Bayes Bots: Collective Bayesian Decision-Making in Decentralized Robot Swarms

Julia T. Ebert[1], Melvin Gauci[1], Frederik Mallmann-Trenn[2], and Radhika Nagpal[1]

***Abstract*— We present a distributed Bayesian algorithm for robot swarms to classify a spatially distributed feature of an environment. This type of "go/no-go" decision appears in applications where a group of robots must collectively choose whether to take action, such as determining if a farm field should be treated for pests. Previous bio-inspired approaches to decentralized decision-making in robotics lack a statistical foundation, while decentralized Bayesian algorithms typically require a strongly connected network of robots. In contrast, our algorithm allows simple, sparsely distributed robots to quickly reach accurate decisions about a binary feature of their environment. We investigate the speed vs. accuracy tradeoff in decision-making by varying the algorithm's parameters. We show that making fewer, less-correlated observations can improve decision-making accuracy, and that a well-chosen combination of prior and decision threshold allows for fast decisions with a small accuracy cost. Both speed and accuracy also improved with the addition of bio-inspired positive feedback. This algorithm is also adaptable to the difficulty of the environment. Compared to a fixed-time benchmark algorithm with accuracy guarantees, our Bayesian approach resulted in equally accurate decisions, while adapting its decision time to the difficulty of the environment.**

I. INTRODUCTION

In order for groups of robots to cooperate in complex scenarios, they must be able to collectively make choices at multiple decision points. In many cases, this takes the form of a "go/no-go" problem: each robot must select the best of two possible choices based on some incomplete information available to them. In swarms of robots, this challenge is compounded: cooperative behavior relies on all the robots quickly coming to the same decision. This problem is prevalent in potential applications, such as robots collaboratively identifying and eliminating pests in an agriculture field, or a collective of robots deciding where to build a habitat for a future human colony on Mars. Solving this problem may require a decentralized approach if the robots cannot rely on a central process for collecting information and taking decisions. It is challenging to have a large, decentralized group of robots quickly and accurately make collective decisions, especially in the binary go/no-go scenario.

This work was supported by a Department of Energy Computational Science Graduate Fellowship under grant number DE-FG02-97ER25308 and a Wyss Institute Technology Development Fellowship.

[1]Julia T. Ebert, Melvin Gauci, and Radhika Nagpal are with the John A. Paulson School of Engineering and Applied Sciences at Harvard University, Cambridge, Massachusetts and the Wyss Institute for Biologically Inspired Engineering, Boston, Massachusetts {ebert, nagpal}@g.harvard.edu, melvingauci@gmail.com

[3]Frederik Mallmann-Trenn is with the Department of Informatics at King's College London, UK
frederik.mallmann-trenn@kcl.ac.uk

Swarm robotics often draws inspiration from biology for distributed decision-making, since binary decisions are common in biology. Insect-based algorithms are inherently decentralized and scalable, making them well-suited for robotic collectives. Bees and ants are known for house-hunting, in which the entire colony must select between multiple possible new nest sites or risk splitting the colony [1], [2]. These strategies typically use random pairwise interactions and positive feedback to push the group to a decision, in which higher-quality options are more heavily communicated and more visited, pushing the colony to consensus.

Several groups have explored robotic decision-making inspired by insect house-hunting. This approach proved effective in a house-hunting task completed by Kilobot robots [3], [4]. Valentini et al. also applied inspiration from the honeybee waggle dance to an environmental perception task, in which a group of robots reached consensus about whether an environment was colored with mostly black or mostly white squares [5]. However, this approach yields a transient consensus on the majority color of the environment, rather than a group-wide go/no-go commitment. Previously, our group extended this algorithm to allow for committed decisions [6], inspired by quorum sensing in bacteria, where a decision was triggered by a continuous estimation value crossing a threshold. We also improved the speed and accuracy of decisions by employing positive feedback, in which robots communicated their decisions. In both robot house hunting and perception, a trade-off was observed between the decision speed and accuracy in selecting the best choice. However, these bio-inspired algorithms include many parameters and are not mathematically grounded, making it difficult to intuitively understand the speed vs. accuracy tradeoff and make optimal parameter selections.

In contrast, Bayesian algorithms provide a statistically grounded approach to the challenge of sensor- and decision-fusion in distributed sensor networks and multi-agent systems. Many agents collect samples of information that must be integrated to form a single estimate or decision. Early approaches involved distributed sensors, but decision-making was centralized [7]. More recently, decentralized Bayesian information fusion has been successful across a variety of domains, such as target tracking [8], source localization [9], self-localization [10], and event classification [11], demonstrating its broad applicability across tasks and domains. While many of these approaches scale across the number of agents [12], they often rely on assumptions such as maintaining a fixed or strongly connected network[13], or their goal is continuous state estimation, rather than a go/no-go decision.

We look to extend the decentralized Bayesian approach to go/no-go decisions on simple, spatially-distributed robots that communicate locally but are not always connected.

We present a novel Bayesian algorithm for a robot collective to achieve fast, accurate decisions about their environment. We abstractly model the go/no-go decision by tasking robots with classifying monochrome environments as filled with a majority of black or white, as in [5], [6], and shown in Fig. 1; this represents any scalar environmental feature that could be observed by robots. Each robot behaves as a Bayesian estimator, while exchanging and integrating observations from nearby robots. We show that collective decisions are possible even with few assumptions about the capabilities of the robots: a collective of 100 simulated Kilobot robots is able to achieve accurate decisions even when they are sparsely distributed and have locally-limited sensing and communication. We find that positive feedback improves both the speed and accuracy of decisions, and that each robot making fewer observations can improve decision accuracy by reducing their spatial correlations. In addition, a well-chosen regularizing prior allows for a lower decision-making threshold with a small accuracy cost. We also demonstrate that the algorithm's speed naturally adapts to the difficulty of the environment. Finally, we compare this approach to a fixed-time benchmark algorithm that provides theoretical accuracy guarantees even in worst-case environments.

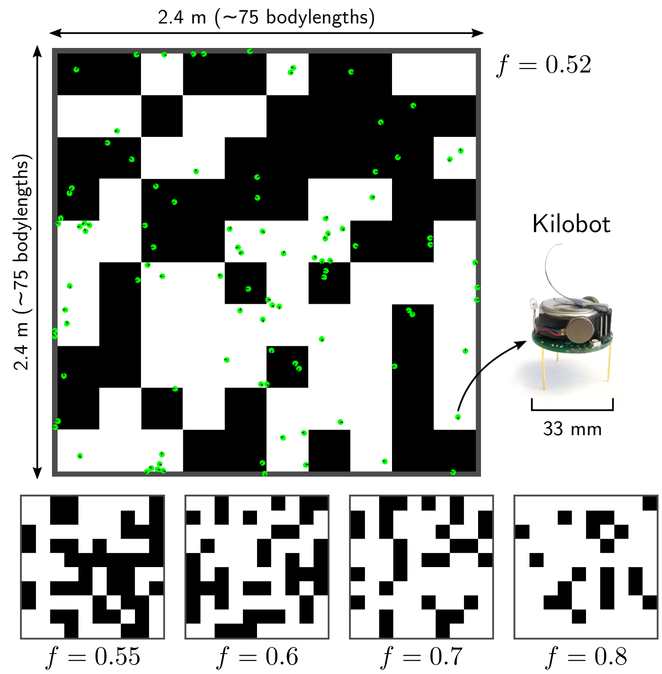

Fig. 1. Examples of simulated environments with different fill ratios f. The goal is for robots to determine whether the environment is mostly white ($f > 0.5$) or mostly black ($f < 0.5$). **Top:** Image from a Kilosim simulation with $f = 0.52$, containing 100 robots, each able to communicate within a radius of 3 bodylengths. **Right:** Kilobot robot, which is the model for the simulated robots. **Bottom:** Example environments with different fill ratios.

II. METHODS

A. Problem Definition

We present a problem in which k robots complete a binary classification task. Robots are placed in a bounded black and white environment, where the proportion of white within the space is the environment's fill ratio f, as shown in Fig. 1. The goal is to collectively decide whether the majority of the environment is filled with black or white (i.e, is the fill ratio above or below 0.5). Because the problem is symmetric, we show results for environments where $f > 0.5$.

Classifying an environment results in a trade-off between the time for all robots to decide and the collective accuracy of the decision. This is particularly pronounced in the most challenging environments, where the fill ratio is close to 0.5; the small difference between black and white area makes it more difficult to distinguish than extreme fill ratios.

This formulation represents an abstraction of real-world problems. In an agricultural application, the white regions would be analogous to pest-damaged areas of a field, with the goal of determining if a field requires pest treatment. Alternatively, the color could represent mineral deposits in a Mars exploration mission. In each case, the goal is to make a go/no-go decision about a single, spatially distributed feature.

B. Robot Model

We investigate this decision-making problem using Kilobots as our model robot platform [14], whose capabilities narrow the complexity of possible decision algorithms. Kilobots are able to sense their environment with an ambient light sensor located on the top of the robot, allowing them to distinguish black, white, and gray regions of an environment projected from above onto a bounded 2D arena. These small robots (33 mm diameter) lack complex bearing or localization capabilities. Therefore, we rely on pseudo-random walks. The robots also have limited communication bandwidth and range; they can broadcast 9-byte messages to robots within approximately 3 bodylengths.

In environmental classification problems, robots are typically sparse. 100 Kilobots cover only 1.5% of the 2.4×2.4 m arena ($\approx 75 \times 75$ bodylengths) available for the physical robots. Therefore, we cannot rely on assumptions required in many distributed algorithms, like maintaining a strongly connected network. However, we demonstrate that it is possible to design robust decision-making algorithms even for robots with limited capabilities.

III. ALGORITHMS

We developed a Bayesian algorithm that allows simulated Kilobot robots to classify black and white environments. The goal was to classify the fill ratio f of the environment as mostly white (a decision of $d_f = 1$) or mostly black ($d_f = 0$). Each robot employs a Bayesian model of the fill ratio and makes decisions using credible intervals of the posterior distribution. We also compare to a benchmark algorithm that provides accuracy guarantees for even the worst-case scenarios, sacrificing speed for accuracy.

A. Bayesian Decision-Making Algorithm

The algorithm followed by each robot is shown in Alg. 1.

Robots make binary color observations C of their environment, which we model as draws from a Bernoulli distribution where the probability of observing white is the fill ratio:

$$C \sim \text{Bernoulli}(f) \tag{1}$$

Each robot models the unknown fill ratio f of the environment as a Beta distribution:

$$f \sim \text{Beta}(\alpha, \beta) \tag{2}$$

resulting in the posterior update for each observation:

$$f \mid C \sim \text{Beta}(\alpha + C, \beta + (1 - C)) \tag{3}$$

Initialization: Robots are placed uniformly in the arena with random orientation. Each robot's prior model is initialized with $\text{Beta}(\alpha, \beta)$, where both parameters are initialized as α_0, a parameter determining how regularizing the prior is. Each robot also sets its observation index $i = 0$.

Movement: For the duration of the trial, each robot performs a pseudo-random walk to cover the arena, defined by segments of movement in a straight line, followed by a random turn. The durations of the straight segments are drawn from an exponential distribution with mean of 240 s, while turns are drawn uniformly from $0 - 2\pi$. This parameterization was previously determined in [6]. The edge of the bounded environment is defined by a gray region, as seen in Fig. 1. If a robot detects gray light, it turns continuously until it exits the border region.

Observation: Each robot makes an observation C every τ seconds: $C = 1$ if white, $C = 0$ if black, and ignoring gray observations. The posterior of the fill ratio is updated with the observation as in Eq. 3 and i increments by 1.

Communication: After a robot makes its first observation, it begins broadcasting its most recent observation index i and observed color C. While continuing to move, observe, and broadcast, all robots also listen for messages from neighboring robots. Upon receiving a new observation, the receiver updates its posterior as with its own observations.

Decision: Each robot checks whether its decision criterion is met after every posterior update. The credible threshold p_c defines the probability mass of the posterior that must lie on one side of 0.5 in order for a decision to be made. If the posterior's cumulative distribution p at 0.5 passes the criterion ($p \geq p_c$), a decision is made that the environment is black ($d_f = 0$), as most of the probability is below 0.5. Conversely, if $(1 - p) \geq p_c$ (i.e., most of the probability mass is above 0.5), the environment is classified as white ($d_f = 1$). *This is the robot's irreversible go/no-go decision.*

After a decision is made, a robot will broadcast its decision in place of its observation if positive feedback (u^+) is used. Otherwise, it will continue to transmit its observations.

This algorithm depends on four parameters:

- **Observation interval** τ (s) is the time between observations, where $\tau > 0$. Shorter observation intervals mean collecting observations quicker, but results in observations that are less spatially distributed. Longer intervals result in more independent observations.
- **Credible threshold** p_c is the minimum probability mass of the posterior that must lie on one side of 0.5 in order to make a decision. We assume $0.5 \leq p_c < 1$. Higher credible thresholds require more observations before enough probability amasses to make a decision.'
- **Prior parameter** α_0 is a positive integer used for both shape parameters of each robot's prior distribution of f. Setting $\alpha_0 = 1$ forms a uniform prior, while $\alpha_0 > 1$ creates a symmetric prior peaked around 0.5. This regularizing prior indicates a lower prior belief that the fill ratio is near 0 or 1, analogous to having previously made $\alpha_0 - 1$ black and $\alpha_0 - 1$ white observations.
- **Positive feedback** u^+ is a boolean indicating whether robots will transmit their decision d_f in place of their most recent observation C after they make decisions. Positive feedback is used effectively for decision-making in insects and bacteria. This feedback reinforces decisions made by robots that decide early, but it may push the group to the wrong decision or split the group if early-deciding robots conflict.

Algorithm 1 Bayesian Decision-Making Algorithm
Input: Observational interval τ, credible threshold p_c, prior parameter α_0, positive feedback indicator u^+, robot UID id
Output: Binary classification of environment d_f

Init counter of white observations $\alpha = \alpha_0$
Init counter of black observations $\beta = \alpha_0$
Init observation index i
Init incomplete decision $d_f = -1$
Init dictionary of received observations $s = \{\text{ID} : (0, 0)\}$
for $t \in [1, T]$ **do**
  Perform pseudo-random walk
  if τ divides t **then**
    $C \leftarrow$ observed color (0, 1)
    $\alpha \leftarrow \alpha + C$
    $\beta \leftarrow \beta + (1 - C)$
    $i \leftarrow i + 1$
  Let $m = (id', i', C')$
  if $s(id') \neq m(id')$ **then**
    $\alpha \leftarrow \alpha + C'$
    $\beta \leftarrow \beta + (1 - C')$
  if $d_f = -1$ **then**
    Let p denote the cumulative distribution function of $\text{Beta}(\alpha + \alpha_0, \beta + \alpha_0)$ at 0.5.
    if $p > p_c$ **then**
      $d_f \leftarrow 0$
    else if $(1 - p) > p_c$ **then**
      $d_f \leftarrow 1$
  if $d \neq -1$ and u^+ **then**
    Broadcast message (id, i, d_f)
  else
    Broadcast message (id, i, C)

While there is intuition behind the trends of these parameters individually, the interactions and optimal choices are unknown. We use a parameter sweep to investigate the effect of parameter values on speed and accuracy, as well as the interactions between the parameters.

B. Benchmark Decision-Making Algorithm

We now describe a fixed-time algorithm for which parameter settings can be derived that guarantee correct decision-making to an arbitrary accuracy in an arbitrary environment of known size, as shown in Alg. 2. Given a worst-case fill ratio that we wish to be able to detect, and a desired accuracy (i.e., tolerance for incorrect decisions), we can compute the number of weakly correlated samples S that a single robot requires to make a decision. If we instead have k robots, robots first independently collects samples (Phase 1), and then disseminate information among each other (Phase 2). The observation phase must be long enough for each robot to collect at least S/k samples; the second phase must be long enough for all pairs of robots to communicate, such that each robot has a total of at least S samples.

1) Phase 1: Sample Collection: We first select a worst-case fill ratio $\hat{f}$ (i.e., how close to 0.5) to be able to distinguish. To make a correct decision, we need enough samples that the sample mean is within $\epsilon = 2 \cdot |f - 0.5|$ of the true fill ratio. For a given confidence level $1 - \delta/2$, we need a total of S uncorrelated[1] samples:

$$S \geq \frac{4\hat{f}(1-\hat{f})Z\left(1-\frac{\delta}{4}\right)^2}{\epsilon^2} \quad (4)$$

using the Z-score of the standard normal distribution. This is derived from the two-tailed $1 - \delta/2$ confidence interval using a Gaussian approximation of the Binomial distribution [15].

The S/k samples each robot collects must be uncorrelated in order for Eq. 4 to hold. If nothing is known about the distribution of colors within the environment, we must design for the worst case, where samples are highly locally-correlated (i.e. a non-homogeneous environment). Then, each robot must move $\tau \geq t_{\text{mix}}$ between samples, where the mixing time t_{mix} is a property of the size and topology of the environment, and the nature of the random walk. Conversely, if the environment is homogeneous (i.e., if each cell is colored independently of its neighbors, as in 1), then the observation interval τ need only be long enough that a robot does not sample more than once from the same grid cell consecutively. Therefore, the Phase 1 duration is $S/k \cdot \tau$.

2) Phase 2: Sample Communication: Each robot now has S/k samples but needs S samples to make an accurate decision. We assume that the robots have IDs, can ignore repeated information, and have a communication radius r_{comm}. When robots A and B are within r_{comm} of each other, A collects and stores B's samples if it has not done so already, and vice-versa. We must now determine how long robots need to move such that each pair of robots has interacted, to some desired confidence level $1 - \delta/2$. This notion is captured by the meeting time, t_{meet}, which is defined as the worst-case expected time for two robots to meet, regardless of their starting location. Note that t_{meet} is a function of: (i) the environment size; (ii) the environment topology; (iii) the nature of the random walk; (iv) the communication radius. To guarantee with probability $1 - \delta/2$ that all pairs of robots have communicated, we require a Phase 2 duration of:

$$t_{\text{comm}} = 2\log\left(\frac{k^2}{\delta}\right) t_{\text{meet}} \quad (5)$$

The probability that two random walks meet after $2t_{\text{meet}}$ steps is, by the Markov inequality, at least $1/2$. Thus, the probability that any two given random walks do not meet after $\log(k^2/\delta)$ intervals of length $2t_{\text{meet}}$ is $\frac{1}{2^{\log(k^2/\delta)}} = \frac{\delta}{k^2}$. Taking the union bound over all $\binom{k}{2}$ pairs gives that the total probability of failure is at most $\frac{k(k-1)}{2}\frac{\delta}{k^2} \leq \frac{\delta}{2}$. [16]. Combining the $\delta/2$ failure risk from each phase, we can guarantee the decision will be correct with probability $1 - \delta$.

Algorithm 2 Benchmark Decision-Making Algorithm
Input: Total communication time t_{comm}, observation interval τ, robot UID id, number of samples S/k
Output: Binary classification of environment d_f

Init counter of white observations $\alpha = 0$
Init counter of black observations $\beta = 0$
Init dictionary of received samples $s = \{\text{ID} : (0,0)\}$
for $t \in \left[1, \frac{S}{k}\tau\right]$ **do**
    Perform pseudo-random walk
    if τ divides t **then**
        $C \leftarrow$ observed color (0, 1)
        $\alpha \leftarrow \alpha + C$
        $\beta \leftarrow \beta + (1 - C)$
$s(id) = (\alpha, \beta)$
for $t \in \left[\frac{S}{k}\tau, \frac{S}{k}\tau + t_{\text{comm}}\right]$ **do**
    if new message (id', α', β') **then**
        $s(id') = (\alpha', \beta')$
    Broadcast message (id, α, β)
Let α_T denote the sum of the α values in s
Let β_T denote the sum of the β values in s
if $\beta_T > \alpha_T$ **then**
    $d_f = 0$
else
    $d_f = 1$

IV. EXPERIMENTS

We conducted experiments testing both algorithms in Kilosim, an open-source Kilobot simulator we developed that is able to run at over 700× real speed for 100 robots [17], allowing us to thoroughly investigate the parameter space. A demonstration video is available on YouTube [18]. All experiments were conducted with 100 robots in a $2.4\text{ m} \times 2.4\text{ m}$ arena. To investigate the performance in settings of varying difficulty, we tested five different fill ratios f: 0.52, 0.55, 0.6, 0.7, 0.8. Fill patterns (as seen in Fig. 1) were generated for each trial by pseudo-randomly filling a 10×10 grid of squares with black or white to match the fill ratio. Trial duration was capped at 50,000 s (≈ 14 hours) each.

[1]As mentioned above, there is a very weak correlation between samples. However, by fine-tuning the time between samples τ, we can make sure that the probability of sampling a white cell is within $f \pm \epsilon/4$. This error is small enough to ensure our calculations hold.

A. Bayesian Algorithm

We conducted a parameter sweep across the following values, running 100 trials for each of the resulting 7,280 parameter combinations.

- τ (s) : $1, 5, 10, 15, 20, 25, 50, 75, 100, 150, 200, 250, 300$
- p_c : $0.9, 0.95, 0.98, 0.99$
- α_0 : $1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 25, 50$
- u^+: True, False

B. Benchmark Algorithm

For the benchmark algorithm, we computed the required time parameters τ and t_{comm} to meet the guarantees of $\delta = 0.1$ (equivalent to the Bayesian $p_c = 0.9$) and $\epsilon = 0.04$, which matches the most difficult environment ($f = 0.52$).

Because the robots' random walk is highly correlated relative to the grid cell size, an upper bound on τ is calculated from the expected time to cross a grid cell. Given a robot speed of 1 bodylength/s and grid cells of approximately 7×7 bodylengths, we selected $\tau = 10$ s, which is the time to cross a cell diagonally. From Eq. 4 have that $S = 2{,}398$ samples, or 24 per robot, resulting in a Phase 1 duration of 240 s.

We computed t_{meet} empirically, because it depends on environment- and robot-specific factors. To match the worst expected meeting time across all possible starting positions, we placed two robots in opposite corners of the arena with random orientation. Over 5,000 trials, we computed the mean time for the robots to first communicate as 1,151 s. From Eq. 5, we therefore set a Phase 2 duration of $t_{\text{comm}} = 26{,}503$ s. We conducted 100 trials with these parameters.

V. RESULTS

We assess the success of the Bayesian decision-making algorithm by considering the speed vs. accuracy trade-off across our parameter sweep. We treat decision-making as a multi-objective optimization problem by comparing the accuracy of decision-making vs. the time for all robots complete decisions in each parameter condition. The optimal parameter selections are those that lie along the Pareto frontier of accuracy and decision time.

We first consider the impact of positive feedback (u^+) in the most difficult environment, where the fill ratio is the most ambiguous at $f = 0.52$. As shown in Fig. 2A, positive feedback is essential for pushing the group to decisions, dramatically improving both the decision accuracy and speed. In many conditions without positive feedback, the entire group was unable to reach decisions within the 14 hour simulation limit, while with positive feedback the worst decision time was under 5 hours. One might expect positive feedback to split the swarm into two groups, resulting in lower overall accuracy; however, this occurs more when positive feedback is *not* used, resulting in a collective accuracy consistently below 70%. This is consistent with both previous robot results in [3], [6], and the use of positive feedback in biological collectives. As a comparison, we look at the bio-inspired algorithm from [6] in the same environment. It lies on the Pareto front, but many parameter choices for the Bayesian algorithm achieve the same accuracy faster.

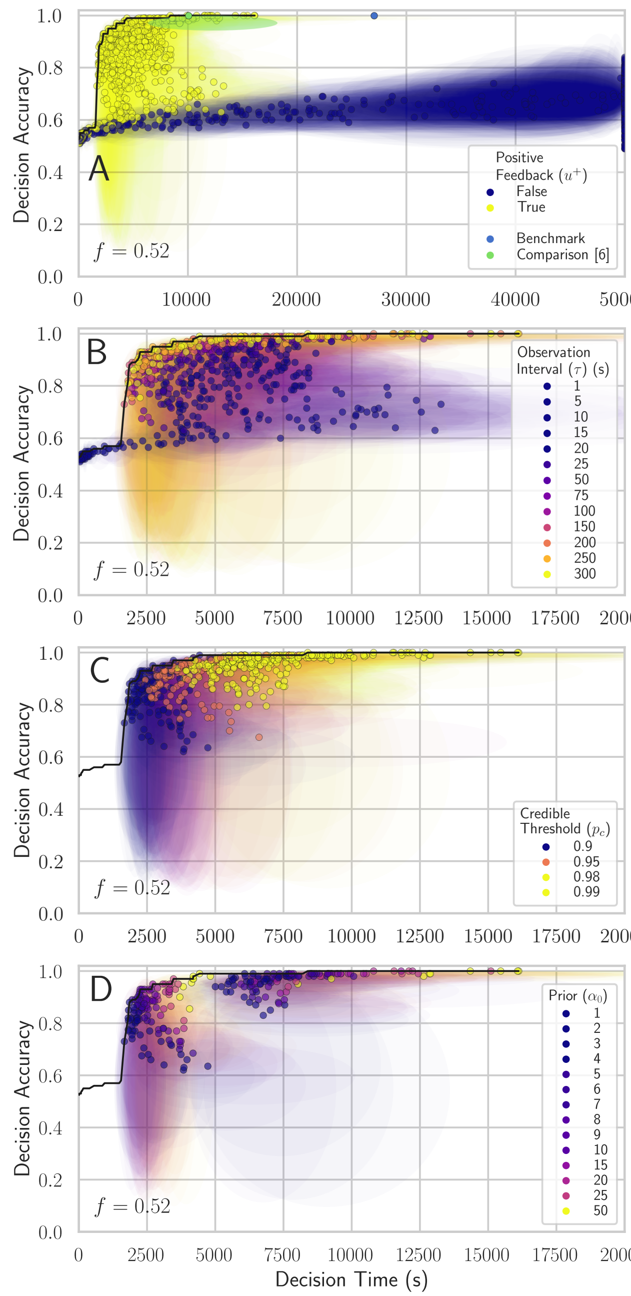

Fig. 2. Speed and accuracy of Bayesian decision-making. Each point represents the median time for all robots to reach a decision and median accuracy of the resulting decisions, over all trials for a particular condition. Ellipses show the 25–75th percentile in each dimension. The black line shows the Pareto front of decision time vs. accuracy. Each successive figure shows a subset of the data from the preceding one. **A:** For a fill ratio of 0.52, decisions were faster and more accurate when positive feedback was used. Comparing to results from [6], certain parameter choices were faster while maintaining high accuracy. The benchmark algorithm exceeded its accuracy guarantees but was slower than comparatively accurate Bayesian parameter combinations. **B:** Longer intervals between observations counter-intuitively resulted in more optimal decisions (showing $u^+ =$ True). **C:** Lower credible thresholds save time with minimal accuracy cost (showing $\tau \geq 15$). **D:** Lower credible thresholds are effective only if a regularizing prior prevents premature decisions (showing $p_c \in \{0.9, 0.99\}$ on left and right, respectively).

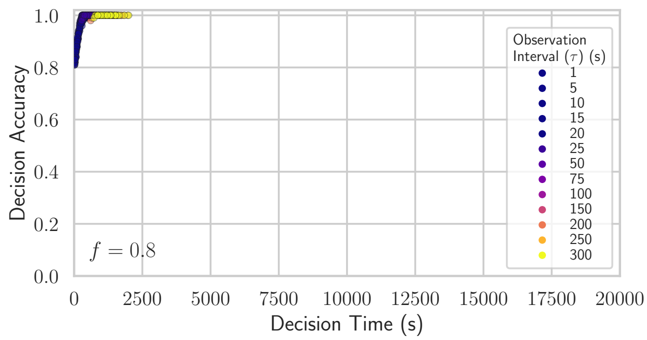

Fig. 3. Analogous plot to Fig. 2B for a fill ratio of 0.8. On the same timescale, decisions were significantly faster and more accurate in this easier environment.

Focusing only on conditions where positive feedback is used, we investigate the impact of the interval between observations (τ), shown in Fig. 2B. Somewhat surprisingly, higher times between observations yields results closer to the Pareto optimal front. While a shorter observation interval yields more total observations, they are highly spatially correlated. A longer interval results in fewer observations but more mixing, therefore resulting in more representative samples and more accurate decisions in a shorter time.

Given the benefits of positive feedback and longer observation intervals, we now look at the effect of credible threshold (p_c), shown in Fig. 2C. An intuitive pattern emerges that higher credible thresholds produce higher accuracy, but choosing the highest threshold of 0.99 can incur a large time cost. In Fig. 2D, we see when this occurs by contrasting the decision results for $p_c = 0.9$ and $p_c = 0.99$. Here it becomes apparent that there is an interaction between the the choice of prior and the credible threshold. When a sufficiently large regularizing prior is used, the credible threshold can be low because the prior prevents premature decisions.

The benchmark algorithm consistently exceeded its 90% accuracy guarantee, as seen in Fig. 2A. The high ratio of time spent communicating vs. observing also underlines the previously-observed benefit of collecting fewer, less correlated samples. However, meeting this algorithm's worst case guarantees incurs a time cost in comparison to many Bayesian parameter configurations. It requires sufficient time for enough direct pairwise robot communications, rather than forming a multi-hop network by re-transmitting observations. However, this accurately reflects the constraints in the modeled robot system, whether bandwidth limitations and channel capacity prevent effective re-communication. Fig. 3 also shows that the Bayesian algorithm is adaptively faster in an easier environment, with $f = 0.8$. The benchmark algorithm performed perfectly, but still took 26,743 s ($\approx$ 7.5 hours) for all robots to reach a decision, because its fixed time is determined by the most difficult fill ratio.

VI. CONCLUSIONS AND FUTURE WORK

We have introduced a decentralized Bayesian algorithm (Alg. 1) that allows simple, sparsely spaced robots to achieve accurate classifications of an environmental feature. With well-selected parameters, the robots were able to achieve this go/no-go conclusion even when the difference between black and white fill was only 4%. When the feature distinction was greater, decision speed and accuracy significantly improved, while becoming less sensitive to parameter choice. This adaptability makes this approach suitable for applications where little *a priori* knowledge is available about the feature under consideration, in contrast with the benchmark algorithm, where providing guarantees for the worst case requires pre-selecting a longer decision time for all environments. However, robots using the Bayesian algorithm do not know whether others have made a decision, in contrast to the guarantee of decisions after the benchmark's fixed duration.

The Bayesian algorithm is also tunable; for example, if an expected fill ratio is known, an informed prior can be selected. If the accuracy requirements are lower in a particular case, the credible threshold could be lowered to speed up decisions. Positive feedback increasing decision speed and accuracy also demonstrates that bio-inspiration can be beneficial when used with statistically-grounded decision models, rather than as an alternative approach.

We also showed that it is possible to create an algorithm with accuracy guarantees for simple robots without knowing the environment's difficulty, but this incurred a trade-off in total decision time, across all environment difficulties. The benchmark algorithm's guarantees may also be fragile in the real world; for example, robot failure after starting would violate the requirement of k robots and result in insufficient observations. In practice, we have shown high accuracy can be consistently achieved with the Bayesian algorithm, without the explicit guarantees of the benchmark.

In future work, we plan to extend the Bayesian algorithm to modularly construct a decision-making framework, through the choice or robot communication and sampling. The positive feedback used here represents a simple form of informed communication, in which robots select information to communicate to facilitate others' decisions. With increased communication bandwidth, robots could re-transmit messages to form a multi-hop network to speed the spread of information. In addition, selecting what to subset of information to communicate can help to solve decision-making problems where noisy or probabilistic observations limit the value of individual samples. Robots capable of localization would also be able to employ multi-agent adaptive sampling to improve the utility of samples to input into their distributions, improving on the Kilobots' random walk. Localization would also allow robots to solve target or source localization problems. These extensions would make the Bayesian decision-making algorithm applicable to a broad class of spatially-distributed decision-making problems.

ACKNOWLEDGMENT

J. T. E. thanks Richard Barnes for assistance with Kilosim that made large-scale simulations possible.

References

[1] S. C. Pratt, E. B. Mallon, D. J. T. Sumpter, and N. R. Franks, "Quorum sensing, recruitment, and collective decision-making during colony emigration by the ant Leptothorax albipennis," *Behav. Ecol. Sociobiol.*, vol. 52, no. 2, pp. 117–127, July 2002.

[2] T. D. Seeley and S. C. Buhrman, "Group decision making in swarms of honey bees," *Behav. Ecol. Sociobiol.*, vol. 45, no. 1, pp. 19–31, 1999.

[3] G. Valentini, E. Ferrante, H. Hamann, and M. Dorigo, "Collective decision with 100 Kilobots: speed versus accuracy in binary discrimination problems," in *Auton. Agent. Multi. Agent. Syst.*, vol. 30, no. 3. Springer New York LLC, May 2016, pp. 553–580.

[4] G. Valentini, H. Hamann, and M. Dorigo, "Efficient decision-making in a self-organizing robot swarm: On the speed versus accuracy trade-off," in *Proc. Int. Jt. Conf. Auton. Agents Multiagent Syst. AAMAS*, vol. 2, 2015, pp. 1305–1314.

[5] G. Valentini, D. Brambilla, H. Hamann, and M. Dorigo, "Collective perception of environmental features in a robot swarm," in *Int. Conf. on Swarm Intelligence*, vol. 9882. Springer Verlag, 2016, pp. 65–76.

[6] J. T. Ebert, M. Gauci, and R. Nagpal, "Multi-feature collective decision making in robot swarms," in *Proc. 17th Int. Conf. Auton. Agents Multiagent Syst. (AAMAS 2018)*. IFAAMAS, 2018, pp. 1711—-1719.

[7] I. Y. Hoballah and P. K. Varshney, "Distributed Bayesian Signal Detection," *IEEE Trans. Inf. Theory*, vol. 35, no. 5, pp. 995–1000, 1989.

[8] S. Bandyopadhyay and S. J. Chung, "Distributed estimation using Bayesian consensus filtering," in *Proc. Am. Control Conf.* Institute of Electrical and Electronics Engineers Inc., 2014, pp. 634–641.

[9] T. Zhao and A. Nehorai, "Distributed sequential Bayesian estimation of a diffusive source in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 55, no. 4, pp. 1511–1524, apr 2007.

[10] D. Fontanella, M. Nicoli, and L. Vandendorpe, "Bayesian localization in sensor networks: Distributed algorithm and fundamental limits," in *IEEE Int. Conf. Commun.*, 2010.

[11] M. Alanyali, S. Venkatesh, O. Savas, and S. Aeron, "Distributed Bayesian hypothesis testing in sensor networks," in *Proc. 2004 Am. Control Conf.* IEEE, 2004, pp. 5369–5374 vol.6.

[12] A. Makarenko and H. Durrant-Whyte, "Decentralized Bayesian algorithms for active sensor networks," *Inf. Fusion*, vol. 7, pp. 418–433, 2006.

[13] P. Landgren, V. Srivastava, and N. E. Leonard, "Distributed cooperative decision-making in multiarmed bandits: Frequentist and bayesian algorithms," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 167–172.

[14] M. Rubenstein, C. Ahler, and R. Nagpal, "Kilobot: A low cost scalable robot system for collective behaviors," in *Proc. - IEEE Int. Conf. Robot. Autom.* Institute of Electrical and Electronics Engineers Inc., 2012, pp. 3293–3298.

[15] S. Wallis, "Binomial confidence intervals and contingency tests: Mathematical fundamentals and the evaluation of alternative methods," *J. Quant. Linguist.*, vol. 20, no. 3, pp. 178–208, Aug 2013.

[16] D. Aldous and J. A. Fill, "Reversible Markov Chains and Random Walks on Graphs," 2002. [Online]. Available: https://www.stat.berkeley.edu/~aldous/RWG/book.pdf

[17] J. Ebert and R. Barnes, "Kilosim," Sep 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3406865

[18] J. Ebert, "Bayes bots simulation [video]," 2020. [Online]. Available: https://youtu.be/O9C3wAPh1IA