Multi-Feature Collective Decision Making in Robot Swarms

Robotics Track

Julia T. Ebert
Harvard University
Cambridge, MA
ebert@g.harvard.edu

Melvin Gauci
Harvard University
Cambridge, MA
mgauci@g.harvard.edu

Radhika Nagpal
Harvard University
Cambridge, MA
rad@eecs.harvard.edu

ABSTRACT

Collective decision making has been studied extensively in the fields of multi-agent systems and swarm robotics, inspired by its pervasiveness in biological systems such as honeybee and ant colonies. However, most previous research has focused on collective decision making on a single feature. In this work, we introduce and investigate the multi-feature collective decision making problem, where a collective must decide on multiple binary features simultaneously, given no *a priori* information about their relative difficulties. Each agent may only estimate one feature at any given time, but the agents can locally communicate their noisy estimates to arrive at a decision. We demonstrate a decentralized algorithm for single-feature decision making and a dynamic task allocation strategy that allows the agents to lock in decisions on multiple features in finite time. We validate our approach using simulated and physical Kilobot robots. Our results show that a collective can correctly classify a multi-feature environment, even if presented with pathological initial agent-to-feature allocations.

KEYWORDS

Collective decision making; heterogeneous collectives; task switching; Kilobots

ACM Reference Format:
Julia T. Ebert, Melvin Gauci, and Radhika Nagpal. 2018. Multi-Feature Collective Decision Making in Robot Swarms. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 9 pages.

1 INTRODUCTION AND RELATED WORK

Collective systems in nature are ones in which a large number of relatively simple agents interact with each other to produce complex behaviors [3]. Decision making is a key behavior that appears across many of these systems and has been extensively studied over the past few decades [4]. In general, the agents are required to choose one out of multiple options present in their environment. They have some means of obtaining noisy information about these options and of influencing each other directly through communication or indirectly through stigmergy to achieve consensus on a single decision [22].

One of the best-known examples of collective decision making is the nest or shelter selection problem in insect colonies, also known as house hunting. The colony must choose exactly one out of a number of options of potentially different qualities—ideally picking the highest-quality option. For this reason, the problem is often called a best-of-N problem. The nest selection process is well-documented in honeybees [13, 17–19], and recently, a cross-inhibition mechanism was discovered in this system and identified as a key for its success [20]. House hunting is also performed by ants, and it has been demonstrated that certain species can reliably solve the best-of-N problem with a low probability of the colony splitting [6, 21] by using stigmergy in the form of pheromone trails. Halloy et al. [9] used robots to explore the shelter selection problem in cockroaches. Naturally, cockroaches prefer darker shelters over lighter ones. The researchers introduced a group of robots coated with pheromone such that they were accepted by a group of cockroaches as conspecifics. The robots, being programmed to prefer the lighter shelter, were able to socially influence the cockroaches so that they, on average, also made this 'unnatural' choice.

In many of these scenarios, it is not sufficient to simply approach consensus. At some point, the decision must be 'locked in', allowing the collective to move on to the next task; for example, the bees or ants must start emigrating and moving their larvae to the new site. In the context of bacterial colonies, this process is known as quorum sensing [11, 25]. When the concentration of an extracellular signaling molecule produced by the bacteria crosses a threshold, the colony can move from stasis to an active state.

The collective decision-making problem in its "best-of-N" form has also been studied within the context of robotic systems. Parker and Zhang [12] studied a scenario in which a group of robots is expected to choose the best out of a number of unequal options. The robots employ an active recruitment strategy that relies on inter-robot communication. The robots start by looking for options and advocating them to each other, always switching their selection to the best of the known options. Once a robot's selection becomes sufficiently popular (reaching a *quorum*), the robot becomes committed to it. This enables the group to reach a consensus where all robots have locked in decisions. Hamann et al. [10] studied how a homogeneous group of robots can collectively choose between two global maxima in a light-intensity field. In their algorithm, each robot moves in a straight line until it encounters another robot. Then, it stops and counts the total number of robots in its neighborhood. If this is above some threshold, the robot measures the light intensity and waits for a time proportional to this intensity. This creates a positive feedback effect which enables symmetry breaking between the two options. Valentini et al. [24] studied a swarm of robots that collectively choose among two unequal options. At any moment in time, each robot has an opinion about which option is best. The robot either explores the option, or exchanges information with its neighbors. In the latter case, the robot locally broadcasts its opinion for a duration that is proportional to the perceived quality

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

of the preferred option, analogous to the honeybee waggle dance. The robot also monitors incoming messages for a fixed time period, and then updates its opinion using the majority rule. The robot then switches to exploring the potentially new option, and the process repeats indefinitely. The results showed that the collective approaches consensus with high accuracy.

Recently, Valentini et al. [23] studied a modified version of the collective decision making problem on a robotic system, in the form of *feature detection*. The agents are able to sense color in a black-and-white environment and are required to estimate whether the environment contains more back or white area. Each agent makes individual estimates of the feature and shares it globally with all other agents (i.e., communication is global). The agents then aggregate the estimates of other agents to form a belief about which color is more prominent. Several aggregation mechanisms were investigated in simulation and on a group of 20 e-puck robots, resulting in different speed/accuracy trade-offs.

One limitation of the work in [23] is that it relies on global communication, whereas collective systems in nature exploit the use of only local communication to achieve scalability [2]. Moreover, the work in [23]—as well as all the others mentioned above—only investigate the case where the collective is asked to make a single decision. However, collective systems in nature are capable of handling multiple tasks simultaneously, adaptively allocating individuals as required to complete these tasks [7, 8, 14, 21]. In this work, we introduce and investigate the *multi-feature collective decision making problem*. We present a scenario in which the collective must make decisions about three color features in an environment using only local communication, extending the scenario in [23]. We also demonstrate a decision making rule and a dynamic task allocation strategy that allow the agents to lock in decisions in finite time.

2 PROBLEM DEFINITION AND MOTIVATION

We consider N agents that move in a 2-dimensional environment, with boundaries that are detectable by the agents. The environment contains M features that individual agents have a means of estimating. The features may either be inherently binary-valued, or else the agents must have some agreed-upon threshold for making them as such. The features can thus be represented as functions:

$$f_i : \mathcal{S} \to \{0, 1\}, \quad i \in \{1, 2, \ldots, M\} \tag{1}$$

where $\mathcal{S}$ is the environment. We can group the features into a vector-valued function over $\mathcal{S}$:

$$\mathbf{f}(\mathcal{S}) = [f_1(\mathcal{S}), f_2(\mathcal{S}), \ldots, f_M(\mathcal{S})]^T \tag{2}$$

The features are defined over the environment *as a whole*. Examples of features include the ratio of white area to the total area in a black-and-white environment (as in [23]), the entropy of a pattern, and the amount of curvature present in a pattern. It is assumed that the agents have some means of making (noisy) estimates of the features. Considering the above examples, the color fill ratio feature can be estimated by calculating the ratio observed over a random walk; the entropy ratio might be estimated by calculating the amount of regularity observed over a straight-line motion; and the curvature might be estimated by performing edge following whenever an edge is detected, and calculating the average radius of curvature. In this work, we focus on features of the type of the first example discussed above (color fill ratio). Specifically, the feature is *locally defined at every point* (x, y) in the environment, mapping the point onto a value in $\{0, 1\}$. The binary value of the feature *over the whole environment* is defined as the rounded value of the fill ratio:

$$f_i(\mathcal{S}) = \text{round}\left\{ \frac{1}{|\mathcal{S}|} \iint_S f_i(x, y)\, dA \right\}, \tag{3}$$

where $|\mathcal{S}|$ is the area of the environment and round$(*)$ is the usual rounding function, outputting 0 if $* < 0.5$, and 1 otherwise. In other words, $f_i(\mathcal{S})$ assumes whichever one of the two values is more prevalent in the environment. Obtaining estimates for features of this type is straightforward: the agent simply samples a subset of points over the course of a random walk.

The agents are limited in what actions they may take. Each agent is capable of estimating every feature in the environment, but is restricted to estimating only one feature during each *observation period* (defined later in Sec. 4.1). This limitation is mainly imposed because in general, estimating different features may require different motion patterns; moreover, in practice the computational power available on simple physical agents may not be enough to sense and process all the features at once.

The agents are able to communicate with each other. They may listen to messages from other agents continuously, but are only allowed to transmit messages while they are not observing the environment. Once again, this limitation is imposed because of the different motion patterns that may be required to estimate the different features; for example, if agents were to disseminate while edge following, this might bias their transmission towards other agents that happen to be performing the same motion pattern in the same region, and so a random walk would be a more desirable motion pattern during transmission.

The problem for the collective is to decide on the value of each feature over the environment as a whole; in other words, to compute Eq. 2. The collective is required to not only have an estimate that converges to Eq. 2 over time, but also to make a unanimous decision in finite time. When collective decision making is used as a primitive component in a composite behavior, this allows the collective to move on to the next action.

3 EXPERIMENTAL METHODS

3.1 Agent Model: The Kilobot Robot

We use the Kilobot robot [15] as a basis for our agents. The Kilobot, shown in Fig. 1, is a miniature mobile robot with a circular body of diameter 33 mm, developed specifically for use in collectives.

Motion. The Kilobot is capable of noisy locomotion in a straight line at approximately 1 bodylength/s and turning on the spot in both directions, completing a full turn in approximately 10 s. Kilobots are individually calibrated for motion, but in practice straight-line motion is inaccurate over long ranges, and both straight-line and turning speeds exhibit significant variation among units.

Communication. Kilobots can communicate with each other locally, transmitting to and receiving from neighbors within a 3 body length radius. Each robot is assigned a unique ID at initialization,

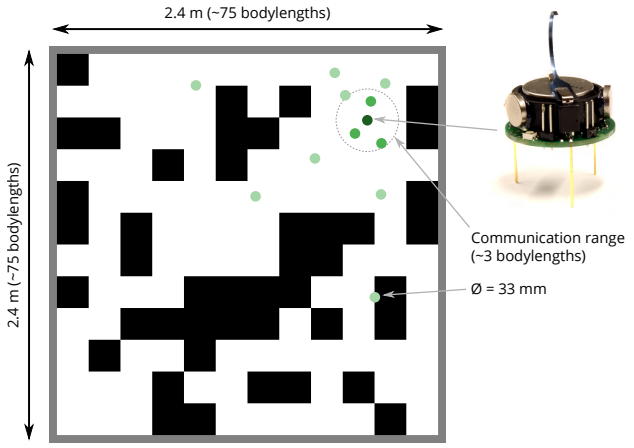

Figure 1: An example of Kilobots in a 2.4×2.4 m (75×75 bodylengths) arena. Kilobots can communicate to others within a radius of approximately 3 body lengths.

allowing their messages to be identified by receivers. They are capable of communicating at a rate of up to 10 messages/s, although this rate decreases at higher robot densities.

Light Sensing. The only environmental sensor on the Kilobot is an ambient light sensor. Therefore, in our setup we use light to represent the environmental features to be estimated.

3.2 Simulation Platform

To perform experiments in simulation, We extended a simulator for the Kilobots originally developed by Espinosa and Rubenstein [5]. This simulator can run experiments with 100 Kilobots 20 x faster than real-time. It implements the kinematics of the Kilobot's motion described above, as well as 'pseudo-physical' collision resolution, in which Kilobots turn on the spot if they collide with each other. The ambient light sensor is emulated by directly feeding the Kilobot the light intensity at its position, and for single feature estimation, we use black and white (i.e., minimum and maximum brightness) areas to represent the two feature values. For multi-feature estimation, we virtually extended the light sensor to detect three different colors: red, green, and blue (see Fig. 2). All simulations were conducted in a 2.4×2.4 m environment (approximately 75×75 bodylengths). This environment is padded by a 50 mm thick border with a gray light value, such that the agents have a means of knowing then they have left the environment.

3.3 Physical Platform

Physical Kilobot robots move using two vibration motors, based on the principle of stick-slip locomotion. Their communication channel is implemented using an infrared transceiver located at the bottom of each robot. Channel sharing is achieved using a CSMA/CD protocol. The light sensor on the physical robots reports the ambient light intensity with a 10-bit resolution.

Physical experiments were run on a whiteboard surface of 1.2×1.2 m An overhead projector was mounted over the surface, which allowed us to project a light pattern onto the surface. The Kilobot's light sensor is sensitive enough to distinguish three levels of brightness: we used two levels to implement the pattern (black and white), and one to define the boundary of the arena (gray). The thresholds for distinguishing light levels were automatically calibrated for each robot before each experiment. An overhead camera was used to record the experiments.

4 SINGLE-FEATURE DECISION MAKING

4.1 Algorithm

The goal of the single feature algorithm is for the collective of agents to combine their noisy estimates of a binary-valued environment feature, arrive at a consensus over its value, and lock in a final decision in finite time; that is, 100% of the agents must agree on the same answer. Our algorithm consists of five components that we detail below. An overview of the agent behavior is shown in Fig. 3. In the following description, we represent the binary-valued feature in terms of colors, with black and white corresponding to the values 0 and 1, respectively.

1. Individual Motion. For the duration of the experiment, agents move in a random walk, with a straight component drawn from an exponential distribution with a mean of 240 s, followed by an on-the-spot rotation sampled uniformly from $[-\pi, \pi)$ rad. If an agent enters the gray border region, it returns to the arena by turning until it no longer detects gray. Recall that agents move forward at 1 bodylength/s and turn at approximately 0.63 rad/s.

2. Estimate and Confidence. An agent makes an *estimate* of the feature in the environment during a 60 s *observation window* by counting the time spent detecting black or white; it pauses its observation timer during any time spent in the gray border region. At the end of an observation period, the agent computes the ratio of white (n_1) to total ($n_0 + n_1$) observation duration. The *confidence* in this estimate is set to be minimum when the observation duration ratio is 0.5, and maximum when it is 0 or 1. It scales linearly between these points. Formally, we define:

$$e = \text{round}\left\{\frac{n_1}{n_0 + n_1}\right\}, \quad c = \frac{\max\{n_0, n_1\}}{n_0 + n_1}$$

An agent then enters a *dissemination period* during which it sends messages containing its ID and feature estimate. The duration of the dissemination period is set to $c \times 120$ s; that is, the more confident an agent is in its estimate, the longer it disseminates it. An analog of this concept is demonstrated in nature, such as the waggle dance in honeybees [16]. Following a dissemination period, an agent begins a new observation period.

3. Belief. During both observation and dissemination periods, an agent receives estimates from other agents. It stores the agent ID and estimate in memory for 180 s. If an agent is heard from more than once within 180 s, only its most recent estimate is kept. At the end of a dissemination period, an agent computes its belief to match the majority of estimates in its 180 s memory, selecting a random belief if the count of each is equal or maintaining its current belief if its memory is currently empty. In essence, the agent is integrating information over the space covered by its neighbors' random walks. In every dissemination period except for the first one, the agent

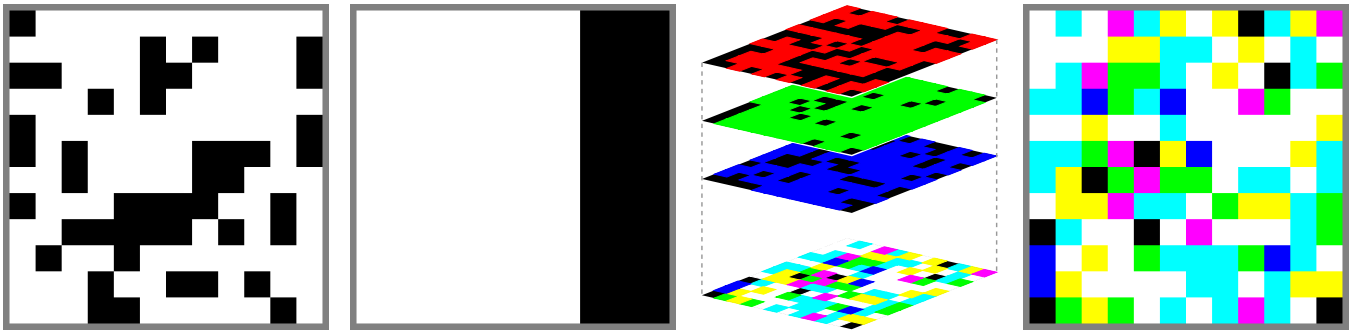

Figure 2: *Far left:* Single feature homogeneous distribution with a 0.7 fill ratio (i.e., proportion of white cells). *Middle left:* Single feature non-homogeneous distribution with a 0.7 fill ratio. *Middle right:* Generation of a multi-feature environment by overlaying 3 single-feature distributions. *Far right:* multi-feature environment with RGB fill ratios (0.55, 0.8, 0.65). Colors are combined according to the standard RGB color model.

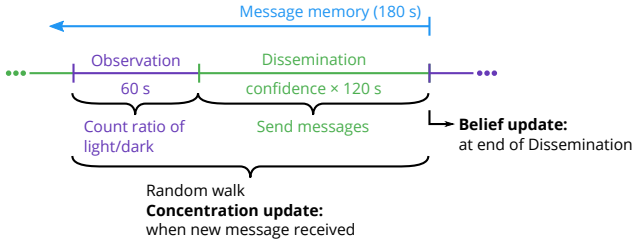

Figure 3: Timeline representing agent behavior during the collective decision making algorithm.

transmits messages containing its belief in addition to its ID and estimate.

4. Concentration. Each agent also maintains a belief concentration C of the feature, which is a moving average that represents its understanding of the collective's feature belief. The concentration is initialized at 0.5 and can range from $0-1$. When an agent receives a message containing a belief b, it updates its concentration if the sending agent is not stored in memory, for a new concentration C^*:

$$C^* = 0.9C + 0.1b$$

The concentration represents an integration of the spatial belief over time, forming a longer-term history than the transient beliefs.

5. Decisions. When the concentration for a feature crosses a threshold of 0.1 from the extrema and remains there for 30 s, an agent makes a non-reversible decision about the feature. A concentration below 0.1 results in a decision of 0 (mostly black), while a concentration above 0.9 yields a decision of 1 (mostly white). Increasing this threshold of After making a decision on the feature, an agent changes from disseminating its belief to disseminating its decision; agents receiving this value interpret it the same as a belief and use it to update their concentrations. This causes positive feedback that will increase the average concentration of the collective and push additional agents toward a decision.

Our algorithm above—in particular components 2 and 3—builds on the work of Valentini et al. [23]. The most important distinction is that our algorithm uses local rather than global communication. It also changes some stochastic computations (e.g., the observation and dissemination period duration) into deterministic ones; pilot experiments confirmed that this does not degrade performance. The notions of concentration and decision-making are not present in [23]; we introduced these inspired by quorum sensing in natural collective systems [11, 25].

4.2 Simulation Results

We tested our algorithm in simulation on both (quasi-)homogeneous and non-homogeneous feature distributions. In a homogeneous feature distribution, each individual agent estimate arising from a random walk is expected to represent a good approximation of the true value. As the environment becomes less homogeneous, individual agent estimates are expected to become, on average, less reflective of the true value, and the variance among them is expected to increase.

Recall that in our environment, the feature values 0 and 1 correspond to the colors black and white, respectively. Therefore, the fill ratio r of the feature is given by the proportion of white area present within the environment.

4.2.1 Homogeneous Feature Distribution. To create homogeneous feature distributions, the square environment was divided into a grid of 12×12 cells of equal size. To create a fill ratio of $r \in [0, 1]$, each cell was randomly assigned a value of black or white with probabilities $1 - r$ and r, respectively, independently of the other cells; an example is shown in Fig. 2.

We ran simulations with 100 agents, which covers approximately 1.5% of the environment. The initial distribution of agents was equally spaced within the environment with random orientations; at this density, agents are roughly 6 bodylengths apart and must move in order to communicate with each other. We conducted 10 simulations for various fill ratios ranging from 0.5 to 0.9. We also tested fill ratios below 0.5 to verify the symmetry of the decision making, but for clarity we only present the upper half of the range.

Table 1: Physical Experiment Results

Trial	Time to first decision (min.)	Time to last decision (min.)
1	7:20	42:15
2	7:15	41:00
3	19:35	53:20
4	8:15	34:25
5	11:30	28:10
Mean (SD):	10:47 (5:13)	39:50 (9:25)

Fig. 4 shows the results of 10 simulations for various fill ratios. The mean feature estimate and belief stabilize within minutes, but with an average higher belief than estimate. The individual agents' concentrations rise more slowly over time; in higher fill ratios with a higher mean belief, the concentration rises faster and this leads to faster decisions. For fill ratios of at least 0.53, no incorrect decisions are made; when the fill ratio is at least 0.6, all agents reach the correct decision within the 150 min. trial.

With a fill ratio of 0.5, no collective decision is made. The mean belief of approximately 0.5 results in concentrations that do not reach the threshold at either extreme. This means that our algorithm does not perform symmetry-breaking for truly ambiguous features, which may or may not be desirable, depending on the application.

4.2.2 Non-Homogeneous Feature Distribution. We implement a non-homogeneous feature distribution simply by splitting the environment into two strips, one of each color; an example is shown in Fig. 2. To achieve a fill ratio of $r \in [0, 1]$, the division line is set such that the area of the white strip as a fraction of the whole area is r, while the remaining $1 - r$ fraction of the whole area is black.

We conducted the same experiments as in the homogeneous feature distribution case. Fig. 5 shows that the resulting mean estimate and belief are lower than for the homogeneous environment, resulting in slower decision-making. Agents were also less capable of classifying fill ratios closer to 0.5, with the collective only consistently making complete decisions for fill ratios of at least 0.7.

The results in Fig. 5 (far left) confirm the expectation that individual agent estimates in a non-homogeneous environment will exhibit a larger variance than in a homogeneous environment. In our two-section environment, the agents can only make accurate estimates of the fill ratio when their random walk is close to the color interface. A random walk that happens to spend most of its time in one of the two areas will heavily bias the estimate towards that area, and will incorrectly increase the agent's confidence in its estimate. This exacerbates the propagation of the noise in the estimates into the beliefs, as shown in Fig. 5 (middle left); in turn, this leads to slower and less accurate decision making.

4.3 Physical Results

We conducted experiments with 30 physical robots in a 1.2×1.2 m environment. In each of 5 trials, we projected onto the surface a randomly-generated homogeneous environment using a grid of 8×8 equally-sized cells (as in Sec. 4.2.1). The fill ratio was 0.7. This physical environment has 25% the area of the simulation environment, but uses 50% as many robots. The parameters of the random walk conducted by the robots was modified from the simulation case to have a straight component drawn from an exponential distribution with mean 60 s and a turning component drawn from a uniform distribution between $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right)$ rad. This shorter, more correlated random walk allowed robots to move more quickly after colliding (in the simulator, collided robots become unstuck quickly due to the pseudo-physical collision resolution).

For each trial, we recorded the time after which the first agent made a decision, and the time until all the agents had made a decision. The results are shown in Table 1. No robots made a wrong decision in any of the trials. The collective successfully classified the environment in, on average, less than 40 min., with the first decision appearing, on average, in just under 11 min.. There was significant inter-trial variability in both the time for the first robot to reach a decision and the time for all robots to decide. This can likely be attributed to the variation in the feature distributions between randomly-generated environments, as well as the random nature of the Kilobots' motion.

The physical robots exhibited some differences from their simulated counterparts. Their movement was less consistent than in simulation, with a straight-line movement that curved to varying degrees, and an inconsistent turning speed. When robots collided, they often failed to separate unless they changed to a turning state, resulting in transient clusters of robots in the environment. Both of these factors increase the locality of robot movement and decrease mixing. In addition, the simulator did not account for the noisy light sensing that was observed on the physical Kilobots. These differences in movement and sensing likely combine to decrease the accuracy of the robots' estimates and beliefs. A video of the physical experiments is available in the online supplementary material [1]; note that due to some minor imperfections in the surface, and the Kilobots' locomotion mechanism (stick/slip with vibration motors), some robots become stuck and rotate around a single point.

5 MULTI-FEATURE DECISION MAKING

5.1 Algorithm

The algorithm for multi-feature decision-making extends that for single-feature decision making, with each agent keeping a belief, concentration, and decision for each of the three color features in our simulation. Each agent observes a single feature and disseminates its estimate of that feature in addition to the index of the feature. Agents receive and store estimates for all features from other agents, which they use to update beliefs for each feature at the end of their own dissemination period. Agents then transmit all beliefs in their future messages. New messages containing beliefs will therefore trigger a concentration update for all features. A decision on each feature is made from its respective concentration, independently of the other features.

Feature Switching. On its own, the algorithm described above would be extremely sensitive to the initial allocation of agents to features; in the worst case, a feature would never be decided upon if no agents are allocated to it. Intuitively, it would make sense to allocate more agents to features that are harder to decide. However,

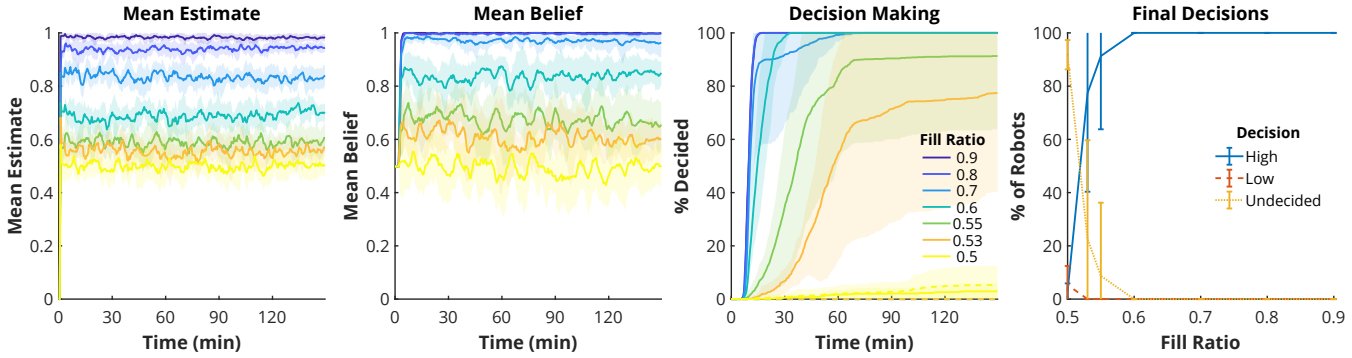

Figure 4: Simulated single-feature decision-making with a homogeneous feature distribution (shading and bars indicate standard deviation.) *Far left:* Mean feature estimate for all agents over time, for differing fill ratios. *Middle left:* Mean belief for all agents over time. *Middle right:* Mean percentage of robots that have made a decision over time. *Far right:* Percentage of robots at end of trial that decided high (white), low (black), or remained undecided.

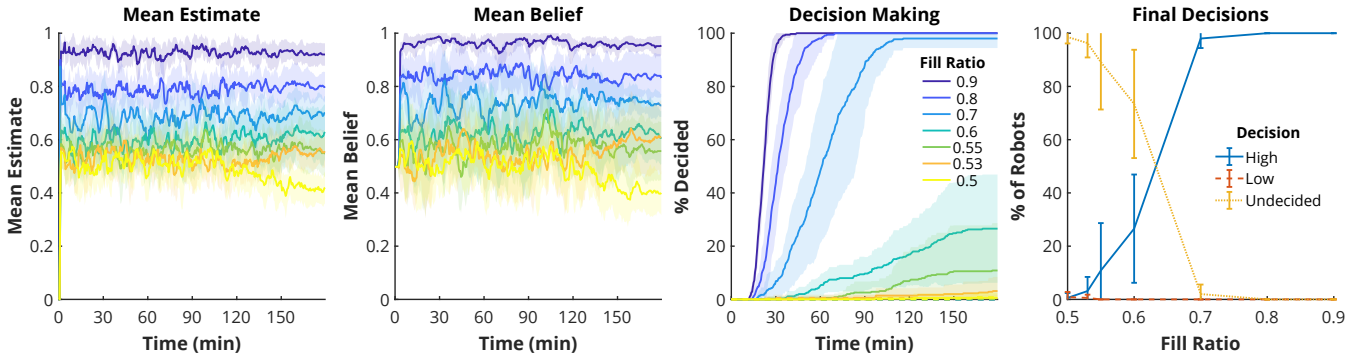

Figure 5: Simulated single-feature decision-making in an environment with a non-homogeneous feature distribution. (Shading and bars show standard deviation.) *Far left:* Mean feature estimate for all agents over time, for differing fill ratios. These are lower (closer to 0.5) than in homogeneously distributed environments. *Middle left:* Mean belief for all agents over time. These are also closer to 0.5 than in homogeneously distributed environments. *Middle right:* Mean percentage of robots that have made a decision over time. Compared to homogeneous environments, decision making was slower, and fewer agents were able to make decisions with fill ratios closer to 0.5. *Far right:* Percentage of robots at end of trial that decided high (white), low (black), or remained undecided. After 180 min., fewer robots made decisions for fill ratios closer to 0.5 than in the homogeneous case.

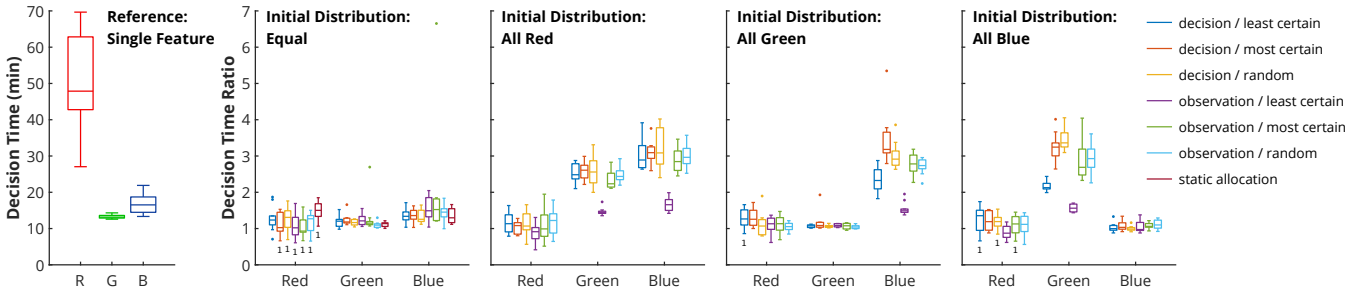

Figure 6: Comparison of time to reach 98% decided in multi-feature decision making, with different approaches to task-switching. Boxes show median and 25th and 75th percentiles; dots are outliers. Numbers below boxes are the number of trials (out of 10) that did not reach 98% decided during the 150 min. trial. *Far left:* Time to decide with 100 agents on a single feature. *Others:* Decision time as a ratio of reference time for each feature. No feature-switching technique shows an advantage when the agents are initially equally distributed between the features. When all agents begin detecting a single feature, easy features are decided faster when they switch to the least certain feature after each observation.

we assume that no *a priori* information is available about this, and we therefore introduce a dynamic task allocation mechanism into the algorithm. While agents can only estimate one feature at a time, they are allowed to switch between estimating different features. We consider two options for when this switching may happen: either before each observation period, or only after a decision has been made on the current feature. Moreover, we consider three possibilities for choosing which feature to switch to: the feature that has a concentration closest to 0.5 (the least certain feature), the feature with a concentration furthest from 0.5 (the most certain feature), or a random feature. An agent may not switch to a feature on which it has already made a decision. If there are no more undecided features, an agent remains allocated to its current feature.

5.2 Simulation Results

To create multi-feature environments, we overlayed single-feature environments. Three homogeneous feature distributions were first independently generated as described in Sec. 4.2.1, but with the color white replaced with one of red, green, or blue. These three feature distributions were then 'added' together so that every cell contained between zero and three colors (inclusive). This process is depicted in Fig. 2 (right two), where the color combinations are represented visually according to the standard RGB color model.

The collective's task now is to decide whether each of the red, green, and blue fill ratios is below or above 0.5. The three fill ratios were chosen so as to provide features of varying difficulties for the collective to decide on: red fill ratio = 0.55 (hard); green fill ratio = 0.8 (easy); and blue fill ratio = 0.65 (intermediate).

We investigated six feature switching laws as discussed in the previous section, based on two possibilities for when the agents are allowed to switch between features, and three possibilities for which feature they switch to:

(1) Agents may switch after deciding on the current feature to the least certain of the undecided features
(2) Agents may switch after deciding on the current feature to the most certain of the undecided features
(3) Agents may switch after deciding on the current feature to a random undecided feature
(4) Agents may switch before each observation period to the least certain of the undecided features
(5) Agents may switch before each observation period to the most certain of the undecided features
(6) Agents may switch before each observation period to a random undecided feature

For each switching law, we ran simulations with four initial agent-to-feature allocations: one with an equal allocation of agents to each feature, and three simulations with all the agents allocated to a single feature. For each switching law and initial allocation (16 combinations in total), we ran 10 simulation trials with 100 agents using randomly-generated feature distributions.

We compare the multi-feature decision-making time to a baseline of 100 agents deciding on each feature, a case of single feature decision making from Sec 4. When agents are initially distributed equally between the features, no feature-switching strategy shows a clear advantage over others. Notably, all remain close to the reference time to reach decisions.

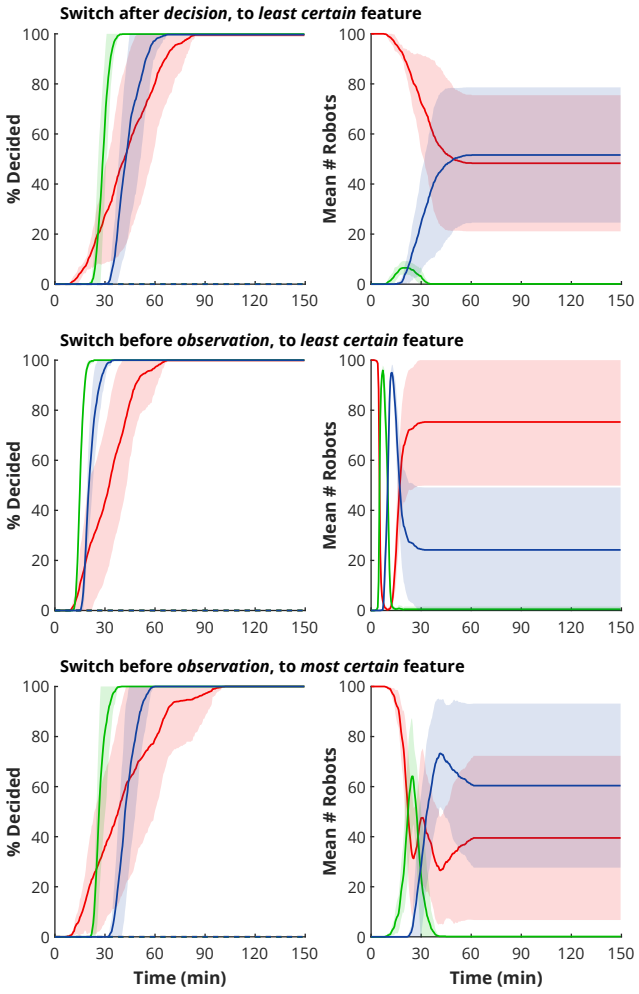

Figure 7: Decision-making progress for different feature switching conditions, with all agents initially allocated to red. (*Left:* Decision-making progress for all features. Shading represents standard deviation. *Right:* Allocation of agents between features.)
***Top:* Agents switch to their least certain undecided feature after making a decision about their current feature. A few agents decide for red and switch to green and blue, both of which are decided faster than red. Fewer than ten agents were allocated to green for all agents to make a decision.**
***Middle:* Agents switch to their least certain undecided feature before every observation period. Agents are more quickly re-allocated to blue and green for a short period of time, resulting in quicker decisions than when feature switching only occurs after decisions.**
***Bottom:* Agents switch to their most certain undecided feature before every observation period. More agents end up allocated to blue than when agents change the least certain feature, reducing the accuracy of beliefs about red and prolonging the feature's decision time.**

However, when agents are not initially equally distributed, there is an advantage in decision-making time when agents switch to the least certain feature at each observation period. For the easier features (blue and green), this strategy produces quicker decisions when agents are not initially allocated to those features.

We investigate the reason for this advantage in Fig. 7, which demonstrates the changes in allocation and decisions over the simulations when agents are initially allocated to red. This is an expansion of Column 3 in Fig. 6.

Switching features for each observation period instead of after a decision results in faster decision making because agents are more quickly reallocated to uncertain features. Looking at the agent allocation in the middle row of Fig. 7, we see that agents detect red for the first observation cycle, then switch almost entirely to green (which, having not been previously observed, is the least certain feature with a concentration of 0.5). Most agents make a decision on green before their next observation period because of the large number of agents dedicated to the task. They then switch to blue (also unobserved and therefore with a concentration near 0.5) and make a fast decision before switching back to red. In contrast, in the top row of Fig. 7, where agents switch features only after decisions, changes in allocation are much slower + However, in this scenario there is no cost for switching features; if such a cost existed (for example, if robots had to move to a new location or replace their sensor for a different feature), the benefit of switching between observation periods could be negated.

Switching to the least certain feature produces an advantage because it prevents over-allocation of agents to easy features. This can be seen in the contrast between the bottom 2 rows of Fig. 7, where agents switch to the most or least certain feature for each observation period. Counter-intuitively, agents that switch to the most certain feature remain on red until some agents have made a decision on the feature; belief updates will push the concentration above 0.5 and make it perceived as the most certain feature. This delays decision making on other features until agents this perceived easy feature is decided by a few agents. However, after agents move to and quickly decide green, they disproportionately switch to blue instead of red (at approximately 30 min. in the bottom right of Fig. 7). If they decide red while observing blue, they will remain detecting blue. Compared to agents switching to the least certain feature, agents are over-allocated to blue. This reduces the accuracy of estimates (and by extension, beliefs) for red, prolonging the decision process.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we introduced and investigated the *multi-feature collective decision making problem*. Our algorithm uses only local communication, and is able to consistently make a correct unanimous decision in finite time, even on features that are almost completely ambiguous. The algorithm can correctly classify a number of features simultaneously in a multi-feature environment. This holds even if the algorithm is presented with pathological initial agent-to-feature allocations, thanks to a dynamic task allocation mechanism. We examined different types of task switching rules, and identified the one that works best over various initial allocations.

In future work, we intend to implement multi-feature collective decision making where the features are fundamentally different from each other. For instance, the agents could be required to evaluate the color fill ratio, entropy, and curvature of the environment, as described in Sec. 2. This will also be implemented on larger swarms of Kilobot robots.

ACKNOWLEDGMENTS

This research was funded by a DARPA DSO grant, a Wyss Institute for Biologically Inspired Engineering Technology Development Fellowship (Gauci), and a Department of Energy Computational Science Graduate Fellowship under grant number DE-FG02-97ER25308 (Ebert).

REFERENCES

[1] The authors. 2017. Online supplementary material. (2017). https://www.dropbox.com/sh/9ievpc800luiehx/AABgNFcFJugt0PJVueoNFZNfa?dl=0
[2] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. 2013. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence* 7, 1 (2013), 1–41.
[3] Scott Camazine. 2003. *Self-organization in biological systems*. Princeton University Press.
[4] Jean-Louis Deneubourg and Simon Goss. 1989. Collective patterns and decision-making. *Ethology Ecology & Evolution* 1, 4 (1989), 295–311.
[5] G. Espinosa and M. Rubenstein. 2018. Using Hardware Specialization and Hierarchy to Simplify Robotic Swarms. In *Proceedings of the 2018 IEEE International Conference on Robotics and Automation*.
[6] Nigel R Franks, Eamonn B Mallon, Helen E Bray, Mathew J Hamilton, and Thomas C Mischler. 2003. Strategies for choosing between alternatives with different attributes: exemplified by house-hunting ants. *Animal Behaviour* 65, 1 (2003), 215–223.
[7] Deborah M Gordon. 1996. The organization of work in social insect colonies. *Nature* 380, 6570 (1996), 121–124.
[8] Deborah M. Gordon and Natasha J. Mehdiabadi. 1999. Encounter rate and task allocation in harvester ants. *Behavioral Ecology and Sociobiology* 45, 5 (1999), 370–377.
[9] José Halloy, Grégory Sempo, Gilles Caprari, Colette Rivault, Masoud Asadpour, Fabien Tâche, Imen Saïd, Virginie Durier, Stéphane Canonge, Jean Marc Amé, et al. 2007. Social integration of robots into groups of cockroaches to control self-organized choices. *Science* 318, 5853 (2007), 1155–1158.
[10] Heiko Hamann, Bernd Meyer, Thomas Schmickl, and Karl Crailsheim. 2010. *A Model of Symmetry Breaking in Collective Decision-Making*. Springer, 639–648.
[11] Melissa B Miller and Bonnie L Bassler. 2001. Quorum sensing in bacteria. *Annual Reviews in Microbiology* 55, 1 (2001), 165–199.
[12] Chris AC Parker and Hong Zhang. 2009. Cooperative decision-making in decentralized multiple-robot systems: The best-of-N problem. *IEEE/ASME Transactions on Mechatronics* 14, 2 (2009), 240–251.
[13] Kevin M Passino and Thomas D Seeley. 2006. Modeling and analysis of nest-site selection by honeybee swarms: the speed and accuracy trade-off. *Behavioral Ecology and Sociobiology* 59, 3 (2006), 427–442.
[14] Elva JH Robinson, Ofer Feinerman, and Nigel R Franks. 2009. Flexible task allocation and the organization of work in ants. *Proceedings of the Royal Society of London B: Biological Sciences* (2009).
[15] Michael Rubenstein, Christian Ahler, and Radhika Nagpal. 2012. Kilobot: A low cost scalable robot system for collective behaviors. In *Proceedings of the 2012 IEEE International Conference on Robotics and Automation*. 3293–3298.
[16] Thomas D Seeley. 1995. *The wisdom of the hive: the social physiology of honey bee colonies*. Harvard University Press.
[17] Thomas D Seeley and Susannah C Buhrman. 2001. Nest-site selection in honey bees: how well do swarms implement the" best-of-N" decision rule? *Behavioral Ecology and Sociobiology* 49, 5 (2001), 416–427.
[18] Thomas D Seeley and P Kirk Visscher. 2004. Group decision making in nest-site selection by honey bees. *Apidologie* 35, 2 (2004), 101–116.
[19] Thomas D Seeley and P Kirk Visscher. 2004. Quorum sensing during nest-site selection by honeybee swarms. *Behavioral Ecology and Sociobiology* 56, 6 (2004), 594–601.
[20] Thomas D Seeley, P Kirk Visscher, Thomas Schlegel, Patrick M Hogan, Nigel R Franks, and James AR Marshall. 2012. Stop signals provide cross inhibition in collective decision-making by honeybee swarms. *Science* 335, 6064 (2012), 108–111.
[21] Ana Sendova-Franks and Nigel R Franks. 1993. Task allocation in ant colonies within variable environments (A study of temporal polyethism: Experimental).

Bulletin of Mathematical Biology 55, 1 (1993), 75–96.

[22] Guy Theraulaz and Eric Bonabeau. 1999. A brief history of stigmergy. *Artificial Life* 5, 2 (1999), 97–116.

[23] Gabriele Valentini, Davide Brambilla, Heiko Hamann, and Marco Dorigo. 2016. Collective Perception of Environmental Features in a Robot Swarm. In *Proceedings of ANTS 2016: Tenth International Conference on Swarm Intelligence*. 65–76.

[24] Gabriele Valentini, Eliseo Ferrante, Heiko Hamann, and Marco Dorigo. 2016. Collective decision with 100 Kilobots: speed versus accuracy in binary discrimination problems. *Autonomous Agents and Multiagent Systems* 30, 3 (2016), 553–580.

[25] Christopher M Waters and Bonnie L Bassler. 2005. Quorum sensing: cell-to-cell communication in bacteria. *The Annual Review of Cell and Developmental Biology* 21 (2005), 319–346.