I. Troch, F. Breitenecker, eds.   ISBN 978-3-901608-35-3

Robustness of Two Interacting Robot Swarms Using the BEECLUST Algorithm

M. Bodi, R. Thenius, T. Schmickl, K. Crailsheim
Karl-Franzens University, Graz, Austria

Corresponding Author: M. Bodi, Karl-Franzens University Graz, Artificial Life Lab of the Department of Zoology, Universitätsplatz 2, 8010 Graz, Austria; michael.bodi@edu.uni-graz.at

Abstract. In this work we investigated how robust a robot swarm acts against disturbances caused by another robot swarm, both using the BEECLUST algorithm. We show that small swarm populations can gain benefit from the presence of other robot swarms. Medium populated swarms are affected neither positively nor negatively. Large swarm populations act robust against disturbances caused by other robot swarms as long as no jamming effects occur. For our investigation we simulated an environment with an ambient illuminance, a light spot and a shadow spot. In such an environment we tested two different castes of Jasmine III robots whereas each caste had to perform a different task. One swarm aggregates at places of high illuminance (light spot), the other one at places of low illuminance (shadow spot). In this article we show that the robustness of the BEECLUST algorithm allows us to control a heterogeneous robot swarm in environments which demand differing controller strategies.

1 Introduction

Swarms of autonomous entities (animals, robots) show interesting features: Huge numbers of agents act and interact in high densities and often perform their tasks in separated specialised cohorts. The regulation of the agents behaviours is often self-organised and swarm intelligent, features that are mainly achieved by various interlinked feedback loops within the swarm system. When creating artificial (e.g., robotic) swarm systems, these feedback networks can quickly get rather complex and swarm behaviours can easily become unpredictable. This generates the demand for finding simple behavioural rules for individuals and also simple rules of interactions. Such rules can often be found in natural swarm systems. This allows extraction of such regulation mechanisms and translation of these feedbacks to artificial swarm systems, which in turn show then comparable behaviour like the natural sources of inspiration. We recently demonstrated such a translation of natural algorithms to a swarm robotic algorithm by developing the BEECLUST algorithm [8]. In the following, we demonstrate how this algorithm scales with swarm sizes and how cooperation can emerge (without being pre-programmed) between several interacting robot swarms:

Compared to single robot systems, robot swarms have the advantage to work quite robust, because the removal or malfunction of single individuals affects the behaviour of the whole swarm only slightly. Robot swarms are also able to act “swarm intelligent” which permits a group of individuals to make decisions to reach a common goal in a decentralised manner [7].

In recent works it has been shown that a swarm of Jasmine III robots using the BEECLUST algorithm, which is inspired by honeybee behaviour, is able to find locations of maximum illumination using a few simple rules [8] (see Figure 1):

1. Robots move straight through the arena. Whenever a robot detects an obstacle, it stops and checks if the obstacle is another robot or a wall by listening for emitted IR signals.
2. If the detected obstacle does not emit IR signals it is a wall. The robot turns randomly and continues with step 1.
3. If the detected obstacle does emit IR signals it is another robot. The robot measures the local illuminance.
4. Depending on the local illuminance, the robot calculates a waiting time: the higher the local illuminance, the longer the waiting time. After the waiting time is over, the robot turns randomly and continues with step 1.

Our recent studies showed [7], that the BEECLUST algorithm allows a robot swarm to aggregate collectively at a light spot. This is achieved by robots that move purely random, having just one illuminance sensor that does not allow a directed gradient exploitation. Also no memory of past illuminance values is involved in the algorithm. The measurements of the environment occur only seldom, only after robot-to-robot collisions. Although the individual "intelligence" of single robots is kept quite low, the swarm as a total was found to act very intelligently, even in dynamic environments. The reasons for this emergent ability are the feedback loops that arise within the robot swarm, as is explained in detail in [4].

Based on the work of [8] we investigated how the robustness of the BEECLUST algorithm is affected by the swarms population and how it is influenced by a second group of robots in the arena performing a different task at the same time.

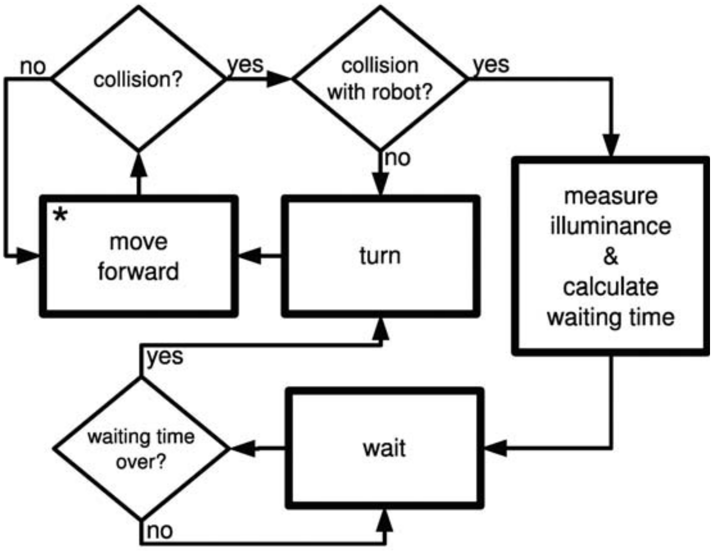

Figure 1. Finite state machine of the BEECLUST controller. Boxes represent the different behavioural states of a robot. * indicates the starting point. Diamonds represent control structures (if-else decisions).

2 Methods

For our experiments we used SMARS (Small Robot Simulator), a simulation environment programmed in NetLogo [5], which was built by us for performing experiments with simulated Jasmine III robots (see Figure 2). In this simulation environment we were able to simulate two different castes of Jasmine III robots which were implemented with the physical properties of the real robots. The only difference between the two castes is that one caste waits longer at places of high illuminance ("light finders"), the other one at places of low illuminance ("shadow finders"). There is a direct proportion between the illuminance and the sensor value whereas the maximal sensor value (180) is reached at 1000 lux and above.

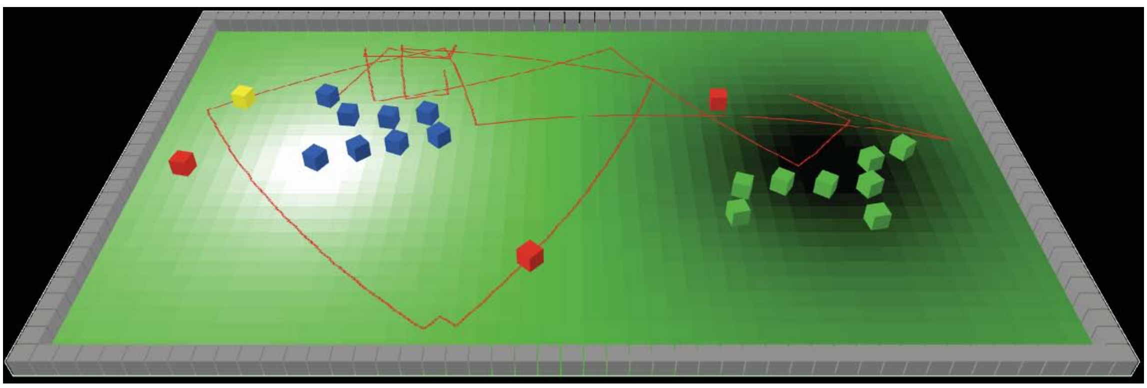

Figure 2. Screenshot of two robot swarms acting in the simulation environment SMARS. The white spot is the optimum for the "light finders" with 1000 lux and the black spot is the optimum for the "shadow finders" with 0 lux. The ambient illuminance is 500 lux. Robots are represented by colored boxes which represent their caste and state: blue boxes are "light finders" in waiting state, red boxes are "light finders" in driving state, green boxes are "shadow finders" in waiting state and yellow boxes are "shadow finders" in driving state. The red line shows a trajectory of a randomly chosen robot for demonstration purposes.

In our model sensor values are a linear function of local illuminance, as it is explained by Equation 1:

$$e_{x,y} = \begin{cases} E_{x,y}\theta, & if\, e_{x,y} \leq E_{max} \\ e_{max}, & else \end{cases} , \tag{1}$$

whereby $e_{x,y}$ represents the sensor value of the robots light sensor at the position x,y in the arena and e_{max} the maximum sensor value of a robot. $E_{x,y}$ is modelled as the illuminance at the position x,y and E_{max} as the maximal perceived light value. θ refers to the translation coefficient (for details see [9]). The waiting time of the "light finders" is modelled as

$$w^L_{x,y} = \frac{w_{max}}{1+\exp^{-(e_{x,y}+o)\sigma}} , \tag{2}$$

for "shadow finders" it is modelled as

$$w^S_{x,y} = \frac{w_{max}}{1+\exp^{-(e_{max}+e_{x,y}+o)\sigma}} , \tag{3}$$

whereby $w^L_{x,y}$ refers to the waiting time at the position *x,y* for "light finders" and $w^S_{x,y}$ for "shadow finders". w_{max} is modelled as the maximum waiting time, $e_{x,y}$ is the robots actual sensor value, e_{max} is the maximum sensor value, *o* is the vertical offset and σ is the steepness. Equation 2 and Equation 3 are both illustrated in Figure 3 The given values for the constants can be seen in Table 1.

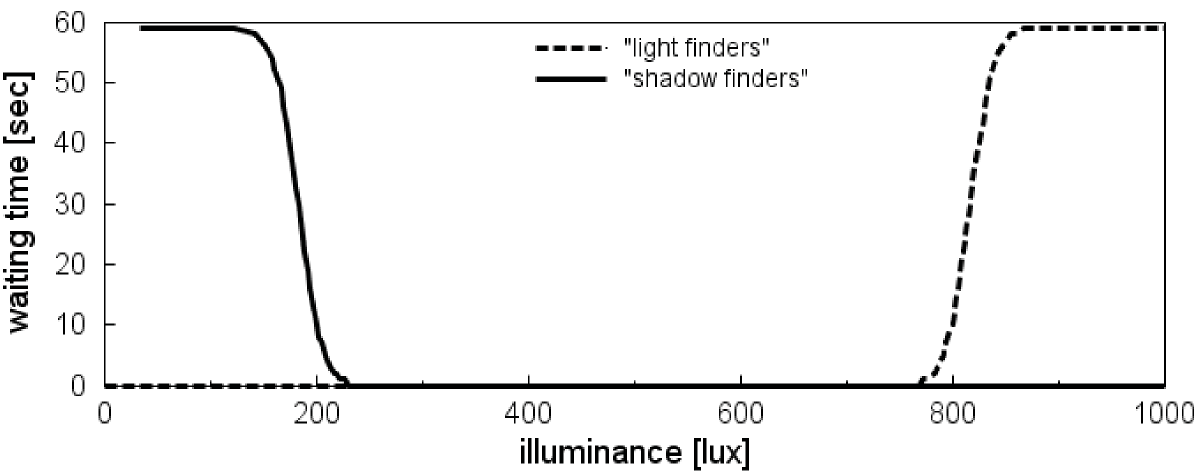

Figure 3. Illustration of the dependence between the local illuminance $e_{x,y}$ and the waiting time $w_{x,y}$. The solid drawn function was implemented in the "light finders", the dashed drawn function was implemented in the "shadow finders".

Constants	Values	Dimensions
w_{max}	60	sec
e_{max}	180	dimensionless
E_{max}	1000	lux
θ	0.18	dimensionless
o	147	dimensionless
σ	0.5	dimensionless

Table 1. Constants and their values used in Equations 1, 2 and 3, in our simulation runs presented in this article .

We generated an environment with an ambient illuminance of 500 lux. In this environment we added two different kinds of optima, a light-spot with 1000 lux and a shadow-spot with 0 lux (Figure 4). We defined a target-zone for the "light finder" caste consisting of all patches having a light value between 600 lux and 1000 lux. This is 40% of the maximum light value in the arena. As mentioned above each robot which meets another robot calculates the waiting time individually, depending on the local illuminance but regardless of the other robots caste. To find the optimal density of a Jasmine III swarm we first tested groups of differing population sizes only consisting of "light finders". For each swarm population we repeated the experiment 6 times for 4 minutes. To get an overview of the final conditions of our robot swarm, we monitored the fraction of the "light finders" in the state "wait" located in the target-zone in the last minute of every repetition. To analyse the aggregation speed we monitored the point of time when 50% of the "light finder" population aggregated in the target zone (TA_{50}). To test the robustness of a robot swarm against disturbances caused by another swarm performing another task we added groups of variable numbers of "shadow finders" and observed how they affect the aggregation behaviour of the "light finders".

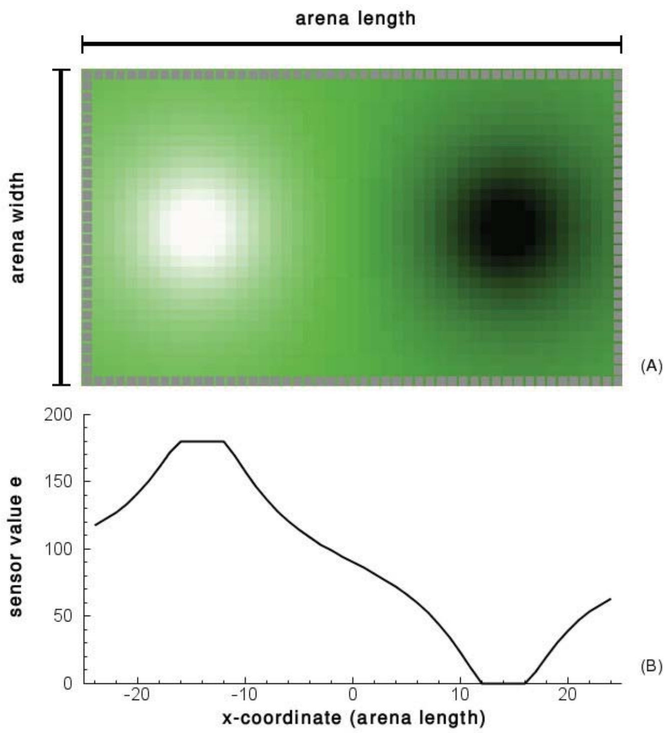

Figure 4. (A) Overhead screenshot of the arena in SMARS. The bright area indicates the light spot, the dark area indicates the shadow spot. (B) Sensor values detected by the robots. Samples are taken from patches with y = 0, which is the vertical centre of the arena.

3 Results

Our results show that there is an optimal density of the tested robot swarms for the used environmental conditions. This optimal density was found at a swarm population of 9 individuals (see Figure 5). A swarm consisting of 9 individuals is able to place 80% of its individuals in the target zone.

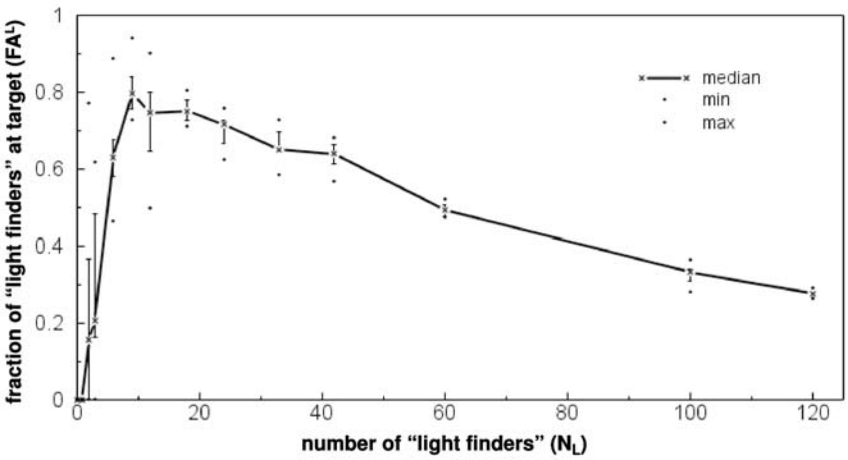

Figure 5. Fraction of robots in target zone over the last minute of observation. The optimal swarm population for the given environment was found at 9 robots. The tested swarm populations were 2, 3, 6, 9, 12, 18, 24, 33, 42, 60, 100, 120 individuals, n = 6 repetitions per experiment.

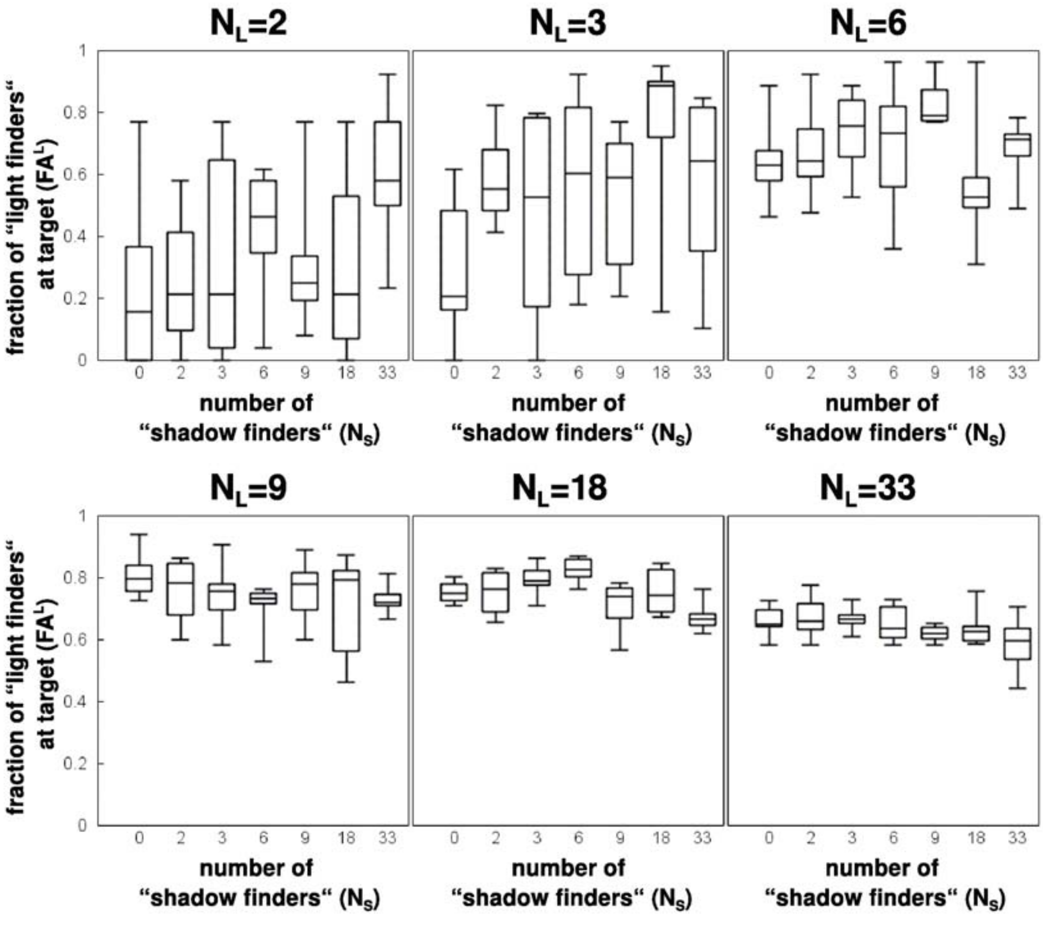

Figure 6. Fractions of "light finders" aggregated in the target zone within the last minute of observation (median ± quartiles, min and max). N_L is the total number of "light finders", n = 6 repetitions per experiment.

In Figure 6 we demonstrate that in small robot populations (N_L = 2, N_L = 3) the fraction of aggregating robots under the light source increases with the number of other robots in the arena. Medium populated robot swarms (N_L = 6 to N_L = 18) have a constant fraction of robots aggregated under the light source, independent of the number of other robots in the arena. Larger swarms (N_L = 33) show a decreasing fraction of aggregated robots with increasing number of other robots in the arena.

Concerning the aggregation speed we show that small robot populations (N_L = 2 to N_L = 3) do no aggregate under the light source as long as no other robots (N_S = 0) are present in the arena (Figure 7). The aggregation speed increases with the number of other robots (N_S > 3). Medium populated robot swarms (N_L = 6 to N_L = 18) aggregate with the same speed under the light source, independent of the number of other robots in the arena. Larger robot swarms (N_L > 18) show a decreasing aggregation speed with increasing number of other robots in the arena.

To allow us investigations of the observed enhancements of aggregation, which the focal robot swarm can draw from the presence of the other swarm, we developed two numerical expressions of aggregation enhancement: ΔFA^L, which expresses the increase of the aggregation fractions in the presence of 33 "shadow finders" compared to runs were "light finders" acted alone (see Equation 4). In addition ΔTA_{50} expresses the increase in the speed of aggregation when such simulation runs were compared (see Equation 5).

$$\Delta FA^L = FA^{(N_S=33)} - FA^{(N_S=0)} \tag{4}$$

Figure 8, where ΔFA^L was evaluated for different swarm sizes of "light finders", shows that small populations show an enhancement of their ability to aggregate by increasing number of other robots, whereas larger populations are not affected, whereby ΔFA^L represents the difference between the fraction of "light finders" in the target zone at a "shadow finder" population of 33 $FA^{(N_S=33)}$ and a "shadow finder" population of 0 $FA^{(N_S=0)}$.

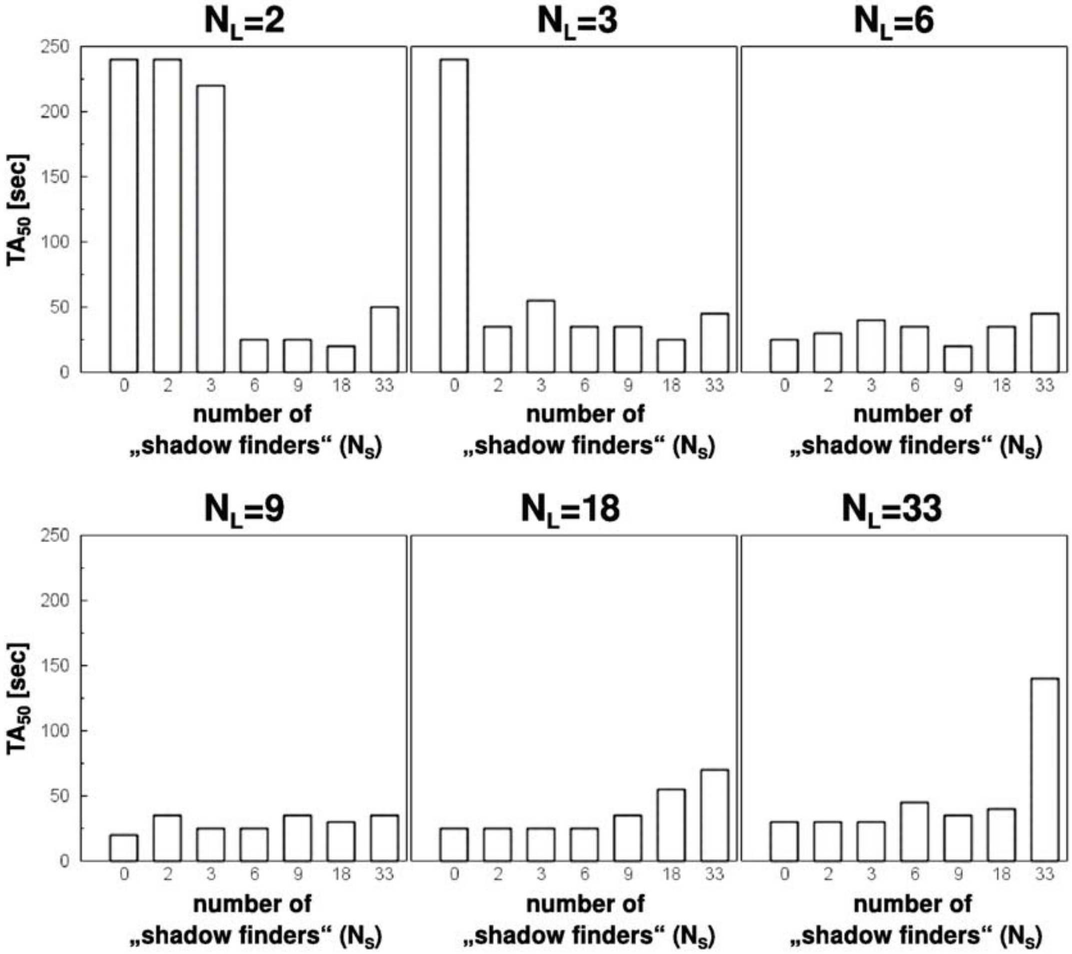

Figure 7. Time in which 50 % of the "light finder" swarm aggregated in the target zone (TA_{50}) at different population sizes of "shadow-finders" (N_S). N_L is the number of "light finders", n = 6 repetitions per experiment.

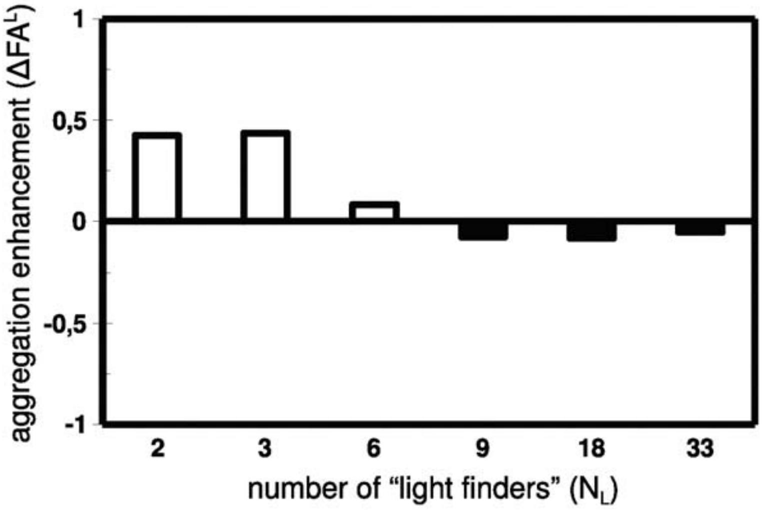

Figure 8. Shown is the enhancement of aggregation ΔFA^{L} in seconds which is the difference of fraction between $N_S = 0$ and $N_S = 33$, for $N_L = 2$ to $N_L = 33$. For details on the formulation of ΔFA^{L}, please see Equation 4.

To investigate also the advancements in aggregation speed, which are caused by the presence of the other robot swarm, we defined an aggregation enhancement ΔTA_{50} according to

$$\Delta TA_{50} = TA_{50}^{(N_S=0)} - TA_{50}^{(N_S=33)} \quad , \tag{5}$$

whereby $TA_{50}^{(N_s=0)}$ is the point of time when 50% of the "light finders" aggregate under the light source by a population of 0 "shadow finders" and $TA_{50}^{(N_s=33)}$ by a population of 33 "shadow finders". In Figure 8 we show that small populations of "light finders" aggregate faster by rising number of "shadow finders." Medium populated swarms only take small disadvantage by a increasing numbers of "shadow finders" and large "light finder" populations take strong disadvantage.

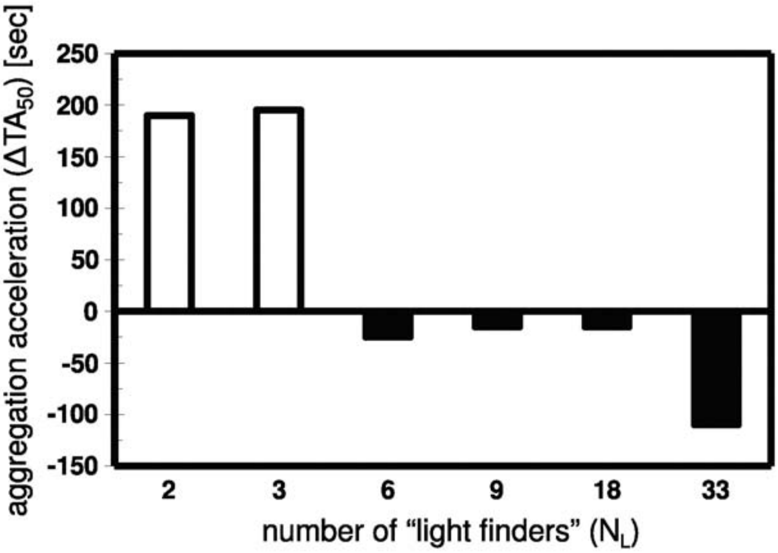

Figure 9. Shown is the acceleration of aggregation ΔTA_{50} in seconds which is the difference of TA_{50} between $N_S = 0$ and $N_S = 33$, for $N_L = 2$ to $N_L = 33$. For details on the formulation of ΔTA_{50}, please see Equation 5.

4 Discussion

As shown in Figure 5, a robot swarm using the BEECLUST algorithm has an optimal swarm population of 9 individuals for the tested arena size and illuminance patterns. First of all the swarm of “light finders” gains efficiency from a higher swarm population, until jamming and collisions affect the aggregation efficiency adversely. We expect a different finding in a diverse dimensioned environment because the optimal population size is dependent on the robot density.

Concerning the robustness of the swarm, we found that small “light finder” populations gain benefit from increasing numbers of “shadow finders” in their ability to aggregate as well as in their aggregation speed. This results from a higher total number of robots in the arena, more collisions happen and lead to an increasing frequency of light measurements. Medium populated swarms act robust in presence of another robot swarm concerning their ability to aggregate as well as concerning their aggregation speed. Both of these observables are affected only slightly. The advantage of more measurements and the disadvantage of jamming cancel each other out. Large swarms take disadvantage from a rising number of other robots, because the robot density gets too high and jamming effects occur. The appearance of advantages in aggregation caused by other robots in the arena is caused by the increase of robot collisions which go along with higher robot density in the arena. This leads to an increasing number of light measurements due to a rising number of robot collisions. But if the density of robots in the arena is too high, jamming effects occur. This leads to a lower quantity of aggregation (Figure 6 and Figure 8) and lower aggregation speed (Figure 7 and Figure 9).

With the work at hand we show that the BEECLUST algorithm is not only capable of working robust in a homogeneous robot swarm, but it is also suitable in heterogeneous robot groups with differing controller strategies, even without discrimination of caste affiliation of other robots. Those gained benefits allow us to use the BEECLUST algorithm in environments which may demand heterogeneous controller strategies, e.g. tanking up energy: Robots with a lack of energy aggregate at the fuelling station, whereas robots with enough energy accomplish their mission.

Bio-inspired algorithms, like our honeybee-inspired BEECLUST algorithm, are frequently investigated in the field of autonomous robotics in recent years and are combined with a big variety of techniques: K. Sims showed the first simulation of virtual creatures which were able to evolve animal-like behaviours and morphologies[9]. Marocco et al. evolved various robot populations [6], which developed several communication schemes by a process of artificial evolution. Other studies experimented with robots which were able to integrate in insect societies [3]. Also the field of bio-inspired control algorithms for large populated robot swarms has been investigated intensively: In the I-Swarm project [10] techniques for controlling a swarm of 1000 microrobots were investigated. The Swarmbot-project [1, 12] investigated the self-organising abilities of a robot swarm, consisting of simpler, insect-like robots. In the Symbrion-project [13] the concept of a swarm of reconfigurable robots is investigated. The Replicator-project [11] plans to develop an autonomous robot swarm for industrial purposes.

The topic of aggregation of swarms of agents is a common benchmark in the field of swarm robotics: Dorigo et al. used an evolving neural network consisting of 12 neurons for robot aggregation [1]. Compared to this work the BEECLUST algorithm is quite simple, because it gets along with just a few basic rules and therefore it needs less processing power and memory. Other experiments with cockroach-like robots have been performed in simulations as well as in real world [2]. The main difference to our work presented here is the ability of cockroach-like robots to measure the number of other nearby robots. This requires a unique ID and communication between the individuals. Our robots, which use the BEECLUST algorithm, neither need to discriminate between several robots nor to know how many individuals are located in a nearby aggregation spot.

In summary, the BEECLUST algorithm is simple and therefore it needs little capacity. It does not use any communication in a biological sense. The robots are able to distinguish other robots from obstacles in the environment via their physical properties and characteristics, but they do not need to know “which” and how many other robots are around. Above all, our study presented here shows how robust the BEECLUST algorithm works and how versatile its applicability is despite its simplicity.

In future we will investigate how the distribution of light in the arena (gradient or discrete) influences the aggregation behaviour. Furthermore we will investigate how the distance between the two optima influences the aggregation behaviour.

5 Acknowledgements

This work was supported by: EU-IST FET project ’I-Swarm’, no. 507006; EU-IST-FET project ’SYMBRION’, no. 216342; EU-ICT project ”REPLICATOR”, no. 216240. Austrian Science Fund (FWF) research grants: P15961-B06 and P19478-B16. Much appreciation for M. Szopek for her creative input.

6 References

[1] Dorigo, M., Trianni, V., Groß, R., Labella, T.H., Baldassarre, G., Nolfi, S., Deneubourg, J.L., Mondada, F., Floreano, D., Gambardella, L.M.: *Evolving Self-Organizing Behaviors for a Swarm-bot.* Autonomous Robots, 17, 2004, 223 – 245.

[2] Garnier, S., Jost, C., Jeanson, R., Gautrais, J., Asadpour, M., Caprari, G., Theraulaz, G.: *Collective decision-making by a group of cockroach-like robots.* In: Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE, 233 – 240.

[3] Halloy, J., Sempo, G., Caprari, G., Rivault, C., Asadpour, M., Tache, F., Said, I., Durier, V., Canonge, S., Ame, J.M., Detrain, C., Correl, N., Martinonli, A., Mondana, F., Siegwart, R., Deneubourg, J.L.: *Social Integration of Robots into Groups of Cockroaches to Control Self-Organized Choice.* Science, vol. 318, no. 5853, 2007, 1155 – 1158.

[4] Hamann, H., Wörn, H., Crailsheim, K., & Schmickl, T.: *Spatial macroscopic models of a bio- inspired robotic swarm algorithm.* In: Proceedings of the IEEE/RSJ 2008 International Conference on Intelligent Robots and Systems, 2008, 1415 – 1420.

[5] http://ccl.northwestern.edu/netlogo Wilensky, U.: *NetLogo.* Center for Connected Learning and Computer-Based Modeling. Northwestern University, Evanston, IL. Accessed December 2008.

[6] Marocco, D., Nolfi, S.: *Origins of Communication in Evolving Robots*. SAB 2006, LNAI 4095, 2006, 789 – 803.

[7] Millonas, M. M.: *Swarms, phase transitions, and collective intelligence*. Artificial Life, 3, 1994, 417–445.

[8] Schmickl, T., Thenius, R., Moeslinger, C., Radspieler, G., Kernbach, S., Szymanski, M., Crailsheim, K.: *Get in touch: cooperative decision making based on robot-to-robot collision*s. Autonomous Agents and Multi-Agent Systems, vol. 18, 2009, 133 – 155.

[9] Sims, K.: *Evolving Virtual Creatures.* Computer Graphics (Siggraph '94 Proceedings), 1994, 15 – 22.

[10] Valdastri, P., Corradi, P., Menciass, A., Schmickl, T., Crailsheim, K., Seyfried, S., Dario, P.: *Micromanipulation, Communication and Swarm Intelligence Issues in a Swarm.* Microrobotic Platform. Robotics and Autonomous Systems 54, 2006, 789 – 804.

[11] www.replicators.eu/ REPLICATOR project website, Accessed December 2008.

[12] www.swarmrobot.org/ Accessed December 2008.

[13] www.symbrion.eu/ SYMBRION project website, Accessed December 2008.