Available online at www.sciencedirect.com
ScienceDirect
IFAC PapersOnLine 51-2 (2018) 385–390

A Model for Bio-Inspired Underwater Swarm Robotic Exploration

Joshua Cherian Varughese [*,**] **Ronald Thenius** [*]
Paul Leitgeb [*] **Franz Wotawa** [**] **Thomas Schmickl** [*]

[*] *Institut für Zoologie, Karl Franzens Universität Graz, Austria*
[**] *Institut für Software Technologie Technische Universität Graz, Austria*

Abstract: Swarm intelligence has interested researchers in various areas of research for several decades because of its stability, resilience and simplicity. Several researchers have used swarm intelligence behaviours to design systems which can accomplish single tasks. In this paper, we will make a step forward by designing a swarm intelligent system that draws from two different natural swarms, bees and slime mould, to form an integrated underwater swarm robotic exploration system. An agent based simulation of such a system is presented in this paper along with some basic performance evaluation measures of the presented system. The main question the authors are attempting to answer through this model is how feasible the such an exploration system would be with regards to time, the number of robots allocated by the decentralized system for exploring interesting locations and the resilience of such a system to failures in robots. The first simulation results obtained from this model shows how decentralized control inspired by swarm intelligence can be used to design systems for real world applications.

Keywords: underwater robotics, swarms, underwater measurements, swarm intelligence, bio-inspiration, bees, slime mould

1. INTRODUCTION

Swarm intelligence is a widely researched phenomenon in bees (Schmickl and Crailsheim, 2004), fireflies (Narayanan et al., 2017), slime mould (Nakagaki, 2001; Trianni et al., 2003) etc. A well known aspect of swarming behaviour that makes it attractive to technical systems is its resilience (Varughese et al., 2017) and its simplicity. While many practical applications for swarm intelligent engineered systems have been suggested (Tan and Zheng, 2013), swarm intelligent systems are only beginning to find applications in engineered systems. Swarm intelligence is especially interesting in systems where a high number of entities makes it difficult to design classical centralized controllers.

The project subCULTron (subCULTron, 2015) aims to develop an autonomous underwater robotic society (Thenius et al., 2016) comprising of three swarms of bio-inspired robots that monitor the environment in a marine habitat. The three robotic swarms forming the society are:

(1) "aPads" which are robots that act as base stations on the ,water surface for docking with other swarm members, communicating wit,h external entities, collecting solar energy, etc.
(2) "aFish" are a group of agile robots which can move around underwater for exploring new areas and exchanging information between sub swarms.
(3) "aMussels" are a swarm of robots with high sensing abilities and very low power consumption. They dive down to the water body bed to collect data and energy.

This robotic society will be deployed in the environmentally diverse and dynamic lagoon of Venice to perform long term measurements and exploration. The subCUTLron system aims to produce time synchronized measurements of various parameters at different locations which will be beneficial for oceanographic research in Venice. The subCULTron system stands out from traditional engineered systems as it utilizes a combination of the strengths of classical control systems and naturally occurring swarm intelligent behaviours to accomplish its goals. The designed system will have to identify areas of interest using the aFish and then communicate the coordinates of these interesting locations to the aMussel swarm. Once there is sufficient data collected, the aMussel swarm collectively decides to split into groups in order to explore the interesting areas suggested by the aFish. In this paper, the authors model one of the ma,in tasks of project subCULTron - a robotic monitoring and exploratory system which is autonomous, self-sustaining and resilient. The question the authors seek to answer through this model are as follows:

(1) How feasible is the designed system with respect to time taken to explore the area?
(2) How well do aMussels split into groups to perform monitoring at the "areas of interest"?
(3) How stable is the collective decision of aMussels in case of aMussel failures during operation?

The exploration system described in this paper is inspired by natural swarms that explore their surroundings to scout for food. During exploration, the agile robots, aFish,

[⋆] This work was supported by EU-H2020 Project no. 640967, subCULTron, funded by the European Unions Horizon 2020 research and innovation programme.

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2018.03.066

perform the role of scouts by moving around in the environment to search for good quality food sources which is later exploited by foragers. Bees are one of the widely researched and most sophisticated foragers in the animal kingdom. In honeybee colonies the foraging activities are controlled by a self-organised process that uses dances as communication vector (Frisch, 1965; Seeley, 1992), as well as individually perceived waiting times (Schmickl et al., 2010; Anderson and Ratnieks, 1999). Due to a complex network of feedbacks of individually experienced queuing delays while unloading and observed dances, different groups of bees are able to adjust workload balance in a de-central, self-organised manner and optimise colony efficiency. Further the colony is able to detect changes in the environment, and react adequately to them (Schmickl and Crailsheim, 2004; Thenius et al., 2008). Since bees do several tasks the subCULTron system will need to cope with, we will model the exploration by aFish inspired by bees. Another aspect of decentralized decision making is ensuring that the agents have the same information as far as possible in order for them to make coherent decisions. In a classical engineering system, this can be accomplished by broadcasting information to all agents. However, since the aMussels and aFish in project subCULTron has local communication using blue light (Thenius et al., 2016), such a broadcasting mechanism is not an option. In order to ensure this, we take inspiration from the chemical signal (cAMP) based communication mechanism of slime mould (Alcantara and Monk, 1974). This communication mechanism has has inspired various algorithms in technical systems (Nakagaki, 2001; Varughese et al., 2016).

2. METHOD

As explained in Section 1, we employ a method which is inspired by the dance based source selection system of bees for aFish and the slime mould inspired scroll waves for information spreading through the swarm. We will briefly look into the behaviours that we take inspiration from and how the behaviours are implemented in the agent based model developed in this paper.

2.1 Recruitment by waggle dance

The mechanism by which worker bees recruit other workers to a food source has been studied for more than half a century (Frisch, 1965; Seeley, 1992). When a rich food source is discovered by a worker bee, she returns to the hive and recruits her nest mates to join with her to exploit the food source (Biesmeijer and Seeley, 2005). The mechanism by which the worker bee recruits her nest mates is known as a “waggle dance” through which a bee performs a dance symbolizing her recent journey to the food source. Through this dance, the nest mates get a rich amount of information including the distance to the food source, its direction and even the odour of the food source. This dance can be decrypted by the other bees into a flight path to the food source referred to by the dancer. Some bees which see this dance fly to the food source and depending on the quality of the food source, more and more bees are recruited by more dancers. The frequency of dance is correlated with the quality of the food source and hence a bee trying to communicate a better source is likely to attract more followers (Seeley et al., 1991).

Fig. 1. A typical instantiation of the subCULTron exploration system modelled in Netlogo in shown in the figure. The array of grey circular dots represent the group of connected aMussels (bottom left corner), the red arrows represent the aFish (surrounding the group of aMussels). The long blue patches represent obstacles and the smaller patches in different shades of yellow represent the interesting areas coloured according coloured according to how interesting the areas are.

.

2.2 Scroll waves based information exchange

Slime mold (*Dictyostelium Discoideum*), is a free living diploid life form that has been studied by many researchers in the past. Chisholm and Firtel (2004) divide its life cycle as follows: aggregation, streaming, slug, culmination and fruiting body. Each organism starts its life as a unicellular amoeba, but during starvation they aggregate with other slime mould cells to form a multicellular organism. During the aggregation phase, some cells release a quickly diffusing chemical into the environment known as Cyclic Adenosine Monophosphate (cAMP) (Siegert and Weijer, 1992). When other cells perceive this chemical spike, they move towards areas of high cAMP concentration and release cAMP themselves, thereby relaying the signal. Through repeated relaying of the signal through the neighbouring cells, the entire swarm is attracted towards the original cell that produced the signal. Slime mould cells are able to release cAMP at an interval of 12-15 seconds (Alcantara and Monk, 1974) during which the cells are insensitive to other incoming cAMP signals. This interval is known as a refractory phase and is crucial towards relaying of the signal. The communication mechanism described above causes “scroll waves” to propagate through the swarm which enables the aggregation of slime mould to the original cell that produced the chemical spike.

2.3 The subCULTron exploratory system

In the agent based model, we will employ two types of agents to represent the aMussels and the aFish. The task of the entire system is to find areas of interest, communicate it to the aMussel so that the aMussels can make a decision to move from the initial deployment area to areas of interest in a purely de-central manner. While introducing a designing for the robotic exploration system it is mandatory to talk briefly about the scope of relevant physical capabilities of the robots developed in project subCULTron as well as physical dimensions of the modelled environment.

Figure 1 shows an instantiation of the subCULTron exploration system before the the respective run started. The aMussels and aFish are represented in Figure 1 as circular and arrow like shapes respectively. The yellow patches shown in Figure 1 are areas that would be of potential interest to the exploration system. Each interest area represented by yellow patches have an associated quality values. Each patch in the model is taken to be an area of one square meter. The area shown in Figure 1 is comparable a 10,000 square meters (or 1 hectare) area in the real world. The blue obstacles are added at random at the beginning of each simulation run to represent shallow areas. In reality, aFish robots can move at an average speed of 0.5 meters per second. In simulation, one tick is considered to be one second in real time and hence the average speed the red arrow like agents representing the aFish in simulation is 0.5 patches per tick. The aMussels are represented by the white dots in simulation; a relevant similarity between the agents in the simulation and the aMussels in the real world is the communication mechanism. aMussels communicate using modulated blue/green light which can be perceived by other robots in its surroundings. This communication constraint is implemented in simulation in it that any message broadcast by any aMussel can be heard by all other agents within a patch distance of 1.5 patches which is analogous to 1.5 meters in the real world environment. In the real world, from our tests, the range of blue light communication under water varies from 0.7 to 2 meters depending on the turbidity of water (Thenius et al., 2016). Unless explicitly stated otherwise, all the experiments conducted in this paper will be done with the three areas of interest to be explored, 100 aMussels and 20 aFish. As far as this model is concerned, the aPad would only acts as a robot which would transport aMussels from one place to the other and therefore, we will omit aPads from this model. Instead of aPads, we can safely assume without affecting the performance of the model that aMussels can make a one time movement from instantiation patch to an area of interest.

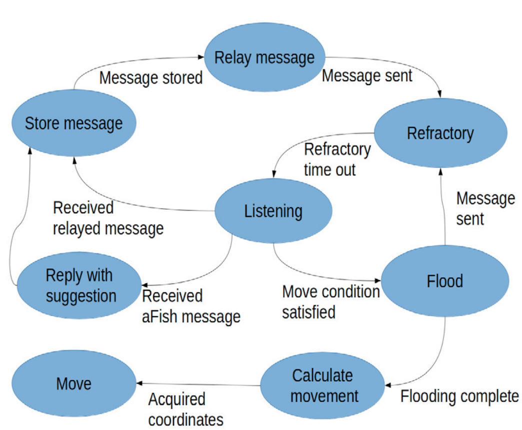

Fig. 2. State transition diagram of aMussels are shown in the diagram

.

aMussel behaviour Inspired by the scroll wave based communication of slime mould, we employ this method for diffusion of information through the swarm. Any message given to any aMussel will be communicated to its neighbours and in consequence, any message will propagate through the swarm similar to the "scroll wave" in case of the slime mould swarm. This scroll wave based message propagation is accomplished by having the following basic behaviour as shown in Figure 2. By default, all the aMussels are in "listening" state. All the aMussels contain a database where it stores the interesting locations it received from the aFish. During the listening state, the aMussel prepares a "source suggestion" for the aFish by inspecting its own database of interesting locations and computing the least reported location. If an aMussel receives a message, it transitions into the "store message" state where it stores the message it received and then it sends out the same message in the "relay message" state. After broadcasting the message by means of a scroll wave, the aMussel transitions into the "refractory" state where it remains insensitive to all incoming messages for a particular period of time called the refractory time, say t_r. Following the time period, t_r, aMussel starts listening again to incoming messages. For the sake of simplicity, we will assume that the termination condition to the listening phase is based on the number of messages received,say M. When any aMussel receives M number of messages, the aMussel will trigger the rest of the swarms transition out of the listening phase into the "flood" state. For all experiments in this paper, M is taken to be 100 messages about the least explored patch. This state is called the "flood" state. In the "flood" state the aMussels floods the rest of the swarm with randomly chosen messages from its database. Flooding the swarm continues for a certain number of cycles, say F and then each agent transitions into the "calculate movement" state where it decides to move to a location based on its internal list of messages. As time passes, the internal database of aMussels will have a list of interesting locations suggested by the aFish. The aMussels cluster this data and produce a reduced number of location suggestions and performs a roulette wheel selection based on the quality values of candidate solution to choose one location. Thus, each aMussel will have one goal after the selection process. Subsequently, the aMussel transition into the "move" state in which the actual movement is executed. In the real world experiments, aPad will transport the aMussels from point A to point B. However, for the sake of simplicity, we will assume that the aMussels can move from point A to point B. The decision on which location to move is an individual decision which emerges into an adequately coherent collective decision on a swarm level.

aFish behaviour The main task of the aFish is to explore the area and identify areas which might be interesting for the rest of the swarm. This is quite similar to what bees do when they scout to find sources of pollen and nectar. Taking inspiration from bees, aFish will move around randomly by default looking for interesting areas as well as for aMussels. As shown in Figure 3, this state is called "Scouting". For the sake of simplicity, we assume that aFish are able to detect interesting areas when they move "interesting" patches in the simulation environment.

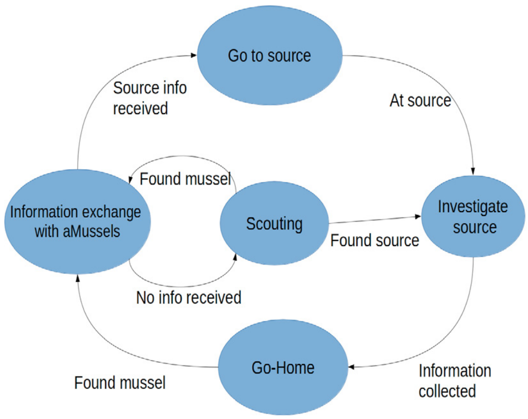

Fig. 3. State transition diagram of aFish are shown in the diagram

.

If an aFish encounters an interesting patch, it transitions into the "inspect source" state and records the location and quality of the patch. If an aFish already knows the position of an aMussel, it moves towards the aMussel directly to report the finding in the "Go Home" state. Otherwise, the aFish resumes the "scouting" state to find an aMussel to report the found location. Once an aMussel is found the aFish reports the interesting patch and also receives a suggestion from the aMussel as to which patch to explore next. If coordinates of a particular source is received from the aMussel, the aFish goes to inspect the source in the "go to source" state. Such a bi-directional transfer of information is inspired by the bee hive in the manner how the bees even while bringing back waggle dances, receive information from the hive and the collective super organism is able to identify better food sources by performing better waggle dances. In case of aFish, a coordinate of an interesting patch to explore is passed on to the aFish by the aMussel swarm.

Performance Evaluation In Section 1, the main questions that the authors seek to answer through this model are stated. Several simulation experiments were conducted to answer these questions. For each run, randomly generated obstacles and interesting areas spread across the arena are used. In order to evaluate the system in the light of the questions, the following performance parameters have been defined:

(1) Time performance(τ): In order to check the feasibility of the system with respect to time taken to explore the area under consideration, we measure the simulation time taken till the last aMussel leaves the instantiation patch. In order to evaluate the time performance of the exploratory system, a total of 250 runs were conducted and at the end of each run (once the aMussels reach their destination patches to explore), the time taken for the last aMussel to leave the instantiation patch was measured. All runs were conducted using 100 aMussels and 20 aFish.
(2) Error in split(E): The error value between the expected number of aMussels in each group and the actual number of aMussels in each group. The expected number of aMussels in each group will be calculated according to the quality of the interesting area. Let P be the total number of aMussels available for exploration, q_i be the quality of the interesting site i, N_{ip} be the number of interesting sites and a_i be the number of aMussels which actually went to site i. Then, Equation 2 describes the error in split E of aMussels based on the decentralized algorithm proposed. Error in split, E, was evaluated by conducting 50 runs each with the number of aMussels varying from 1 to 100.
(3) Resilience: In order to test the resilience of the exploratory system we introduce probability of failure (each tick in the simulation) to the aMussels while the aFish scout the arenas. At the end of run, we measured the percentage of aMussels that failed and the error split as per Equation 2. The resilience of the system was evaluated by running an experiment similar to the "Error in split". Here, 50 runs each were conducted for each probability of failure of aMussels.

$$Q = \sum_{i=1}^{N_{ip}} q_i \quad where\ i \in \mathbb{N} \tag{1}$$

$$E = \frac{1}{P} \sum_{i=1}^{N_{ip}} \left| \frac{P q_i}{Q} - a_i \right| \quad where\ i \in \mathbb{N} \tag{2}$$

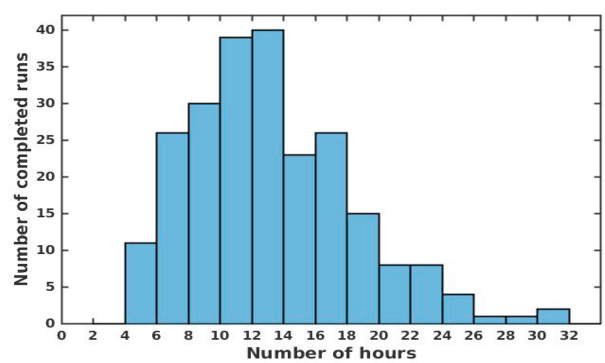

Fig. 4. Histogram shows the time performance of the system with three interest points, 100 aMussels and 20 aFish.

3. RESULTS

From Figure 4, we can see that the time performance of the system ranges from 4 to 32 hours depending on the difficulty of the region to be explored. However, most of the runs with the area of 100 square meters lies requires 10 - 14 hours to explore.

Figure 5 shows how the exploratory system designed in this paper performs with respect to the assignment of aMussels according to quality of patches to be explored. Here, we see that the error in split (E) decreases with the increasing number of aMussels.

Figure 6 shows the result of the resilience test conducted according to the performance measure introduced in the Section 2. The figure shows how the error in split, E varies with aMussel failure during simulation runs. In order to represent how many aMussel failed during the runs, the

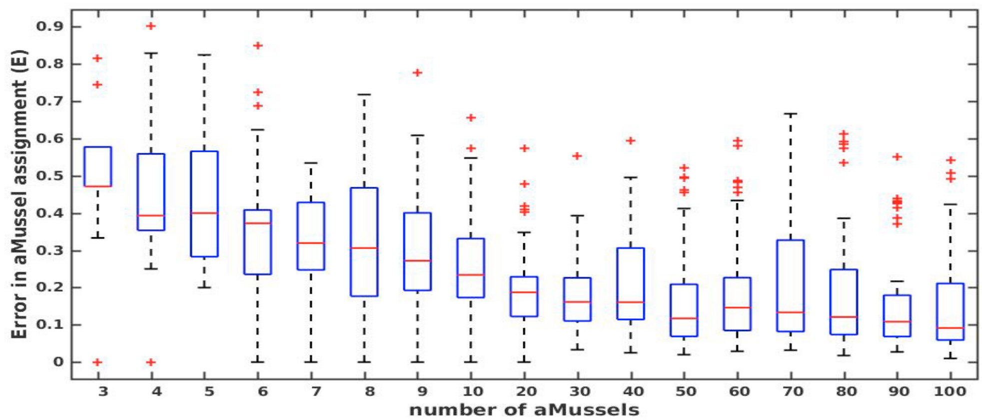

Fig. 5. Box plot showing the error in aMussel assignment calculated according to Equations 1 and 2. The plot shows that a larger the number of aMussels, smaller the error in assignment becomes.

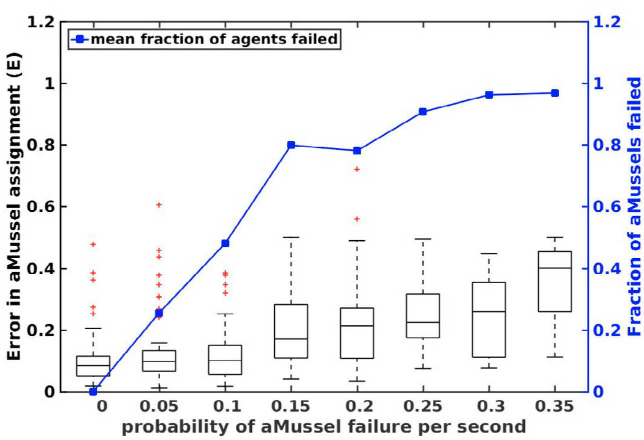

Fig. 6. Box plot showing how the error in split performs as the probability of aMussel failure increases. 50 runs were conducted per failure probability of failure and the overlaid plot (in blue) shows the fraction of aMussel that failed during the run time of the simulation.

plot showing the mean of the fraction of aMussels failed during each of the 50 runs is overlaid on the box plot. Runs were terminated and runs were discarded if the number of aMussels fell below 3 since the goal cannot be accomplished with a lower than three aMussels.

4. DISCUSSION

From Figure 4, it is evident that the run time of the system is most likely between 10 - 14 hours for every hectare of underwater area. Since the subCULTron system aims to explore underwater areas with several weeks of operation time (Thenius et al., 2016), the time taken for exploration is well within the physical operating limits of aFish and aMussel robots (Thenius et al., 2016). Thus the first question asked in Section 1 is answered with Figure 4. The time performance of the system is expected to be influenced by the number of aFish as well as the termination condition chosen for the aMussels before the "flood" state. Here, the termination condition is when an aMussel has 100 messages about the least reported point in its database.

Figure 5 shows how well the completely de-central decision to split individual aMussels performs with increasing number of aMussels. It is seen that the error is higher when the number of aMussels are low. As the number of aMussels decreases, the error in the assignment of aMussels (according to the patch quality) decreases. Theoretically, the error should tend to zero as the number of aMussels becomes large. Although such an observation speaks for using as many aMussels as possible, considering the cost of robots, a trade off between decreasing error and cost can be chosen. Additionally, this observation is extremely relevant for systems like subCULTron which operates without a central controller for assigning robots to different locations. Intuitively the error is also affected by how well the individual databases match with each other. This in turn is dependent on various factors like connectivity of the aMussels on the instantiation patch. In the simulation, we have assumed that all messages sent from any agent to another agent is reliably transmitted and received. In reality, there might be deviations from this assumption, however, Varughese et al. (2017) has shown that slime mould based behaviours are by nature shows higher resilience against communication failures. Additionally, the condition which waits for 100 data points per interesting location ensures sufficient time for the aFish to compensate for any such mishaps in communication.

The resilience of the system is shown by Figure 6. From the overlaid line plot, we see the the fraction of aMussels that failed during the simulation. It can be seen that even with around 50 % of the aMussels failing during the experiment, the system works but with larger errors in aMussel assignment. From Figure 6, the resilience of the system is sufficiently large to ensure that even as high as 10 % robots fail, the exploration system will still be able to split into groups proportional to the quality of patches. From the box plot, in case of 10 % aMussels fail, the error in aMussel assignment as per Equation 2 will be below 0.05. In case of aFish failure, we can intuitively predict that the exploration will be slower and hence the time performance will suffer without any effect on the E.

5. CONCLUSION AND FUTURE WORK

In Section 1, three questions that were asked which were to be answered by the model designed in this paper. From Figures 4, it was discussed that the time performance of the system is on average 10-14 hours but it is also dependent on the difficulty of the patches to be discovered.

The performance of the system is measured by the "Error in split (E)" of the aMussels as described in Section 2. From the discussion in Section 4, it is clear that for minimizing E, it is better to use as many aMussels as possible. Based on the model presented here, an appropriate number of aMussels can be chosen according to the application are considering error in split E, the minimum number of parallel data collection nodes required and additionally, the cost of individual robots.

As far as the resilience of the exploration system is concerned, the error in split, E, is very small for realistic probabilities of failures. Additionally, in spite of having unrealistically large amount of robots failing (as high as 50%) per run, the system is able to achieve its exploration goals. However, such runs have very high error in split, E as shown in Figure 6.

Presently, there exists hardly any technology that simultaneously draws from various swarm intelligent sources to accomplish an integrated system. The usual swarm intelligence research is limited to systems performing certain simple tasks. Although swarm intelligence is hardly implemented in real world systems, research done in project subCULTron is one of the first to integrate swarm intelligence from various sources into one single system. In the future, in addition to the already suggested behaviours, more bio-inspiration can be added to various aspects of the model.

Another scope for future work is that, currently the exploration runs are terminated when the aMussels split into groups to explore the areas of interest. The suggested exploration system can be extended to accommodate with dynamic new areas of interest that the aFish keep reporting to the aMussels even while they continue to move to explore.

REFERENCES

Alcantara, F. and Monk, M. (1974). Signal propagation during aggregation in the slime mould dictyostelium discoideum. *Microbiology*, 85(2), 321–334.

Anderson, C. and Ratnieks, F.L.W. (1999). Task partitioning in insect societies. I. effect of colony size on queueing delay and colony ergonomic efficiency. *American Naturalist*, 154, 521–535.

Biesmeijer, J.C. and Seeley, T.D. (2005). The use of waggle dance information by honey bees throughout their foraging careers. *Behavioral Ecology and Sociobiology*, 59(1), 133–142. doi:10.1007/s00265-005-0019-6. URL `https://doi.org/10.1007/s00265-005-0019-6`.

Chisholm, R.L. and Firtel, R.A. (2004). Insights into morphogenesis from a simple developmental system. *Nature reviews Molecular cell biology*, 5(7), 531–541.

Frisch, K.v. (1965). *Tanzsprache und Orientierung der Bienen*. Springer-Verlag, Berlin, Heidelberg, New York.

Nakagaki, T. (2001). Smart behavior of true slime mold in a labyrinth. *Research in Microbiology*, 152, 767–770.

Narayanan, S.J., Perumal, B., and Rohra, J.G. (2017). Swarm-based nature-inspired metaheuristics for neural network optimization. *Handbook of Research on Modeling, Analysis, and Application of Nature-Inspired Metaheuristic Algorithms*, 23.

Schmickl, T. and Crailsheim, K. (2004). Costs of environmental fluctuations an benefits of dynamic foraging decisions in honey bees. *Adapitve Behavior*, 12, 263–277.

Schmickl, T., Thenius, R., and Crailsheim, K. (2010). Swarm-intelligent foraging in honeybees: benefits and costs of task-partitioning and environmental fluctuations. *Neural Computing & Applications*. doi:10.1007/s00521-010-0357-9. URL `http://dx.doi.org/10.1007/s00521-010-0357-9`.

Seeley, T.D. (1992). The tremble dance of the honey bee: message and meanings. *Behavioral Ecology and Sociobiology*, 31, 375–383.

Seeley, T.D., Camazine, S., and Sneyd, J. (1991). Collective decision-making in honey bees: how colonies choose among nectar sources. *Behavioral Ecology and Sociobiology*, 28(4), 277–290. doi:10.1007/BF00175101. URL `http://dx.doi.org/10.1007/BF00175101`.

Siegert, F. and Weijer, C.J. (1992). Three-dimensional scroll waves organize *Dictyostelium slugs*. *PNAS*, 89(14), 6433–6437.

subCULTron (2015). Submarine cultures perform long-term robotic exploration of unconventional environmental niches. http://www.subcultron.eu/.

Tan, Y. and Zheng, Z.y. (2013). Research Advance in Swarm Robotics. *Defence Technology*, 9(1), 18–39. doi:10.1016/j.dt.2013.03.001. URL `http://www.sciencedirect.com/science/article/pii/S221491471300024X`.

Thenius, R., Moser, D., Varughese, J.C., Kernbach, S., Kuksin, I., Kernbach, O., Kuksina, E., Miškovi, N., Bogdan, S., Petrovi, T., Babi, A., Boyer, F., Lebastard, V., Bazeille, S., William Ferrari, G., Donati, E., Pelliccia, R., Romano, D., Jansen Van Vuuren, G., Stefanini, C., Morgantin, M., Campo, A., and Schmickl, T. (2016). subCULTron -Cultural Development as a Tool in Underwater Robotics Consortium for coordination of research activities concerning the Venice lagoon system. In *Artificial Life and Intelligent Agents*. Springer. In print.

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

Trianni, V., Groß, R., Labella, T.H., Sahin, E., and Dorigo, M. (2003). Evolving aggregation behaviors in a swarm of robots. In *ECAL*, 865–874. Springer.

Varughese, J.C., Thenius, R., Schmickl, T., and Wotawa, F. (2017). Quantification and analysis of the resilience of two swarm intelligent algorithms. In *GCAI 2017. 2nd Global Conference on Artificial Intelligence*, EPiC Series in Computing. EasyChair. In press.

Varughese, J.C., Thenius, R., Wotawa, F., and Schmickl, T. (2016). Fstaxis algorithm: Bio-inspired emergent gradient taxis. In *Proceedings of the Fifteenth International Conference on the Synthesis and Simulation of Living Systems*. MIT Press.