Plasticity in Collective Decision-Making for Robots: Creating Global Reference Frames, Detecting Dynamic Environments, and Preventing Lock-ins

Mohammad Divband Soorati[1,2,5], Maximilian Krome[2,5], Marco Mora-Mendoza[3], Javad Ghofrani[4], and Heiko Hamann[2]

[1]School of Electronics and Computer Science, University of Southampton, UK

[2]Institute of Computer Engineering, University of Lübeck, Germany

[3]Department of Computer Science, Stanford University, Stanford, CA, USA

[4]Department of Informatics and Mathematics, Dresden University of Applied Sciences, Germany

[5]both authors contributed equally

January 2020

Abstract

Swarm robots operate as autonomous agents and a swarm as a whole gets autonomous by its capability of collective decision-making. Despite intensive research on models of collective decision-making, the implementation in multi-robot systems is still challenging. Here, we advance the state of the art by introducing more plasticity to the decision-making process and by increasing the scenario difficulty. Most studies on large-scale multi-robot decision-making are limited to one instance of an iterated exploration-dissemination phase followed by successful and permanent convergence. We investigate a dynamic environment that requires constant collective monitoring of option qualities. Once a significant change in qualities is detected by the swarm, it has to collectively reconsider its previous decision accordingly. This is only possible by preventing lock-ins, a global consensus state of no return (i.e., a dominant majority of robots prevents the swarm from switching to another, possibly better option). In addition, we introduce a scenario of increased difficulty as the robots must locate themselves to assess the quality of an option. Using local communication, swarm robots propagate hop-count information throughout the swarm to form a global reference frame. We successfully validate our implementation in many swarm robot experiments concerning robustness to disruptions of the reference frame, scalability, and adaptivity to a dynamic environment.

1 Introduction

Collective decision-making [19] is an essential capability that is required to make robot swarms [5] autonomous on the swarm-level. Similar to decision-making in single agents using solid brains, collective decisions of liquid brains need to be rational [15]. In dynamic environments, made decisions may require reconsideration, hence we require an adaptive swarm behavior of decision-making that collectively monitors dynamic features of the environment and switches to a better decision when necessary. A precondition for adaptivity is that the swarm stays flexible at all times, which means that the swarm must not be trapped in a lock-in state of an outdated decision [1, 6, 19, 11]. This way we address a shortcoming of the literature on collective decision-making in robot swarms [9], since it is often ignored that a collective decision is embedded in a context of a before and after decision, and that global awareness in the swarm is required to determine when to start a collective decision-making process and when to end it. With our approach, we achieve that the swarm monitors and adapts to changes in option quality but without reaching swarm-level awareness of when the current collective decision-making process is completed.

In terms of biological inspiration, the work of Dussutour et al. [10, 3] is of importance. They show the relevance of noise to keep a collective decision-making system adaptive to dynamic environments. Also in collective decision-making we face an exploration-exploitation tradeoff between exploring new options or just revisiting known options to monitor changes in quality and exploiting already known option qualities. In our approach balancing this tradeoff is simpler because exploration has basically zero cost for an individual robot (sensing the local light intensity). However, information about option qualities is distributed in space and requires a combination of collective perception and swarm-wide awareness.

Here we study a collective decision-making scenario of increased complexity. We start from previous works on collective perception [18] and a similar work with focus on multi-feature decision-making [4]. The robots need to explore the environment to learn about the quality of two (or more) options. The task is that studied by Valentini et al. [18]. There are black and white tiles and the robots need to agree on whether the black or the white tiles are in majority. The percentage of the black tiles is the quality of the black option and vice versa. The robots share their knowledge about the environment and collectively try to converge on a consensus or a large majority in favor of the better choice. We add a level of difficulty since the environment is dynamic and the swarm needs to form a global reference frame for relative localization of robots. In our task, we require environmental features to be measured relative to spatial regions identified by beacons and multi-hop communication. We test the adaptivity and robustness of our approach to collective decision-making by either making the environmental feature dynamic or by disrupting the global reference frame (moving beacons). In addition, we test the scalability of our approach in large-scale simulations.

Our main contribution is the adaptivity of collective decision-making by pre-

venting lock-ins in a scenario of increased difficulty (environmental monitoring relative to space). We study our concept in large-scale robot experiments with up to 64 real, physical robots and up to 1024 robots in simulation. To the best of our knowledge, only a few large-scale studies on robot hardware for collective decision-making have been reported [20, 4, 16, 17]. Otherwise, there are many studies based on simulations and models, such as investigations of collective perception [14], software-engineering aspects [12], and density-relative performance [21, 8].

2 Method

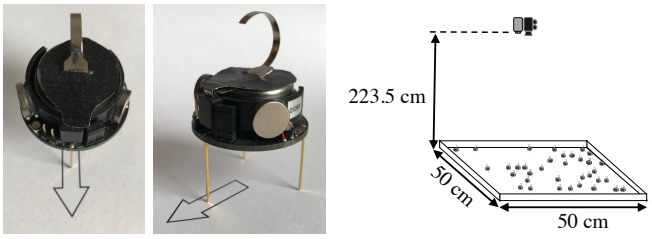
2.1 Setup

We used Kilobots for the physical experiments and the Kilombo platform to do the simulations. The Kilobot is a small (diameter of 3.3 cm), low-cost robot with an infrared receiver and transmitter for communication. Messages are 12 bytes in length with 9 bytes reserved for the payload. The messages can reach other robots in approximately 10 cm range at a rate of up to 30 kb/s. The robots also have RGB LED light emitters, two independent vibrating motors for differential drive locomotion, and a light sensor for sensing the ambient light. The maximum speed of the robot is 1 cm/s and its rotational speed is of $\pi/4$ rad/s [13]. Our physical experimental environment consists of a square arena with a total area of 50×50 cm^2. The arena is divided into a grid of square cells of equal size, 6.25×6.25 cm^2. Each of these cells can either be illuminated or not, with the illuminated cells receiving the highest light intensity achieved by a video projector with maximum of 1200 lumen. A camera captures video at a distance of 223.5 cm (see Fig. 1(c)). A black piece of paper covers the top of each robot to prevent any light reflections (see Fig. 1(a)) that can interfere in detecting the color of the LEDs in the videos.

The Kilombo simulator, developed by Jansson et al., provides a virtual environment with features that facilitate the development and testing of Kilobot software, allowing the same code to be used in both virtual and physical robots [7]. The robots are represented by circles with arrows that point in the direction the virtual Kilobot is facing. The number of robots can be conveniently set, and features such as additional screen statements make it easy for debugging. Our virtual environment consists of a limited square arena in which the robots are allowed to move. The arena is further divided into a grid of multiple smaller, equally-sized square areas that are either bright or dark in a similar way to the real robot experiments (see Fig. 2). 1024 robots are scattered randomly across the arena, with two stationed seed Kilobots on opposite sides.

2.2 Experiment Scenarios

We use two robots as *seeds* that are fixed to the opposite sides of the arena, marked with red and blue circles in Fig. 2. Having the seeds on opposite sides



(a) Top view (b) Side view (c) Setup

Figure 1: A Kilobot from the top (a) and the side view (b) is shown. A projector at the top projects the light patterns and a camera captures the videos (c).

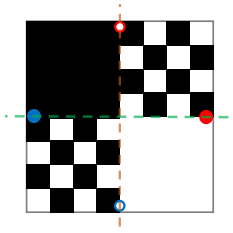


Figure 2: Light distribution with the positions of the seeds before (rings) and after relocation (solid circles).

allows to determine the seed in the brighter side and easily evaluate the performance of the swarm using the setup shown in Fig. 2. The task of the swarm is to reach a (temporary) consensus on which seed is located in the brighter half of the arena. We consider the task successfully completed if two thirds of robots are in favor of the better option. We evaluate the adaptivity of the swarm with two types of changes: by relocating the seeds we disrupt the internal reference frame used by the robots for localization and by changing the light condition we implement a dynamic environment.

2.2.1 Seed relocation

In this scenario, we test for robustness of the robots' reference frame to a disruption. The arena consists of a grid with 32 white and 32 black cells. For the first phase of the experiment, the seeds are at the top and the bottom of the arena, marked with red and blue rings in Fig. 2. The swarm has to collectively decide which seed is located in the brighter side. The green dashed line separates the upper half and lower half, with 8 and 24 white cells respectively, making the lower half the brighter side. At the beginning of the second phase, the seeds are relocated to the left and the right sides of the arena, marked with red and blue filled circles in Fig. 2). The brown dashed line marks the separation of the two halves, with 8 white cells at the left and 24 at the right. As an effect to the

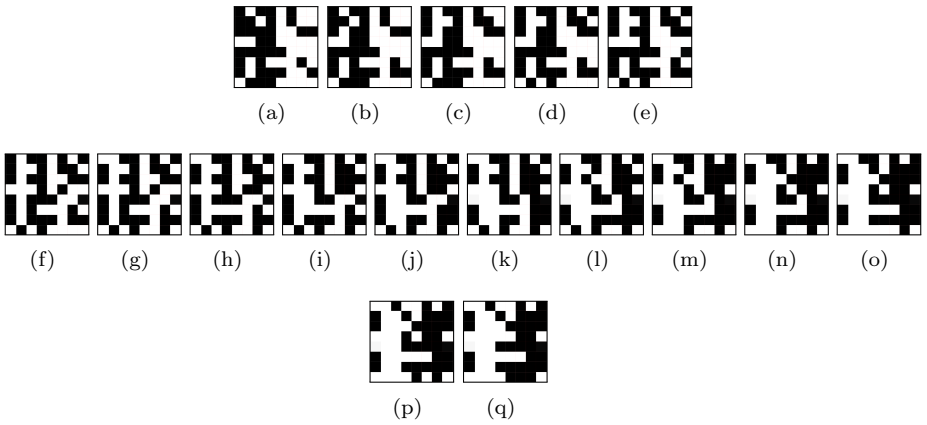


Figure 3: Projected light sequence for the changing light experiments. The first light setup (a) is used for the first 150 s of the experiments. The sequence (b to p) changes in time steps of 20 s. The last light setup (q) lasts longer (150 s for physical and 750 s for simulated robots) to give the robots enough time to adapt to the condition.

relocation of the seeds after 150 s, the robots need to adapt and update their relative positions to the beacons, while the swarm has to continue to perform the task. We perform two sets of experiments with the seed relocation for 90° degrees (as described above) and 180° degrees clockwise evaluating the effect of each change during three experiments.

2.2.2 Changing light condition

For the second scenario the seeds are fixed for the entire runtime but the lighting condition changes gradually over time. We test the robots' adaptivity to a dynamic environment. For the first 150 s the grid shown in Fig. 3(a) is projected to the arena, followed by Figs. 3(b)-3(p) each for 20 s. For each consecutive pair, a black cell from the left half of the gird is replaced by a randomly chosen white cell from the right half. The last grid shown in Fig. 3(q) lasts longer (150 s for physical and 750 s for simulated robots) giving the swarm enough time to fully adapt to the change. All scenarios are evaluated by three real and twenty simulated robot experiments.

2.3 Localization

In order for the robots to determine the seed with the best lighting conditions, they must be able to know their relative distance to each of the seeds. They can then make individual decisions about which seed's side they estimate to have better environmental conditions. They share their opinion by messaging with their neighbors and try to reach consensus. To do this, we implement

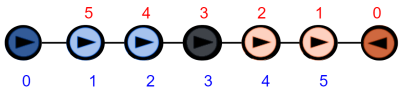


Figure 4: Gradient values of each Kilobot for respective seed. The colors of the robots signal the color of closest seed, determined by lowest gradient value. Robots with zero gradient are the seeds.

a gradient algorithm that makes use of the limited messaging capabilities of each Kilobot to locally communicate their distance to each seed. Each robot allocates part of its memory to store a history buffer for each of the seeds that pushes the last 6 gradient values that the robot has received from its neighbors, the maximum number of neighbors a robot can have because of its size. At the same time, each robot stores calculated values called gradient g_A for seed A and gradient g_B for seed B that are initialized to a placeholder maximal value. This variable corresponds to the geodesic distance of the robot to each seed (hop count). The value of g_A and g_B corresponds to one more than the minimum value in the history buffer array for that seed and is stored as a local variable to be transmitted. By default, the seed Kilobots communicate a gradient message with a value of zero and ignore the incoming messages.

All Kilobots, including the seeds, transmit their gradient value to the surrounding robots. The further away a robot is from a seed, the more Kilobots between the seed and that robot have relayed the hop-count message and the higher its gradient value. The closest seed to each Kilobot is determined by the lowest gradient value of the two (see Fig. 4).

The Kilobots update their values continuously while moving and changing their position relative to the seeds. If a Kilobot stops receiving messages from other robots for longer than a threshold θ_{discon} then it marks itself as disconnected. If it continues to not receive any messages beyond a threshold $\theta_{\text{reset}} > \theta_{\text{discon}}$ it proceeds to clear its history buffers, stops sending messages, and resets its gradient for that particular seed to be a maximal placeholder value, waiting to be updated to any lower value as soon as the robot receives a new message. The used parameters are $\theta_{\text{reset}} = 200$ s, $\theta_{\text{discon}} = 100$ s, and $\theta_{\text{brightness}} = 300$.

2.4 Collective Decision-Making

There are two sub-populations in the robot swarm: the majority of deciders that collect the messages of neighboring robots, processes them, reconsiders their own opinion, and disseminates this opinion; a minority functioning as sensors that measure light, determine their opinion based on the brightness level, and disseminate it to the surrounding neighbors. Each robot can be in one of two states: *measuring* or *deciding* (see Fig. 5). All robots start in the *measuring* state to gather information about the environment and proceed to disseminate their opinion $o \in \{-1, +1\}$ about which seed is in a brighter location. Kilobots

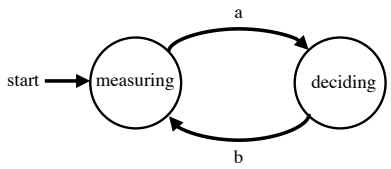


Figure 5: Finite state machine of the robot controller

measure the light in range [0,1023] and the threshold $\theta_{\text{brightness}}$ gives the minimum light value needed to consider an area bright. Robots in state *measuring* determine their opinion depending on the measured light and the two gradient values g_A and g_B (see Sec. 2.3) by

$$o = \begin{cases} -1, & (g_A < g_B \wedge [\text{bright}]) \vee (g_A > g_B \wedge [\text{dark}])) \\ +1, & (g_A < g_B \wedge [\text{dark}]) \vee (g_A > g_B \wedge [\text{bright}])) \end{cases} . \tag{1}$$

A robot perceiving bright light takes the opinion $o = -1$ that the close-by seed is brighter. In the case that it perceives a dark area, it takes opinion $o = +1$ that the farther away seed is brighter. All robots accumulate received opinions o_i of their i neighbors during the past. A robot's confidence $c \in [-c_{\max}, +c_{\max}]$ determines its decision of which seed seems to represent the brighter half of the arena currently (see [2] for a similar approach). We set $c_{\max} = 1000$ for swarm size $N = 64$ and $c_{\max} = 5000$ for swarm size $N = 1024$. Robots in state *deciding* determine their opinion depending on their confidence: $o = -1$ if $c < 0$ and $o = +1$ if $c > 0$. When a neighbor's message (o_i, w_i) is received, robots in either of both states update their confidence c_t by

$$c_{t+1} = \begin{cases} c_t + o_i w_i, & \text{if } |c_t + o_i w_i| < c_{\max} \\ c_{\text{t}}/2, & \text{otherwise} \end{cases} , \tag{2}$$

where $o = \pm 1$ is the neighbor's opinion and $w \in \{1, 10\}$ is the neighbor's weight. If the incoming opinion is from a robot in state *measuring*, then the weight is $w = 10$, otherwise the weight is $w = 1$. We need to limit the confidence to a maximal value $|c| < c_{\max}$ to prevent lock-in states and to implement adaptivity. Without this limit robots would accumulate big confidence values $|c|$ and would barely be able to react to a dynamic environment or changes in the seeds. Once $|c| \geq c_{\max}$, a robot's confidence is halved (c/2). An additional mechanism to prevent lock-ins is a time-out: if a robot receives no message (o_i, w_i) with its own opinion $(o = o_i)$ for 60 s, then it resets its confidence $c = 0$. When a waiting time of one second elapses and no message from a measuring robot is received, a transition to state *measuring* is triggered. A robot may leave state *measuring* if a neighboring robot is also in state *measuring*, then a probabilistic transition to state *deciding* is triggered with probability 0.1. For our experiments with physical Kilobots we have an additional case by time-out: a transition to state

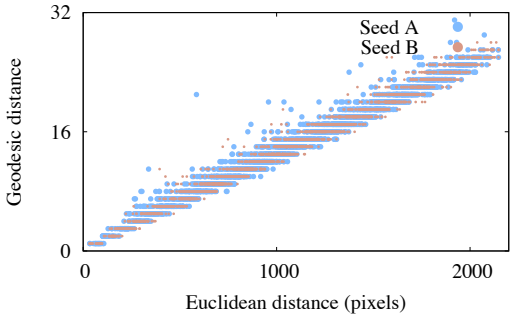
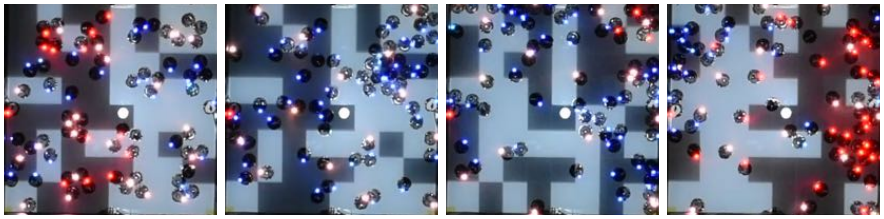


Figure 6: Test of the formed reference frame as described in Sec. 2.3; geodesic (hop-count) and Euclidean distance from the seeds of a simulated robot.

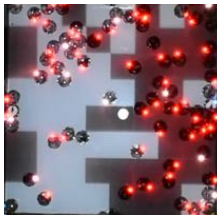
deciding is unconditionally triggered after 30 s (robots in state *measuring* can get blocked and would measure the same value for a long time). At all times, all robots process incoming messages and update their confidence. Due to the restricted space of each Kilobot message to be 9 bytes, we limit the size of the votes exchanged between robots to be 1 bit of information. A message of 1 from a neighboring robot corresponds to a vote for seed A and a message of 0 corresponds to a vote for seed B.

3 Results

Our robots rely on the gradient values as described in Sec. 2.3 to create a global reference frame—hop counts from each of the two seeds—and to measure their proximity to the seeds. To evaluate the quality of the formed reference frame we compare the geodesic distance (hop-count) to the Euclidean distance. To assess the correlation between the gradient values and the Euclidean distances, we let the simulated robots move randomly in the arena and record the gradient and the Euclidean distances of one of the robots to both seeds. Fig. 6 shows the expected linear relation between the two distance measures. The formed reference frame works properly and can be used to estimate distances. Fig. 7 shows photos of an experiment with changing light conditions taken every 2.5 minutes. The light condition is fixed for the first 2.5 minutes and the swarm reaches a consensus that the side with the blue seed (right side) is brighter, as signaled by the LED of the individuals turning blue (see Fig. 7(b)). According to the light changing scenario (see Fig. 3), the lighting then changes gradually until the left half is brighter. As shown in Fig. 7(e), the swarm adapts almost completely to the change and switches the opinion to vote for the left half being the brighter side, as signaled by the LEDs on each robot turning red. Fig. 8 shows a simulated experiment of the side relocation scenario over time. The swarm starts with all robots in *measuring* state signaled by a white LED. Within a short time a majority of blue is reached ($t = 150$ s), the swarm collectively agrees



(a) $t = 1$ s (b) $t = 150$ s (c) $t = 300$ s (d) $t = 450$ s

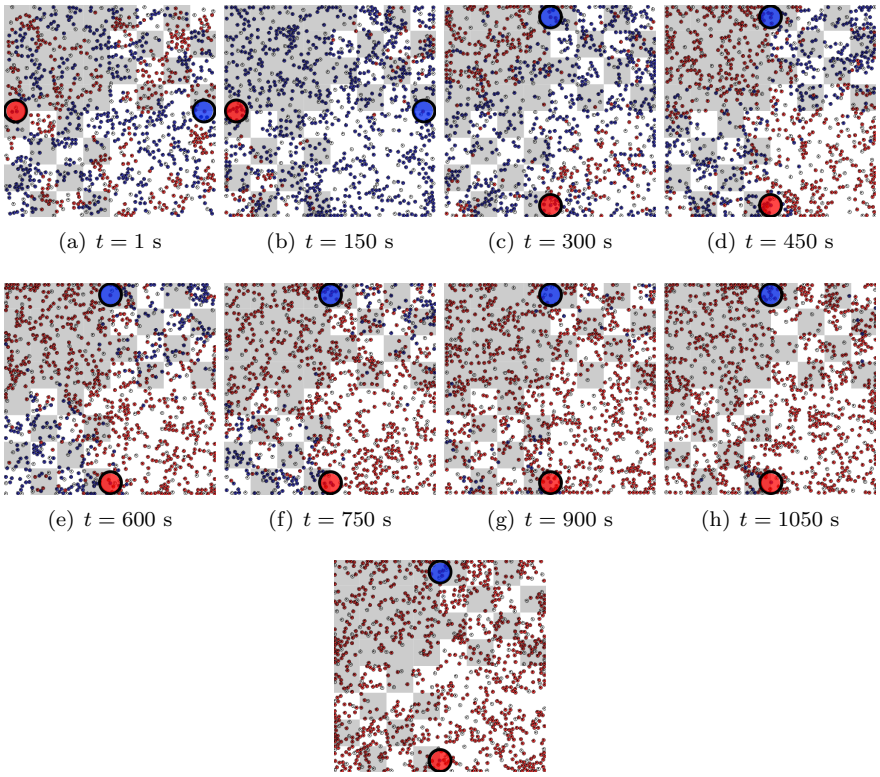


(e) $t = 600$ s

Figure 7: Photos of the robot experiment for the scenario of a dynamic environment with changing light conditions for 600 s, $N = 64$ robots, red seed on the left, blue seed on the right, (a) initial setup, (b) a majority is in favor of the blue seed, that is, correctly voting for the right half being brighter, (c,d) gradual changes of the light conditions, (e) final situation of the experiment with a majority for the red seed (left half) being brighter.

that the blue seed is brighter (blue seed represents right half). By relocating the seeds at $t = 151$ s (marked with black rings) the swarm successfully detects the change, updates its collective decision, and decides that the lower half with the red seed is brighter now ($t = 1200$ s, see Fig. 8(i)). We let the simulations to run for $t = 1200$ s while in reality each repetition takes only $t = 600$ s. The reason is that there are 1024 simulated robots that need to adapt, compared to 64 in reality.

We record videos of the robot experiments and process them to extract the total number of robots with blue and red LEDs. Using blob detection in the b* and a* channels of the CIELAB color space the blue and the red emissions of the LEDs are detected and counted. Fig. 9 shows the median number of robots that vote for seed A and B with red and blue lines over time. The red and blue areas show the range between the upper and lower quartiles. When two third of the voting robots decide for a seed, we say the swarm has a collective decision. We sum over the median number of votes at a time and plot the two third of this value with a green line. All the results for the experiments with physical robots show that the swarm was able to reach or exceed the two third of the votes for seed B during the first phase. After the changes the swarm detects the change, reconsiders its past decision, and correctly votes in majority for seed A (red). For the 90° (see Fig. 9(a)) and 180° relocations of the seeds (see Fig. 9(c)) the

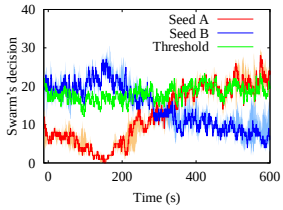


(a) $t = 1$ s (b) $t = 150$ s (c) $t = 300$ s (d) $t = 450$ s

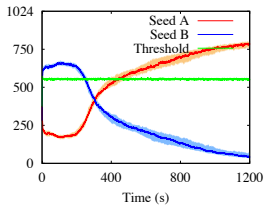
(e) $t = 600$ s (f) $t = 750$ s (g) $t = 900$ s (h) $t = 1050$ s

(i) $t = 1200$ s

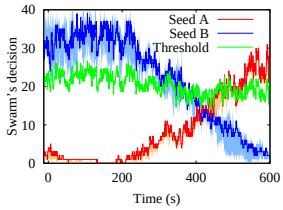
Figure 8: Simulation of the instant seed relocation scenario with $N = 1024$ robots for 1200 s. (a) initial situation, (b) progress over time, (c) correct majority voting for the blue seed on the right half being brighter, (d) relocation of the seeds marked by red and blue circles, (e) final situation of the experiment with a majority correctly voting for the bottom half being brighter.



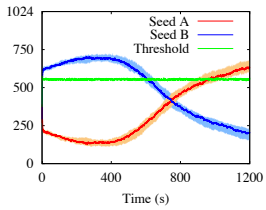
(a) Instant seed relocation (90°) at $t = 150$ s with physical robots



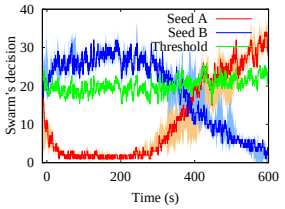
(b) Instant seed relocation (90°) at $t = 150$ s with simulated robots



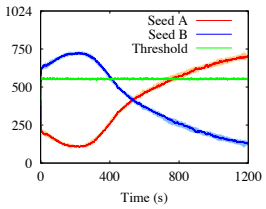
(c) Instant seed relocation (180°) at $t = 150$ s with physical robots



(d) Smooth seed relocation (moving seeds, 90°) with simulated robots



(e) Dynamic environment, gradual light change during time interval from $t = 150$ s until $t = 450$ s with physical robots



(f) Dynamic environment, gradual light change during time interval from $t = 150$ s until $t = 450$ s with simulated robots

Figure 9: Robot experiments with a swarm of $N = 64$ physical robots (a, c, and e) for 600 s and $N = 1024$ simulated robots (b, d, and f) for 1200 s. Red lines give the number of robots voting for seed A and blue lines give the number of robots voting for seed B. Green lines give two third of the total number of robots that contribute in voting (robots in state *deciding*, excluding robots in state *measuring*). Values are smoothed over a time window, shades give minimum and maximum values.

majority decides to vote for seed B first. When the seeds are relocated, the swarm quickly detects that change and re-decides by voting for the other seed. The initial number of robots in favor of seed B is different between Fig. 9(a) and Fig. 9(c) because we project different light patterns. In the case of the 90° relocation (see Fig. 9(a)) we project the checker board pattern (see Fig. 2) that has increased difficulty (checker board areas are large neutral areas) and for the 180° relocation (see Fig. 9(c)) we project a more randomized pattern. A similar pattern is observed for the experiments with dynamic light conditions shown in Fig. 9(e). A clear majority in favor of seed B forms, then the changing light conditions are detected, the swarm collectively reconsiders its past decision, and a majority in favor of seed A forms.

The scalability of our method for 1024 simulated robots is shown with a similar observation from the simulations (see Fig. 9(b, d, and f)). The relocation in the simulation is done via an instant swapping of the seeds similar to the real experiments as well as a smooth relocation where the robots switch sides by moving in a predefined curved line. Fig. 9(d) shows the result of the smooth relocation where the swarm also reacts slowly to the change compared to the quick response in Fig. 9(b). Comparing the results of the physical robot experiments with the results of the simulations (see Fig. 9), the agreement is very good except for more noise in the robot experiments as expected. A video is available as supplementary material; images of higher resolution and videos of all experiments are available online[1].

4 Conclusion

We have implemented and successfully validated our approach to increased plasticity in collective decision-making for large-scale robot swarms. Instead of a one-shot collective decision-making process concluded by a lock-in, our robot swarm stays adaptive by collectively monitoring a dynamic environment and reconsidering past decisions. Adaptivity necessarily comes with the cost of being more susceptible to noise, including the possibility of switching temporarily to a wrong decision. In our experiments we have not seen any such effect yet, hence future work will include a study of increased noise levels. Similarly, there must be limits of how quickly and reliably the swarm can converge compared to its capability of staying locked in a decision state as long as no switch is required. There are multiple other options of how to extend this work. An example is an in-depth study of scenarios with multiple seeds that could be used to mark areas of interest. Introducing scenarios with multiple environmental features [4] is another possibility (e.g., green vs yellow in addition to white vs black patches).

[1] https://doi.org/10.5281/zenodo.2581670

References

[1] W. Brian Arthur. Competing technologies, increasing returns, and lock-in by historical events. *The Economic Journal*, 99(394):116–131, 1989.

[2] Marco A. Montes de Oca, Eliseo Ferrante, Alexander Scheidler, and Louis F. Rossi. Binary consensus via exponential smoothing. In Kristin Glass, Richard Colbaugh, Paul Ormerod, and Jeffrey Tsao, editors, *Complex Sciences*, pages 244–255, Cham, 2013. Springer International Publishing.

[3] Audrey Dussutour, Madeleine Beekman, Stamatios C. Nicolis, and Bernd Meyer. Noise improves collective decision-making by ants in dynamic environments. *Proceedings of the Royal Society London B*, 276:4353–4361, December 2009.

[4] Julia T. Ebert, Melvin Gauci, and Radhika Nagpal. Multi-feature collective decision making in robot swarms. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, pages 1711–1719, Richland, SC, 2018. International Foundation for Autonomous Agents and Multiagent Systems.

[5] Heiko Hamann. *Swarm Robotics: A Formal Approach*. Springer, 2018.

[6] Heiko Hamann and Gabriele Valentini. Swarm in a fly bottle: Feedback-based analysis of self-organizing temporary lock-ins. In Marco Dorigo, Mauro Birattari, Simon Garnier, Heiko Hamann, Marco Montes de Oca, Christine Solnon, and Thomas Stützle, editors, *Ninth Int. Conf. on Swarm Intelligence (ANTS 2014)*, volume 8667 of *LNCS*, pages 170–181. Springer, 2014.

[7] Fredrik Jansson, Matthew Hartley, Martin Hinsch, Ivica Slavkov, Noemí Carranza, Tjelvar SG Olsson, Roland M Dries, Johanna H Grönqvist, Athanasius FM Marée, James Sharpe, et al. Kilombo: a kilobot simulator to enable effective research in swarm robotics. *arXiv preprint arXiv:1511.04285*, 2015.

[8] Yara Khaluf, Carlo Pinciroli, Gabriele Valentini, and Heiko Hamann. The impact of agent density on scalability in collective systems: noise-induced versus majority-based bistability. *Swarm Intelligence*, 11(2):155–179, Jun 2017.

[9] Yara Khaluf, Pieter Simoens, and Heiko Hamann. The neglected pieces of designing collective decision-making processes. *Frontiers in Robotics and AI*, 6:16, 2019.

[10] Bernd Meyer, Madeleine Beekman, and Audrey Dussutour. Noise-induced adaptive decision-making in ant-foraging. In *Simulation of Adaptive Behavior (SAB)*, number 5040 in LNCS, pages 415–425. Springer, 2008.

[11] Judhi Prasetyo, Giulia De Masi, Pallavi Ranjan, and Eliseo Ferrante. The best-of-n problem with dynamic site qualities: Achieving adaptability with stubborn individuals. In Marco Dorigo, Mauro Birattari, Christian Blum, Anders L. Christensen, Andreagiovanni Reina, and Vito Trianni, editors, *Swarm Intelligence*, pages 239–251, Cham, 2018. Springer International Publishing.

[12] Andreagiovanni Reina, Gabriele Valentini, Cristian Fernandez-Oto, Marco Dorigo, and Vito Trianni. A design pattern for decentralised decision making. *PLOS ONE*, 10(10):1–18, 10 2015.

[13] Michael Rubenstein, Alejandro Cornejo, and Radhika Nagpal. Programmable self-assembly in a thousand-robot swarm. *Science*, 345(6198):795–799, 2014.

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

[15] Ricard Solé, Melanie Moses, and Stephanie Forrest. Liquid brains, solid brains. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 374(1774):20190040, 2019.

[16] Mohammad Divband Soorati, Javad Ghofrani, Payam Zahadat, and Heiko Hamann. Robust and adaptive robot self-assembly based on vascular morphogenesis. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4282–4287. IEEE, 2018.

[17] Mohammad Divband Soorati, Mary Katherine Heinrich, Javad Ghofrani, Payam Zahadat, and Heiko Hamann. Photomorphogenesis for robot self-assembly: adaptivity, collective decision-making, and self-repair. *Bioinspiration & biomimetics*, 14(5):056006, 2019.

[18] Gabriele Valentini, Davide Brambilla, Heiko Hamann, and Marco Dorigo. Collective perception of environmental features in a robot swarm. In *10th Int. Conf. on Swarm Intelligence, ANTS 2016*, volume 9882 of *LNCS*, pages 65–76. Springer, 2016.

[19] Gabriele Valentini, Eliseo Ferrante, and Marco Dorigo. The best-of-n problem in robot swarms: Formalization, state of the art, and novel perspectives. *Frontiers in Robotics and AI*, 4:9, 2017.

[20] Gabriele Valentini, Eliseo Ferrante, Heiko Hamann, and Marco Dorigo. Collective decision with 100 Kilobots: Speed vs accuracy in binary discrimination problems. *Journal of Autonomous Agents and Multi-Agent Systems*, 30(3):553–580, 2016.

[21] Christian A. Yates, Radek Erban, Carlos Escudero, Iain D. Couzin, Jerome Buhl, Ioannis G. Kevrekidis, Philip K. Maini, and David J. T. Sumpter. Inherent noise can facilitate coherence in collective swarm motion. *Proc. Natl. Acad. Sci. USA*, 106(14):5464–5469, 2009.