during 50000 time-steps. In this figure, for $\delta = 0.01$ there was not a single switch-back in all the runs. As δ increases, the frequency of switch-backs also increases. In short, if the step-size (δ) is too small, the system is less reactive, and convergence takes longer. But if the value is too big, the system gets instable and results differ from case to case.

Simulated robot experiment

In this experiment a simulated robot swarm running the proposed algorithm is investigated for its behaviour and adaptability to the changes in the workforce and task demands.

A square arena is set up with a light source located in one side (see Figure 7). A swarm of robots is supposed to be split

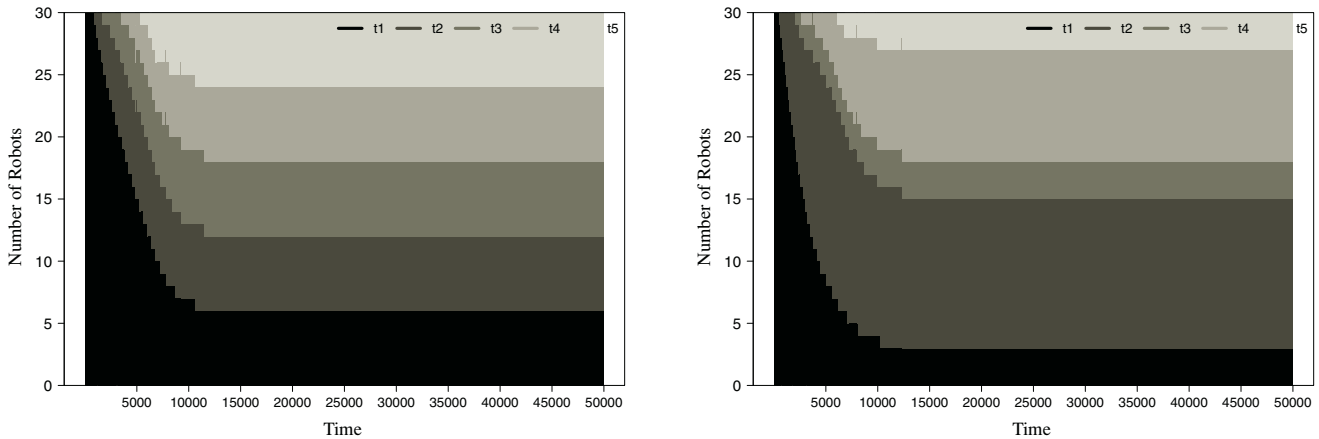


Figure 3: Medians of the number of agents in each task over time for 25 independent runs. All of the runs reached the stable solution for both experiments. Tasks are represented by t_1, t_2, t_3, t_4, t_5 . The left figure represents five tasks with fixed equal demands. The right figure represents five tasks with fixed demands of 10%, 40%, 10%, 30%, 10% respectively. Number of agents in both experiments are 30.

<i>Time</i>	0	20000	40000
$th_{1:2}$	20	10	10
$th_{2:3}$	40	50	20
$th_{3:4}$	60	60	30
$th_{4:5}$	80	90	60
<i>Swarm Size</i>	30	30	30

<i>Time</i>	0	15000	30000
$th_{1:2}$	10	10	10
$th_{2:3}$	50	50	50
$th_{3:4}$	60	60	60
$th_{4:5}$	90	90	90
<i>Swarm Size</i>	30	10	20

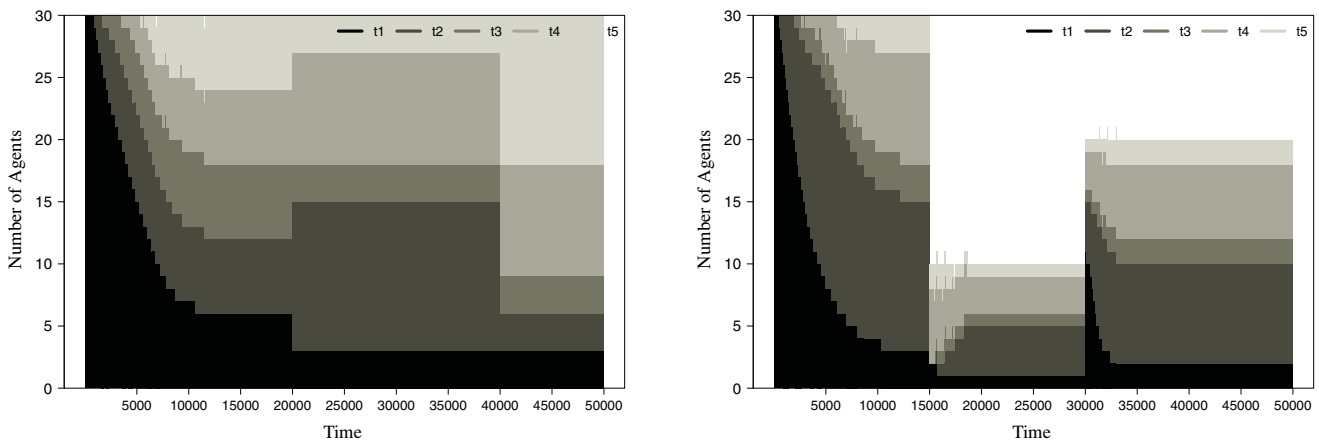


Figure 4: Medians of the number of robots in each task in 25 runs. All of the 25 runs reached the solution for both experiments. In both experiments the swarm starts with 30 agents. The left figure represents changes in the demands in time-steps 20000 and 40000. The right figure represents changes in the number of robots. In time-step 15000, 20 agents including all the agents in second and third task and randomly chosen agents are removed from the system. In time-step 30000, 10 agents are added to the system in the first task.

Downloaded from <http://direct.mit.edu/isaip/proceedings-pdf/ecal/2013/25/609/1901353/978-0-262-31709-2-ch087.pdf> by guest on 18 October 2021

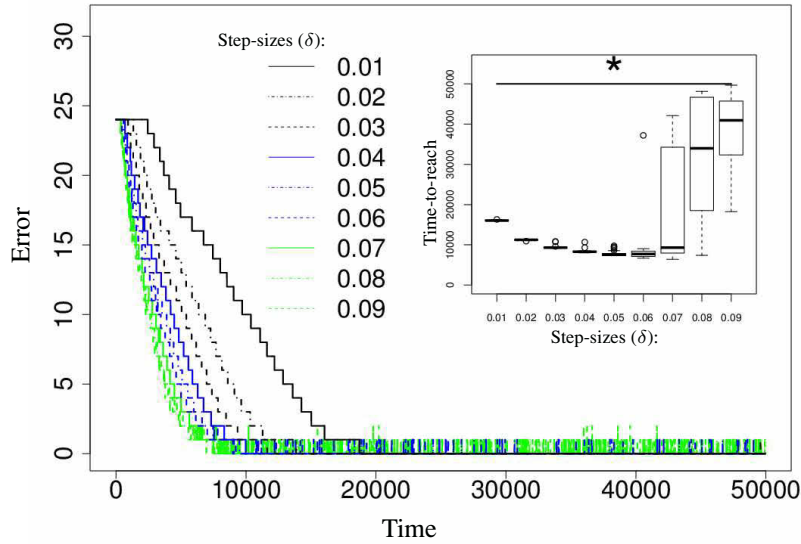


Figure 6: Comparison of error over time and also time-to-reach the stable status for nine different values of step-size δ based on 25 independent runs. Main figure represents the median of the number of agents in wrong tasks (errors) over time for 25 runs. Inline figure represents the boxplots of the time to reach a swarm state that is stable with maximum one error out of 30 agents (more than 95% correct task-assignment) (*: $p < 0.01$ for all pairs of δ except (0.01,0.07) where $p < 0.25$; Wilcoxon signed-rank test, unpaired data, "two.sided"-hypothesis). Box-plots indicate median and quartiles, whiskers indicate minimum and maximum, circles indicate outliers.

Table 1: Experimental settings for the proposed social inhibition algorithm

	virtual scenarios	robot scenario
x_{min}	0	0
x_{max}	100	100
δ	0.03	0.1
φ	$\delta/30$	$\delta/3$
l_u, l_b	1	1

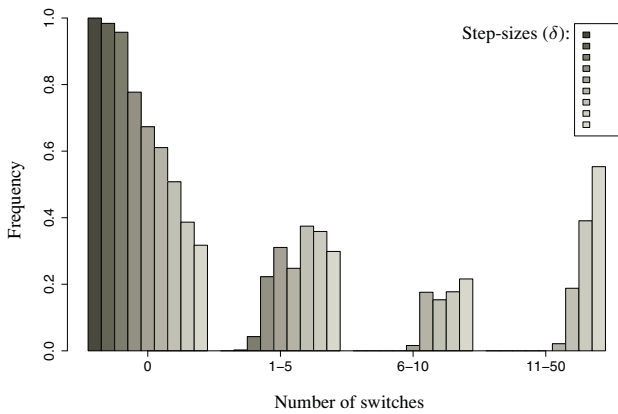


Figure 5: Frequency of number of switch-backs in 50000 time-steps for all the 30 agents and 25 runs. The figure represents the frequency for different values of step-size δ .

up into three regions of the arena in order to cover the arena with different densities. Every robot has a luminance sensor perceiving the brightness of the light sensor. The three regions are located in different distances from the light source and the robots identify them based on their brightness. Each robot is able to rotate or move forward. When a robot decides to switch the task, it moves uphill or downhill the luminance gradient until it reaches the appropriate working region. Robots move randomly in their working region and do not leave it unless they decide to switch the task. A robot can interact with another robot which is located in its communication range by exchanging the x values. The arena is of size 32×32 and the communication-range is five times bigger than the robots diameter.

Every robot i in the simulation performs the following:

- If the robot is in the region of its assigned task:

- Perform a random walk inside the task-region.
- Otherwise (the robot is out of its task-region):
 - turn towards the proper task-region based on the brightness gradient and step forward.
- regulate the state variables:
 1. update x^{low} and x^{high} using Eq. 5.
 2. if there is a robot j in the communication range:
 - (a) update x^{low} and x^{high} using Eq. 3 and Eq. 4.
 - (b) update x using Eq. 2.
 - (c) update the assigned task using Eq. 6.

At the beginning of the experiments, the number of robots is 16 and the task-demands are 25%, 50%, and 25% respectively which means 4, 8, and 4 robots are desired for each task-region. At time-step 20000, all the 8 robots in *task2* are removed from the arena but the proportional demands are fixed. At time-step 40000, the demands are changed to 25%, 25%, and 50% respectively. The experiment is repeated for 25 independent runs.

Figure 8 represents the progression of the number of robots in each task over time. As it is demonstrated in the figure, the swarm reacts properly to the changes in the number of robots by switching a proper number of robots from *task1* and *task3* into *task2*. The system also quickly responds to the changes in the task-demands by switching two robots from *task2* to *task3*.

Figure 9 represents the frequency of switch-backs during the first 10000 time-steps of all the runs. The figure represents that about 0.47% of the robots have not a single switch-back during the whole evaluation-time and very few robots had more than 10 switch-backs representing specialization for the robots in the tasks they perform.

Conclusion

This paper introduces a novel decentralized, self-organized and self-regulated division of labour in artificial swarms inspired by temporal polyethism in honey bees. The algorithm is based on local communication while the communications need to be possible only between the agents that perform the same task or neighbouring tasks. The logic behind the algorithm is simple and it is easy to implement while the interesting properties are maintained for the swarm. Experiments investigating the behaviour of the swarm in response to changes in the swarm members or task-demands represents that the algorithm provides a high adaptivity for the swarm. In addition, it is demonstrated that the agents are specialized in the tasks and unnecessary switchings between the tasks are limited. The sensitivity of the system to a pre-defined parameter of the algorithm (step-size) is also investigated indicating that there is a trade-off between the speed of approaching the solution and stability of the swarm. In the future the algorithm will be extended for more complicated requirements and will be used in real-robot scenarios.

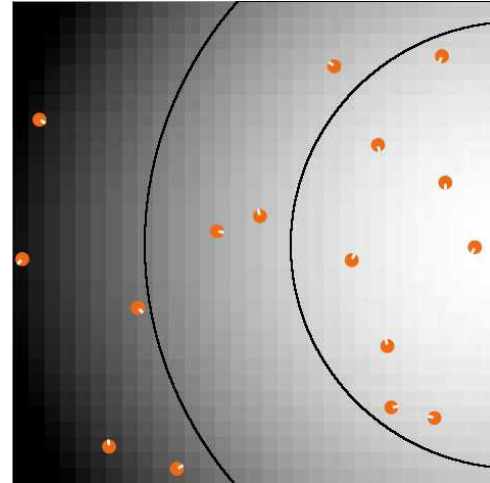


Figure 7: A screenshot of the robot arena. Three different task-regions are located based on their distances from the light source.

Time	0	20000	35000
$th_{1:2}$	25	25	25
$th_{2:3}$	75	75	50
Swarm Size	16	8	8

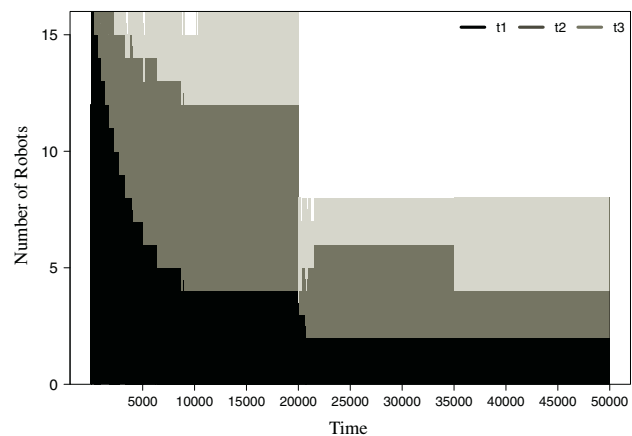


Figure 8: Medians of the number of robots in each task in 25 runs. The system starts with 16 robots while the demands for the three tasks are 25%, 50%, and 25% respectively. At time-step 20000, all the robots in the second task are removed from the arena. In time-step 35000, the demands change to 25%, 25%, and 50% respectively. Tasks are represented with t_1 , t_2 , t_3 .

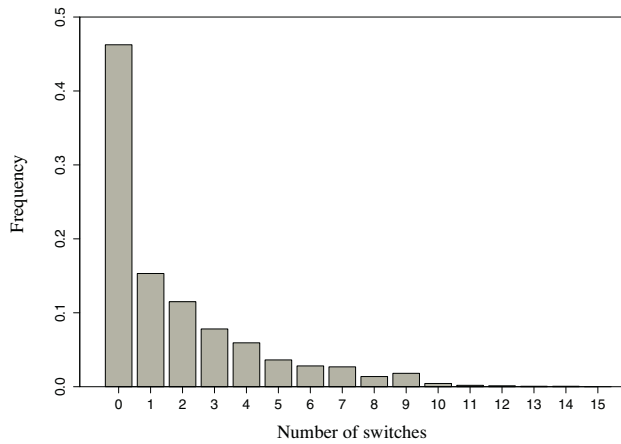


Figure 9: Frequency of number of switch-backs in the first 10000 time-steps for all the 16 agents and 25 runs.

Acknowledgements

This work is supported by: EU-ICT project ‘CoCoRo’, no. 270382; EU-IST-FET project ‘SYMBRION’, no. 216342; EU-ICT project ‘REPLICATOR’, no. 216240; Austrian Federal Ministry of Science and Research (BM.W F); EU-ICT project ‘ASSISI.bf’, no. 601074.

References

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

Beshers, S. and Fewell, J. (2001). Models of division of labor in social insects. *Annu. Rev. Entomol.*, 46:413–440.

Beshers, S., Robinson, G., and Mittenthal, J. (1999). Response thresholds and division of labor in social insects. In Detrain, C., Deneubourg, J., and Pasteels, J., editors, *Information Processing in Social Insects*, pages 115–139. Birkhuser Basel.

Bonabeau, E., Sobkowski, A., Theraulaz, G., and Deneubourg, J.-L. (1997). 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.

Gadagkar, R. (2001). 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.

Gross, R., Nouyan, S., Bonani, M., Mondada, F., and Dorigo, M. (2008). Division of labour in self-organised groups. In *SAB*, pages 426–436.

Hölldobler, B. and Wilson, E. (2008). *The Superorganism: The Beauty, Elegance, and Strangeness of Insect Societies*. W. W. Norton and Company, New York.

Huang, Z.-Y. and Robinson, G. (1999). Social control of division of labor in honey bee colonies. In Detrain, C., Deneubourg,

J., and Pasteels, J., editors, *Information Processing in Social Insects*, pages 165–186. Birkhuser Basel.

Huang, Z. Y. and Robinson, G. E. (1992). 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.

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

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

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

Karsai, I. and Schmickl, T. (2011). Regulation of task partitioning by a “common stomach”: a model of nest construction in social wasps. *Behavioral Ecology*, 22:819–830.

Labella, T. H., Dorigo, M., and Deneubourg, J.-L. (2006). Division of labor in a group of robots inspired by ants’ foraging behavior. *ACM Trans. Auton. Adapt. Syst.*, 1(1):4–25.

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

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

Schmickl, T., Möslinger, C., and Crailsheim, K. (2007). Collective perception in a robot swarm. In Şahin, E., Spears, W. M., and Winfield, A. F. T., editors, *Swarm Robotics - Second SAB 2006 International Workshop*, volume 4433 of *LNCS*, Heidelberg/Berlin, Germany. Springer-Verlag.

Seeley, T. D. (1982). Adaptive significance of the age polyethism schedule in honeybee colonies. *Behavioral Ecology and Sociobiology*, 11:287–293.

Theraulaz, G., Bonabeau, E., and Deneubourg, J.-L. (1998). Response threshold reinforcement and division of labour in insect societies. In *In Proc. Royal Society of London*, volume 265 of 5, pages 327–332.

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

White, T. and Helferty, J. (2005). Emergent Team Formation: Applying Division of Labour Principles to Robot Soccer. *Engineering Self-Organising Systems*, pages 180–194.

Wilson, E. O. (1971). *The insect societies*. Belknap Press of Harvard University Press.

Winston, M. (1987). *The biology of the honey bee*. Harvard University Press.

Yang, Y., Zhou, C., and Tian, Y. (2009). Swarm robots task allocation based on response threshold model. In Gupta, G. S. and Mukhopadhyay, S. C., editors, *ICARA*, pages 171–176. IEEE.