Revisiting BEECLUST: Aggregation of Swarm Robots with Adaptiveness to Different Light Settings

Mostafa Wahby
Heinz Nixdorf Institute
Dep. of Computer Science
University of Paderborn,
Paderborn, Germany
mostafa.wahby@uni-paderborn.de

Alexander Weinhold
Heinz Nixdorf Institute
Dep. of Computer Science
University of Paderborn,
Paderborn, Germany
gutmet@mail.uni-paderborn.de

Heiko Hamann
Heinz Nixdorf Institute
Dep. of Computer Science
University of Paderborn,
Paderborn, Germany
heiko.hamann@uni-paderborn.de

ABSTRACT

Aggregation is a crucial task in swarm robotics to ensure cooperation. We investigate the task of aggregation on an area specified indirectly by certain environmental features, here it is a light distribution. We extend the original BEECLUST algorithm, that implements an aggregation behavior, to an adaptive variant that automatically adapts to any light conditions. We compare these two control algorithms in a number of swarm robot experiments with different light conditions. The improved, adaptive variant is found to be significantly better in the tested setup.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: intelligent agents, multiagent systems

General Terms

Algorithms, Performance, Theory

Keywords

beeclust, adaptive behavior, collective decision making, aggregation, swarm robots

1. INTRODUCTION

A crucial task in swarm robotics [3, 7, 4] is to aggregate the swarm [25, 22, 20, 21]. Staying together is essential to allow for cooperation and interaction in robot swarms. While sometimes it might be enough to aggregate anywhere, in other cases we want the robot swarm to aggregate at either a directly designated area or an area specified indirectly by certain features. For example, a robot swarm may need to collectively agree on an appropriate construction site before starting construction [23, 5, 27] or the swarm may need to aggregate on an area of a desired temperature as observed in young honeybees [24]. We call the task of aggregating swarm robots on an area of a desired feature 'adaptive aggregation'. In addition, the environment could be dynamic in which case we want the robot swarm to stay adaptive to changing conditions. If a new area of high quality emerges, then the swarm should disperse and aggregate again at that new area. Also note that adaptive aggregation can be interpreted as a collective decision-making problem because all swarm members have to find a consensus on where to aggregate. A control algorithm for the task of adaptive aggregation is BEECLUST [17, 19, 13]. Swarm systems controlled by BEECLUST have been analyzed in depth in a variety of models [18, 11, 9, 26, 6]. Also extensions based on fuzzy logic have been proposed [2, 1].

In the following we investigate the adaptive aggregation task in a swarm of simple, mobile robots. The desired aggregation area is indicated by light and the swarm is asked to aggregate at the brightest spot. We also investigate a light setting that has a bright light and a dimmed light at the same time. Such a situation can be interpreted in terms of an optimization problem, that is, we have a local and a global optimum. Then the swarm has to create a global awareness to avoid a greedy local aggregation because we want the swarm not to split.

There are two main challenges in adaptive aggregation. First, robots require the capability of kin recognition [15], that is, they have to detect close-by robots and need to discriminate them from any other type of object in their environment. Implementing kin recognition can be a challenge depending on the available sensors of the applied robot platform. The second challenge is less obvious. Implementing a self-organizing aggregation behavior that relies only on local information requires a cluster formation behavior. Robots form new clusters, join existent clusters (exploitation), and leave clusters (exploration). Typically, that requires an appropriate choice of waiting times for how long to stay in a cluster in order to avoid deadlocks [12] and to achieve fast convergence of aggregating all robots in one cluster at the right spot. Fast convergence is achieved in a competition between clusters and growth of clusters is implemented by positive feedback. For example, robots in bigger clusters wait longer than robots in smaller clusters. Appropriate waiting times depend on the swarm size, area (i.e., swarm density), and here the properties of the light distribution. In the following we keep the swarm size constant (except for

BICT 2015, December 03-05, New York City, United States
Copyright © 2016 ICST
DOI 10.4108/eai.3-12-2015.2262877

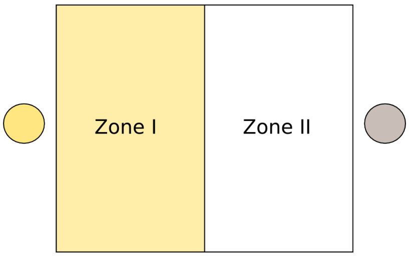


Figure 1: Division of the arena into two zones: left, zone I, bright light; right, zone II, dimmed or no light.

small changes by individual robot hardware failures). Hence, we are interested in adapting the robot behavior to different light settings only and keep adaption to dynamic swarm sizes/densities for future work. In robotics it is generally tempting to merely optimize a robot controller for an arbitrary, static environment—the robot lab. However, such a robot controller is not adaptive to changes and arguably makes use of global knowledge about the environment, instead of generating global awareness troughout the swarm about environmental conditions in a self-organizing process.

We implement the original BEECLUST on Thymio II robots, then we extend the original algorithm to an adaptive variant that automatically adapts to any light conditions. Then we investigate the impact of the introduced improvement on the performance of the algorithm in different light settings.

2. TASK: ADAPTIVE AGGREGATION

Aggregation is a standard task in swarm robotics and following Brambilla et al. [4] aggregation belongs to the class of 'spatially-organizing behaviors'. It creates a swarm configuration that is necessary for many other swarm behaviors, such as flocking, collective decision-making, and construction. While in general the aggregation task might not specify a particular meeting area, here we focus on a variant that we call adaptive aggregation. The idea is that the swarm aggregates on an area of a particular environmental quality. We assume that the robot has a sensor to measure this quality and that it knows whether this quality should be maximized or minimized. For example, the adaptive aggregation task could be to aggregate at the brightest spot within a bounded area, that is, the considered quality is light and it should be maximized. Besides aggregation at a spot of desired quality, we define three more requirements of adaptive aggregation. First, in an environment that has multiple spots of maximal or approximately maximal quality we require that the swarm is able to break that (approximate) symmetry [8]. Second, the environment is allowed to be dynamic. As a consequence the swarm has to stay adaptive to these environmental changes, to keep exploring the surrounding, and has to relocate the swarm to a new spot of maximum quality. Third, all of that should be done quickly with minimal time requirements.

The setup of the following experiments is inspired by previous work [10]. The robot arena is bounded and rectangular. At each of two opposite sides of the arena, a light source is fixed in centered position at a height of 107cm (see Figure 1). We test different light settings: one bright light (only one light source is active and is set to bright mode), one dimmed light (only one light source is active and is set to dimmed mode), one bright & one dimmed light. For later use we divide the arena logically into two zones: zone I always contains the brighter light source in the experiments and zone II is either the dimmed or dark zone (see Figure 1).

3. IMPROVED BEECLUST

The BEECLUST algorithm is inspired by observations and a behavioral model of young honeybees that navigate in temperature gradients within beehives [17]. Young honeybees prefer a temperature of 36°C and gather at such spots. Hence, the BEECLUST algorithm generates a clustering behavior in robot swarms as required by the above described task of adaptive aggregation. The original algorithm is defined as follows:

```
1. Each agent moves straight until it encounters
   an object within its sensor range.

2. If this obstacle is a wall (not another robot),
   it turns away and continues with step 1.

3. If the obstacle is another robot, the robot
   measures the local luminance.

4. The higher the luminance is, the longer the
   robot stays stationary at that position (i.e.,
   the robot switches to the stopped state).

5. After the waiting time is over, the robot turns
   away from the other robot and continues with
   step 1.
```

The waiting time used for step 4 is determined by a sigmoid function

$$w(I) = \frac{w_{\max} I^2}{I^2 + c}, \tag{1}$$

where $w_{\max}$ is the maximum waiting time in seconds, I is the measured luminance, and constant $c = 4.86 \times 10^5$ [17].

The algorithm makes use of a positive feedback mechanism. According to the waiting time function, see Equation (1), the higher the illumination is, the longer the robot stays stationary after having encountered another robot. Therefore, clusters formed at brighter spots persist for longer than those at darker spots. Hence, clusters positioned at brighter spots have a higher chance of growing into bigger clusters. Also, bigger clusters cover bigger areas, which increases the probability of being encountered by a moving robot. Finally, robots might get trapped within big clusters being surrounded by other stopped robots. Consequently, big clusters have higher growth rates than small clusters (i.e., positive feedback). In conclusion, this process results in one big cluster typically positioned at a bright spot.

Previous works have reported good performance of this algorithm [17]. However, only if the parameters ($w_{\max}$ and c) were chosen correctly which was usually done in preliminary experiments. If the parameters are not well chosen for a given light distribution, then the waiting time in the brightest spots might not be long enough to complete the task. Hence, there is the option of a straightforward improvement that goes beyond the pre-determined relation between the waiting time and illumination. A simple approach could be to set the maximal waiting time $w_{\max}$ to a big value such that for any light distribution robots wait long enough to form clusters. In addition, a translation and scaling of the waiting time function is necessary to prevent clusters to form at places with low ambient light and to allow clustering at relevant brightly illuminated areas. However, this requires to include global knowledge about the experiment setup that was acquired by a human observer beforehand (e.g., maximum and minimum light intensities). Therefore, we want the robots to translate and scale the waiting time function themselves by determining a minimum light intensity $I_{\min}$ and maximum light intensity $I_{\max}$. These are then used to calculate a rescaled light intensity $\bar{I}$ of arbitrary range from 0 to 1500 that adapts the waiting times accordingly. We get

$$\bar{I} = (I - I_{\min})\frac{1500}{I_{\max} - I_{\min}}. \quad (2)$$

We use $\bar{I}$ instead of I in eq. (1) and hence the waiting time function is adapted to the particular conditions of the experiment. $I_{\min}$ is defined as baseline illumination during the experiment, that is, the minimal illumination intensity at maximal distance from all light sources. $I_{\max}$ is defined by the illumination directly below the brightest light source. If the intensity drops steeply (the sigmoid function is bounding the function values for big $\bar{I}$ anyways), the maximum illumination $I_{\max}$ is diminished to get better results (done by averaging, see Sec. 4).

In this work, we present a new approach which does not require manually adapting the waiting time function for the experiment setup. Instead, we introduced a calibration round which allows for an automatic adaption. In this calibration round, each robot drives around the arena, measuring the illumination in regular intervals, and determines $I_{\min}$ and $I_{\max}$ individually. For the used arena dimensions (3m×2.5m), a calibration time of three minutes is sufficient for random exploration to cover big parts of the arena. Once the calibration phase is done, the actual aggregation phase starts. In the next section, we present our experiment setup, the capabilities of the Thymio II robot hardware, and the implemented necessary hardware extensions.

4. EXPERIMENT SETUP AND THE THYMIO II ROBOT

The robots move in a rectangular arena surrounded by a wall. The size of the arena is 3m×2.5m and is inspired by former experiments [19]. As light sources we use dimmable, traditional filament lamps with a power consumption of up to 100 watts. The bright light mode is set to 100% illumination, while the dimmed light mode is set to around 20%. The light distribution in the arena is measured by manually placing the robot and saving the illumination values on a straight line between the center of the bright and dimmed spots in the arena, as shown in Figure 3. The adaptive waiting time function (see Equations 1 and 2) maps the luminance distribution in the arena into suitable waiting times in these different light settings (see Figure 4). In our experiments a maximum waiting time ($w_{\max}$) of 50 seconds is used.

The Thymio II robot (see Figure 5) measures approximately 11cm×11cm×5cm [16]. It is designed as open hardware

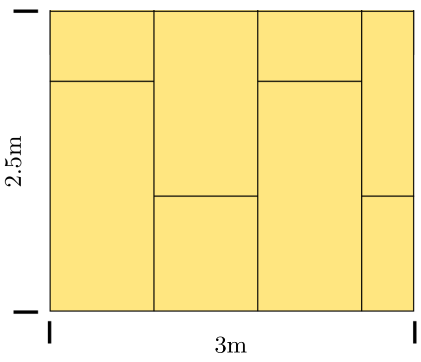


Figure 2: Dimensions of the robot arena.

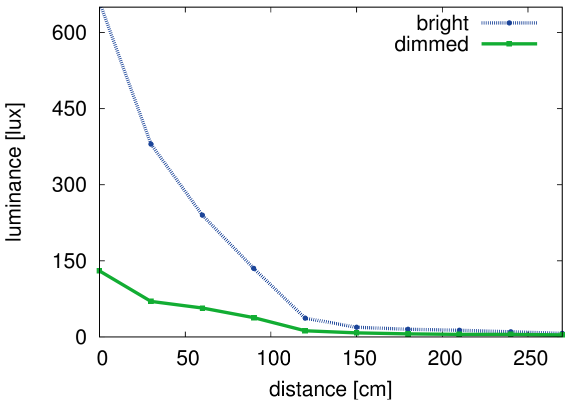


Figure 3: Distribution of light along a centered line from bright light to other side.

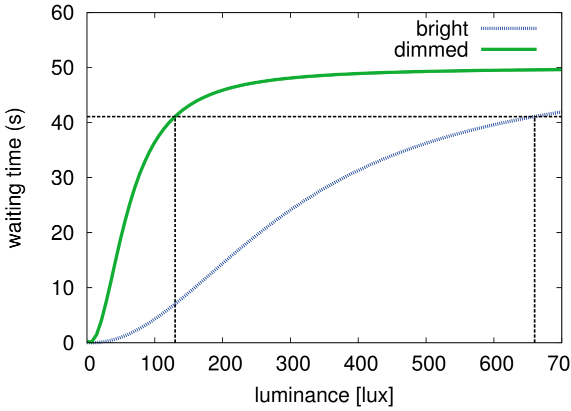


Figure 4: Mapping of luminance values to waiting time (equations 1 and 2) for two light settings: bright and dimmed (black lines indicate maximal luminance values and $w_{\max}$ is 50 seconds).

and open source software for educational purposes. The robot features two actuators and many sensors, such as two DC motors for differential drive; seven IR sensors (five at the front and two at the back), which are used for communication and proximity sensing; two IR sensors directed to the ground as ground sensors; an RC-5 capable IR sensor, which allows controlling the robot by a remote control; a 3-axes accelerometer, which provides the values of the acceleration along the three axes and could also be used to detect collisions; a temperature sensor; several LEDs, which are associated with the sensors and can highlight their activations. Moreover, Thymio II robots give the opportunity of installing extensions by offering attachment knobs for Lego$^{\mathrm{TM}}$ bricks on top and sockets for Lego Technic rods on the wheels.

Thymio II robots use the seven horizontal IR sensors for communication. In our experiments, this feature is used for kin recognition. After the calibration round each robot constantly tries to communicate by a pulsed signal which allows a robot to discriminate the wall from a robot. The robots can communicate up to a maximum distance of about 25cm, as long as at least one of the seven horizontal IR sensors is within the line of sight of a horizontal sensor of the other robot. However, the seven IR sensors do not provide circumferential view such that angular displacement decreases the communication range dramatically. The sides of the Thymio II have no IR sensors (see Figure 5), leaving large blind spots for communication. Therefore, robots fail to recognize their kin sometimes (e.g., when encountering the side of another robot), that is, they classify a closeby robot as wall. We addressed this problem by letting the robots rotate while in the 'stopped state' (i.e., robots rotate but do not cover any distance), which vastly increases the chances of kin recognition.

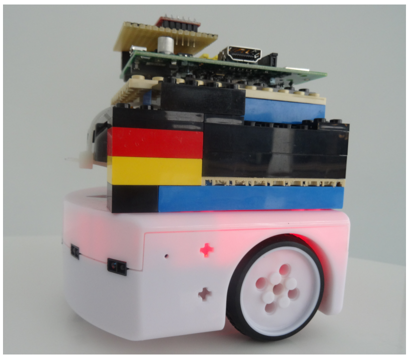


Figure 5: Thymio II robot with a Raspberry Pi and an ambient light sensor attached on top.

Off-the-shelf Thymio II robots are not equipped with any sensors for visible light. A hardware extension including a Raspberry Pi and the TSL45315 ambient light sensor was designed and attached to the robots (see Figure 5). The TSL45315 provides direct output in lux with wide dynamic range (from 3 lux to 2.2×10^5 lux), and is interfaced with the Raspberry Pi through the I^{2}C bus. The Raspberry Pi, which is considered here as the main processing unit, is connected to the Thymio II robot through the USB port and the D-Bus interface. In our setting, the Thymio II robot is treated only as a sensor/actuator unit that reacts to global events sent by the Raspberry Pi and passes on events raised by the Aseba VM[1] [14].

We exploited the RC-5 capabilities of the Thymio II robots by using a universal remote control to start/stop the main program on the Raspberry Pi (i.e., to start an experiment). The robots are initially placed in an evenly spaced grid, before starting an experiment. When the main program starts, the Raspberry Pi instructs the robot to take a random turn then, in the case of running the improved BEECLUST, starts the calibration period which lasts for three minutes. During the calibration phase the light intensity is measured every two seconds (90 samples in total) and stored. The average of the five lowest measurements defines the minimum light intensity $I_{\min}$ and the average of the five highest measurements defines the maximum intensity $I_{\max}$ (thereby taking into account the comparatively steep intensity dropoff of the light distribution). Then the algorithm is executed for 20 minutes. In the case of the original BEECLUST algorithm, the calibration phase is left out and the robot directly starts the BEECLUST algorithm. When two robots approach each other and the kin recognition is successful then the Aseba VM raises an appropriate event which is passed to the Raspberry Pi. The Raspberry Pi measures the sensor values, calculates the waiting time according to equation (1), then issues commands for the Thymio II robot to stop, spin, and later turn, and move straight. The improved BEECLUST algorithm for the Thymio II robot is defined as shown in Figure 6.

For each of the previously mentioned three light settings (one bright light, one dimmed light, and one bright & one dimmed light) a set of five experiments was conducted for each algorithm implementation (the original and improved BEECLUST). We investigate the adaptivity to light conditions and of the improved BEECLUST algorithm and its performance by comparing it to the original BEECLUST. In the case of the original BEECLUST, the minimum light intensity $I_{\min}$ and the maximum intensity $I_{\max}$ were manually set to appropriate values of the bright light setting only (see Figure 3). That way we are able to show that, on the one hand, the original BEECLUST algorithm works well once correctly parameterized (here in the bright light setting). On the other hand, however, it performs poorly compared to the improved BEECLUST if it is not well parameterized (here in the dimmed light setting). We investigate a robot swarm of size nine. The performance of an algorithm is evaluated by monitoring the number of robots in stopped state in zone I during a ten-minute window of the experiment. The philosophy of the BEECLUST algorithm is to stay adaptive to changes in the environment (on- and off-turning lights). As a consequence it does not converge to a state where all robots are stopped. Instead there are always a few robots exploring the arena. Therefore, we measure the performance during an interval of ten minutes once the influence of the initial position of robots is negligible. In the next section, we report results of the previously mentioned experiments, followed by a quick discussion and conclusion.

[1] Aseba VM is a virtual machine running on Thymio II robot, which handles the low-level control of the sensors and actuators and provides a high-level abstraction for programs which run on this virtual machine.

1. Each robot waits for a start signal from the remote control.
2. When a start signal is received, each robot takes a random turn and starts a calibration phase for three minutes.
3. During the calibration phase each robot moves straight while taking light intensity measurements every two seconds.
4. If the robot encounters any obstacle (robot or wall), it turns away and continues with step 3.
5. When the calibration phase ends, the aggregation phase starts and the robots start to try to communicate by pulsed signals.
6. Each robot moves straight until it detects an object within its sensors range.
7. If a communication signal is not received, it means that the robot has encountered a wall and consequently, it immediately turns away and continues with step 6.
8. If a communication signal is received, the robot measures the light intensity then starts spinning around its axis to announce its presence (increasing chances for kin recognition).
9. The higher the illumination is, the longer the robot stays stationary.
10. After the waiting time is over, the robot u-turns and continues with step 6.

Figure 6: Improved BEECLUST swarm control algorithm for the Thymio II robot.

5. RESULTS

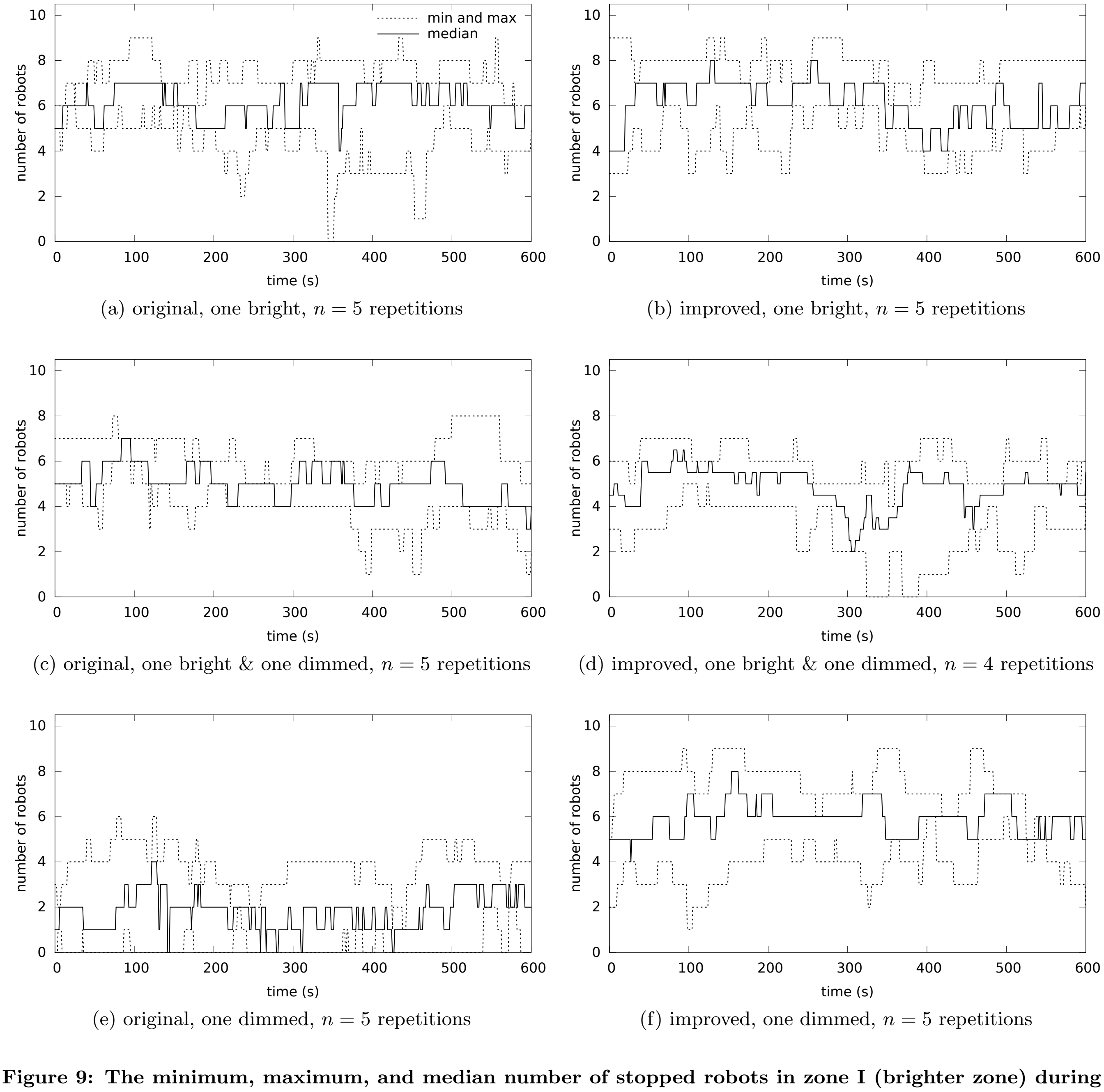
In Figure 9 we show the results of the robot experiments for all six experiment settings (one bright light, one bright & one dimmed light, one dimmed light) and each with the original BEECLUST parameterized for the bright light setting and the improved BEECLUST algorithm that calibrates automatically. Each diagram shows the minimum, maximum, and median number of robots in stopped state in zone I out of all n repetitions with $n \in \{4, 5, 6\}$ depending on the experiment setting.

Considering the original BEECLUST experiments in general, the robots have shown successful clustering behavior towards the brightest spot (an example of good clustering behavior from our experiments can be observed in Figure 7a). Specifically, in the one bright light and one bright & one dimmed light experiments (see Figures 9a and 9c). For the one bright light experiments, we analyze the maximal cluster sizes of aggregated robots in stopped state in zone I (i.e., the brighter zone) that were observed during the whole time interval for each repetition. These maxima range between cluster sizes of seven and nine robots indicating effective aggregation. The one bright & one dimmed light experiments have shown slightly lower performance, with maxima of cluster sizes ranging from six to eight. An example is shown in Figure 7b, where a cluster of seven robots is formed in zone I and a cluster of two robots is formed in zone II. In contrast, the robots have shown an expected (according to previous analysis in [17]) poor clustering behavior in the one dimmed light experiments, where the maxima of cluster sizes range from only four to six robots (see Figure 9e). An example of this poor clustering behavior is shown in Figure 7c, where only four robots form a cluster in zone I. As mentioned in the previous section, the waiting function was not manually re-adapted to the different light distributions in the one bright & one dimmed light and the one dimmed light setting. Therefore, in the one dimmed light experiments the maximum illumination was configured to a value 80% higher than the actual maximum. That results in too short waiting times at the brightest spot in the arena and a poor clustering behavior. A scouting behavior of some robots was observed during the experiments, where one or more robots would leave the cluster then move around randomly. This behavior is useful, as it allows for the detection of changes in the environment (e.g., emergence of a brighter area).

Regarding the improved BEECLUST experiments, the robots show a relatively similar clustering behavior as in the original BEECLUST implementation for the one bright light and one bright & one dimmed light experiments (see Figures 9b and 9d). Starting with the one bright light experiments, the maximal cluster sizes observed during each of the repetions range from eight to nine stopped robots in zone I. Unfortunately, the total number of robots in the one bright & one dimmed light experiments is only eight due to a robot hardware failure. That should be taken into account when comparing the absolute cluster sizes to those achieved by the original BEECLUST algorithm. However, the performance still is competitive with maximal cluster sizes of seven during the repetitions. Finally, in the one dimmed light experiments, maximal cluster sizes range from seven to nine robots (see Figure 9f). The one dimmed light experiment is crucial to determine whether the automatic calibration of the improved BEECLUST algorithm is beneficial. Figure 8 shows the box-plots for the pooled data points (robots in stopped state in zone I) from the full ten-minute time interval of all repetitions. According to the Wilcoxon Rank-Sum Test the performance of the improved BEECLUST algorithm is significantly better (p-value $< 2.2 \times 10^{-16}$). The better performance is also noticeable by comparing the number of robots in stopped state in Figures 9e and 9f). This proves that the improved BEECLUST algorithm is able to perform under different light conditions without requiring the manual adaptation of the waiting function.

6. DISCUSSION AND CONCLUSION

We have proposed an adaptive variant of the original BEECLUST swarm robot control algorithm. The two variants were compared in a number of swarm robot experiments with different environmental conditions. The adaptive variant performs significantly better in the dimmed light condition and slightly better in bright light conditions (i.e., it successfully adapts to different light conditions). We have not tested the improved algorithm with a dynamic environment where the minimum and maximum brightness might change over time. That is left for future work but has been shown for the original BEECLUST algorithm [18]. Also we have not tested the improved algorithm in a symmetry-breaking



(a) original, one bright, $n = 5$ repetitions

(b) improved, one bright, $n = 5$ repetitions

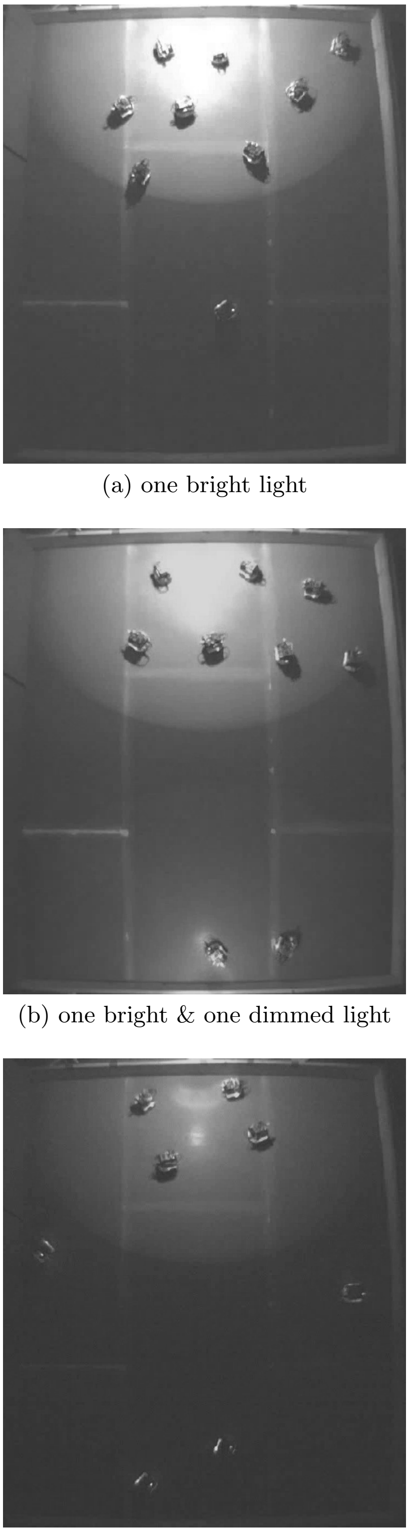
(c) original, one bright & one dimmed, $n = 5$ repetitions

(d) improved, one bright & one dimmed, $n = 4$ repetitions

(e) original, one dimmed, $n = 5$ repetitions

(f) improved, one dimmed, $n = 5$ repetitions

Figure 9: The minimum, maximum, and median number of stopped robots in zone I (brighter zone) during a ten-minute time interval for each experiment setting.



(a) one bright light

(b) one bright & one dimmed light

(c) one dimmed light

Figure 7: Typical situations observed during the experiment for different light settings.

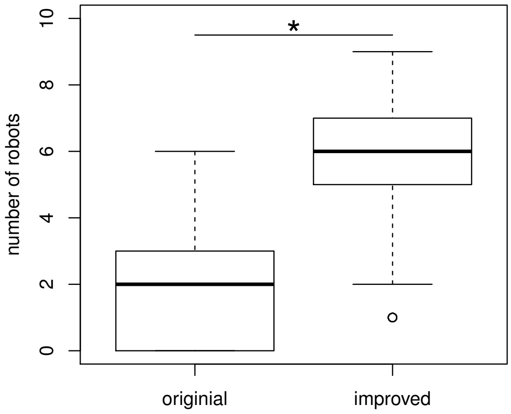


Figure 8: Comparison between the performance of both BEECLUST implementations in terms of the number of stopped robots in zone I during the last ten minutes time interval in the one dimmed light experiments.

problem but also that has been shown before for the original BEECLUST algorithm [8].

There are still many options of further improvements. In future work we plan to investigate the following options. Instead of only measuring the light distribution in a previous calibration phase the maximum and minimum light intensity can also be measured during the actual execution of the experiment. Older measurements could be neglected and that way the robots could achieve fully autonomous adaptivity to dynamic environments. Instead of cue-based communication only (robot–robot recognition), the robots could also explicitly communicate and share maximum and minimum light intensities and increase the consistency of their behavior and performance. Finally, we also want to introduce adaptiveness to changing swarm sizes/densities (i.e., adapting the maximal waiting time $w_{\max}$), for example, due to hardware failures, an opened door making more space available, or because robots are added later during the experiment. This would require to estimate the swarm size from local measurements (note that usually we want to avoid use of identification numbers in swarms because that approach does not scale; hence, just accumulating a list of IDs is not a promising option). That way we could achieve a fully adaptive robot swarm that solves its task efficiently also when swarm size, swarm density, or the light distribution are dynamic.

Acknowledgment

Project '*flora robotica*' has received funding from the European Union's Horizon 2020 research and innovation program under the FET grant agreement, no. 640959.

7. REFERENCES

[1] F. Arvin, A. E. Turgut, F. Bazyari, K. B. Arikan, N. Bellotto, and S. Yue. Cue-based aggregation with a mobile robot swarm: a novel fuzzy-based method. *Adaptive Behavior*, 22(3):189–206, 2014.

[2] F. Arvin, A. E. Turgut, and S. Yue. Fuzzy-based aggregation with a mobile robot swarm. In *Swarm*

Intelligence (ANTS'12), volume 7461 of *Lecture Notes in Computer Science*, pages 346–347, Berlin, 2012. Springer.

[3] G. Beni. From swarm intelligence to swarm robotics. In E. Şahin and W. M. Spears, editors, *Swarm Robotics - SAB 2004 International Workshop*, volume 3342 of *LNCS*, pages 1–9, Santa Monica, CA, July 2005. Springer-Verlag.

[4] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41, 2013.

[5] D. Cook and A. Vardy. Terrain leveling by a swarm of simple agents. In *Proceedings of the 8th International Conference on Bioinspired Information and Communications Technologies*, BICT '14, pages 55–58. ICST, 2014.

[6] N. Correll and H. Hamann. Probabilistic modeling of swarming systems. In J. Kacprzyk and W. Pedrycz, editors, *Springer Handbook of Computational Intelligence*, pages 1423–1431. Springer, 2015.

[7] M. Dorigo, M. Birattari, and M. Brambilla. Swarm robotics. *Scholarpedia*, 9(1):1463, 2014.

[8] H. Hamann, B. Meyer, T. Schmickl, and K. Crailsheim. A model of symmetry breaking in collective decision-making. In S. Doncieux, B. Girard, A. Guillot, J. Hallam, J.-A. Meyer, and J.-B. Mouret, editors, *From Animals to Animats 11*, volume 6226 of *LNAI*, pages 639–648, Berlin, Germany, 2010. Springer-Verlag.

[9] H. Hamann, T. Schmickl, and K. Crailsheim. Analysis of swarm behaviors based on an inversion of the fluctuation theorem. *Artificial Life*, 20(1):77–93, 2014.

[10] H. Hamann, T. Schmickl, H. Wörn, and K. Crailsheim. Analysis of emergent symmetry breaking in collective decision making. *Neural Computing & Applications*, 21(2):207–218, Mar. 2012.

[11] J. M. Hereford. Analysis of BEECLUST swarm algorithm. In *Proc. of the IEEE Symposium on Swarm Intelligence (SIS 2011)*, pages 192–198. IEEE, 2011.

[12] A. J. Ijspeert, A. Martinoli, A. Billard, and L. M. Gambardella. Collaboration through the exploitation of local interactions in autonomous collective robotics: The stick pulling experiment. *Autonomous Robots*, 11:149–171, 2001.

[13] S. Kernbach, R. Thenius, O. Kornienko, and T. Schmickl. Re-embodiment of honeybee aggregation behavior in an artificial micro-robotic swarm. *Adaptive Behavior*, 17:237–259, 2009.

[14] S. Magnenat, P. Rétornaz, M. Bonani, V. Longchamp, and F. Mondada. ASEBA: A modular architecture for event-based control of complex robots. *Mechatronics, IEEE/ASME Transactions on*, 16(2):321–329, 2011.

[15] M. J. Mataric. Kin recognition, similarity, and group behavior. In *Proceedings of the Fifteenth Annual Cognitive Science Society Conference*, pages 705–710, 1993.

[16] F. Riedo, M. Chevalier, S. Magnenat, and F. Mondada. Thymio II, a robot that grows wiser with children. In *IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO 2013)*, pages 187–193. IEEE, 2013.

[17] T. Schmickl and H. Hamann. BEECLUST: A swarm algorithm derived from honeybees. In Y. Xiao, editor, *Bio-inspired Computing and Communication Networks*. CRC Press, Mar. 2011.

[18] T. Schmickl, H. Hamann, H. Wörn, and K. Crailsheim. Two different approaches to a macroscopic model of a bio-inspired robotic swarm. *Robotics and Autonomous Systems*, 57(9):913–921, Sept. 2009.

[19] T. Schmickl, R. Thenius, C. Möslinger, G. Radspieler, S. Kernbach, and K. Crailsheim. Get in touch: Cooperative decision making based on robot-to-robot collisions. *Autonomous Agents and Multi-Agent Systems*, 18(1):133–155, Feb. 2008.

[20] O. Soysal and E. Şahin. Probabilistic aggregation strategies in swarm robotic systems. In *Proceedings of the Swarm Intelligence Symposium (SIS 2005)*, pages 325–332. IEEE, 2005.

[21] O. Soysal and E. Şahin. A macroscopic model for self-organized aggregation in swarm robotic systems. In E. Şahin, W. M. Spears, and A. F. T. Winfield, editors, *Swarm Robotics - Second SAB 2006 International Workshop*, volume 4433 of *LNCS*, pages 27–42, Berlin, Germany, 2007. Springer-Verlag.

[22] W. M. Spears, D. F. Spears, J. C. Hamann, and R. Heil. Distributed, physics-based control of swarms of vehicles. *Autonomous Robots*, 17(2-3):137–162, 2004.

[23] R. L. Stewart and R. A. Russell. A distributed feedback mechanism to regulate wall construction by a robotic swarm. *Adaptive Behavior*, 14(1):21–51, 2006.

[24] M. Szopek, T. Schmickl, R. Thenius, G. Radspieler, and K. Crailsheim. Dynamics of collective decision making of honeybees in complex temperature fields. *PLoS ONE*, 8(10):e76250, 10 2013.

[25] V. Trianni, R. Groß, T. H. Labella, E. Şahin, and M. Dorigo. Evolving aggregation behaviors in a swarm of robots. *Lecture Notes in Artfcial Intelligence*, 2801:865–874, 2003.

[26] G. Valentini and H. Hamann. Time-variant feedback processes in collective decision-making systems: Influence and effect of dynamic neighborhood sizes. *Swarm Intelligence*, 9(2-3):153–176, September 2015.

[27] J. Werfel, K. Petersen, and R. Nagpal. Designing collective behavior in a termite-inspired robot construction team. *Science*, 343(6172):754–758, 2014.