Collective Decision Making in Swarm Robotics with Distributed Bayesian Hypothesis Testing

Chapter · October 2020

DOI: 10.1007/978-3-030-60376-2_5

Qihao Shan
Otto-von-Guericke-Universität Magdeburg

Collective Decision Making in Swarm Robotics with Distributed Bayesian Hypothesis Testing

Qihao Shan(✉) and Sanaz Mostaghim

Faculty of Computer Science,
Otto von Guericke University Magdeburg, Magdeburg, Germany
{qihao.shan,sanaz.mostaghim}@ovgu.de

Abstract. In this paper, we propose Distributed Bayesian Hypothesis Testing (DBHT) as a novel collective decision-making strategy to solve the collective perception problem. We experimented with different sampling and dissemination intervals for DBHT and concluded that the selection of both intervals presents a trade-off between speed and accuracy. After that, we compare the performance of DBHT in simulation with that of 3 other commonly used collective decision-making strategies, DVMD, DMMD and DC. We tested them on collective perception problems with different difficulties and feature patterns. We have concluded that DBHT outperforms considered existing algorithms significantly in collective perception tasks with high difficulty, namely close proportion of features and clustered feature distribution.

1 Introduction

Collective decision making has been a longstanding topic of study within swarm intelligence. The aim of this research area is to explain how groups of natural intelligent agents make decisions together, as well as to construct decision-making strategies that enable groups of artificial intelligent agents to come to a decision. The problems being investigated usually require the agents to form a collective decision using only their individual information and local interaction with their peers. There are two categories of problems that are primarily investigated within collective decision making, consensus achievement and task allocation. In the former category, agents need to form a singular opinion, while in the latter category, agents need to be allocated to different tasks.

In this paper, we address the problem of collective perception, which is a discrete consensus achievement problem. Morlino et al. [8] introduced this problem in 2010. Many collective decision-making strategies have been adopted to address this problem. Valentini et al. proposed Direct Modulation of Voter-based Decisions (DMVD) in [16]. Valentini et al. also proposed Direct Modulation of Majority-based Decisions (DMMD) in [15], and further analyzed it in [14]. Direct Comparison (DC) of option quality was proposed by Parker et al. in [9].

M. Dorigo et al. (Eds.): ANTS 2020, LNCS 12421, pp. 55–67, 2020.
https://doi.org/10.1007/978-3-030-60376-2_5

These strategies usually draw inspirations from natural systems and focus on consensus forming among robots of different opinions.

In this paper, we are proposing Distributed Bayesian Hypothesis Testing (DBHT) as a novel method to perform collective perception. Our method takes inspiration from sensor fusion techniques used in sensor networks, where Bayesian reasoning has been widely used. Hoballah et al. [6] and Varshney et al. [17] have designed various decentralized detection algorithms based on Bayesian hypothesis testing in sensor networks. Alanyali et al. explored how to make multiple connected noisy sensors reach consensus using message passing based on belief propagation [1]. Such algorithms usually require strong and fixed communication between sensor nodes and are thus rarely used for swarm robotics. We have improved upon this characteristics by adopting a self-organizing communication topology. In our method, individual robots first form their estimation of the likelihood of various hypotheses by observing their immediate surrounding environment. Then a leader periodically collects opinions from other robots, and forms the final estimate of the whole swarm. In this paper, we will evaluate the performance of our proposed decision-making strategy and compare it to that of DMVD, DMMD and DC.

This paper is structured as follows. Sect. 2 presents previous works related to this paper and details about benchmark decision-making strategies. Sect. 3 provides the mathematical derivation and detailed description of our proposed algorithm. In Sect. 4, we show our experiments and evaluations of the results. Finally, Sect. 5 contains the conclusion and future work.

2 Problem Statement and Related Works

We follow Valentini et al. [13], Strobel et al. [11] and Bartashevich et al.'s [2] definition of the collective perception problem. There is an arena with a number of tiles that can either be black or white. The goal is to determine whether black or white tiles are in the majority using N mobile robots. The robots are assumed to have rudimentary low level control and perform random walk to sample the environment. They can only observe the color of the tile directly beneath themselves. We also assume that the robots have a maximum communication radius and can only perceive and communicate with their peers within the radius. The collective perception problem was first proposed by Morlino et al. [8]. The current form of the problem was established by Valentini et al., who explored the performance of several widely used collective decision-making algorithms, DMVD, DMMD and DC, in solving the collective perception problem [13]. Strobel et al., in their exploration of collective decision-making performance with Byzantine robots, tested the performance of DMVD, DMMD and DC on environments with different proportion of black tiles [11]. Ebert et al. extended collective decision making to environments with more than 2 colors [4]. Bartashevich et al. proposed novel benchmarks for collective decision-making task. They showed that apart from the proportion of black and white tiles, the pattern of them also have a great impact on the performance of collective decision-making strategies [2]. Ebert et al. then also proposed a novel collective decision-making strategy based on Bayesian statistics [3]. In this paper, we closely examine DMVD, DMMD and DC and test how their performances compare to that of our proposed algorithm.

DMVD and DMMD implement decision making by individual robot as a probabilistic finite state machine [15,16]. When applied to the collective perception problem, each robot can be in one of 4 states, exploration $E_i, i \in \{B, W\}$ and dissemination $D_i, i \in \{B, W\}$. B and W indicates the color that the robot thinks is in the majority. Both strategies start by randomly assigning half of the robots to state E_B and the other half E_W. In the exploration states, a robot samples the environment at every control loop on its own and computes the quality of its current opinion $\rho_i \in (0, 1]$. The durations of exploration states are random and exponentially distributed with a mean of σ sec. In the dissemination states, a robot broadcasts its opinion to its neighbors. The duration of dissemination states is exponentially distributed with a mean of $\rho_i g$, which is proportional to the quality of the robot's opinion. Here we follow [11,13] and set σ and g to 10 sec. Random walk routine as used in [13] is executed in all states. The process will continue until all robots' opinion become the same. DMVD and DMMD differ in their behaviors during the dissemination states. When using DMMD strategy, a robot takes the opinion favored by a majority of its neighbors plus itself. When using DMVD strategy, a robot takes the opinion of a random neighbor.

DC uses similar probabilistic finite state machines on an individual level to achieve collective decision making, except that the mean length of dissemination states is no longer $\rho_i g$ but g [9]. Thus all states have a mean duration of 10 s in our case. During dissemination, a robot will broadcast both its opinion and its quality estimate of the opinion ρ_i. At the end of dissemination states, a robot will compare its own quality estimate ρ_i with that of a random neighbor ρ_j. If $\rho_i < \rho_j$, the robot will switch its decision to j.

3 Distributed Bayesian Hypothesis Testing

In this section, we introduce our proposed approach Distributed Bayesian Hypothesis Testing (DBHT). In order to obtain an estimation of the proportion of white tiles, we model the environment as a discrete random variable with 2 possible states $V = White/Black$. $P(V = White) = P_{\mathrm{W}}$ is the probability that a random place in the arena is white, thus the proportion of white tiles in the arena. At a given point of time, N robots each have made S observations of the tile colors beneath themselves. The observations can be either black or white, and we label $N \times S$ observations as $ob1, 1 ... ob_{\mathrm{N,S}}$.

Given a hypothesis of the proportion of white tiles, $P_W = h$, we compute its likelihood given the past observations made, i.e., $P(P_W = h | ob_{1,1} ... ob_{\mathrm{N,S}})$. We first use Bayes rule:

$$P(P_W = h | ob_{1,1} ... ob_{N,S}) = \frac{P(ob_{1,1} ... ob_{N,S} | P_W = h) P(P_W = h)}{P(ob_{1,1} ... ob_{\mathrm{N,S}})} \tag{1}$$

Here $P(P_W = h)$ is the prior and $P(ob_{1,1}...ob_{\mathrm{N,S}})$ is the marginal likelihood, both of which we assume to be the same for all hypotheses. In this case, we can apply the chain rule to $P(ob_{1,1}...ob_{\mathrm{N,S}}|P_W = h)$

$$= P(ob_{1,1}|P_W = h)P(ob_{1,2}|P_W = h, ob_{1,1})...P(ob_{\mathrm{N,S}}|P_W = h, ob_{1,1}, ..., ob_{\mathrm{N,S\text{-}1}}) \tag{2}$$

Assuming the observations are all independent from each other, we have:

$$= P(ob_{1,1}|P_W = h)P(ob_{1,2}|P_W = h)...P(ob_{\mathrm{N,S}}|P_W = h) \tag{3}$$

Thus the likelihood of a hypothesis can be computed by multiplying the conditional probability of each observation given the hypothesis. An individual robot's estimate based on its observations and hence its opinion can be computed as follows:

$$Op_1 = P(P_W = h|ob_{1,1}...ob_{1,\mathrm{S}}) \tag{4}$$

$$= P(ob_{1,1}|P_W = h)...P(ob_{1,\mathrm{S}}|P_W = h)P(P_W = h) \tag{5}$$

Therefore, the estimate of the whole swarm can be computed by calculating the product of the opinion of individual robots:

$$P(P_W = h|ob_{1,1}...ob_{\mathrm{N,S}}) = Op_1 Op_2 ... Op_N \tag{6}$$

Numerically, we set 10 hypotheses expressed in the following matrix. The first column is the proportion of white tiles (P_W) and the second column is the proportion of black tiles ($1 - P_W$).

$$H = \begin{bmatrix} 0.05 & 0.95 \\ 0.15 & 0.85 \\ \cdots & \\ 0.95 & 0.05 \end{bmatrix} \tag{7}$$

An observation can be either $ob = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ for white or $ob = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ for black tiles. Therefore, the opinion of individual n can be calculated:

$$Op_n = \Pi_{s=1..S} H \cdot ob_s \tag{8}$$

Algorithm 1: Distributed Bayesian Hypothesis Testing

```
Result: MaxIndex(Op*)
n ∈ [1..N] Index of the robot itself, Leader has n = 1
Op_n = [0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]
Neigh(n) = [] List of Neighbors of robot n
State_n = 0, t=0, Randomly Place Robots in the Arena
Define T_S, T_D, MaxCommDistance, M_max= Max Number of Neighbors
while Max(Op*) < 0.99 do
    if State_n = 0 then
        Random Walk
        if t%T_S = 0 then
            Collect Observation ob
            Op_n = Op_n ∘ (H · ob)
        if t%T_D = 0 & is Leader then
            State_n = 1
    else if State_n = 1 then
        for m=1..N do
            if Distance(n, m) < MaxCommDistance & State_m = 0 & m ≠ n
             & length(Neigh(n) < M_max) then
                Append Neigh(n) with m; Append Neigh(m) with n
                State_m = 1
        State_n = 2
    else if State_n = 2 then
        if is Leader then
            if Messages from all Neighbors Received then
                Op* = Normalize(Op_1 ∘ Π_{m∈Neigh(1)} Message_{m → 1} )
                State_n = 3
        else
            if Messages from all Neighbors except m Received then
                Message_{n → m} =
                 Normalize(Op_n ∘ Π_{x∈Neigh(n)\{m}} Message_{x → n})
                State_n = 3
    else
        if is Leader or Any Neighbor is in State 0 then
            State_n = 0 ; Neigh(n) = []
    t = t + 1
```

Fusion of the opinions of the individual is done through a self-organizing communication network with a tree topology. To construct such topology, each robot's behavior is designed as a finite state machine with 4 states as shown in Algorithm 1. One robot is designated as the leader, which is tasked with both observing its immediate vicinity and collecting the opinion of other robots. The leader can be chosen by the user or elected in a self-organizing way, e.g. as in [12], before the decision making process. In our experiments, the robot with index 1 is designated as the leader. All other robots are also assigned an unique index. More in details, all robots start in state 0, where they perform random walk routine the same as in [13] and modify their own opinion by sampling the color of the ground beneath themselves as shown below. Sampling is done periodically with the interval T_S. Individual robot's opinion is computed iteratively, thus only opinion after the last sampling need to be stored, making the memory and computation complexity of our method $O(N_H M_{\max})$, where N_H is the number of hypotheses and $M_{\max}$ is the maximum number of neighbors a robot can have.

$$Op_{\mathrm{n,s}} = Op_{\mathrm{n,s\text{-}1}} \circ (H \cdot ob_{\mathrm{n,s}}) \tag{9}$$

$$Op_{\mathrm{n,0}} = \left[0.1\ 0.1\ 0.1\ 0.1\ 0.1\ 0.1\ 0.1\ 0.1\ 0.1\ 0.1\right]^T \tag{10}$$

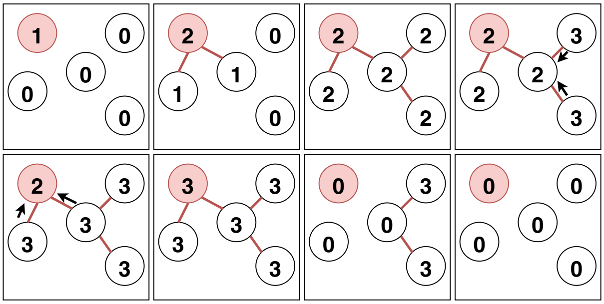

Fig. 1. Illustration of states (numbers), communication links and message passing during dissemination.

At every dissemination interval T_D, the leader will start a dissemination session. It switches to state 1, stops moving and sends out signals to look for robots nearby. Only robots within communication distance that are in state 0 will respond, to ensure that the final communication topology has a tree structure. They will establish connection with the leader and switch to state 1 too. They will also stop moving and send out signals to look for neighbors themselves, and the process continues. After searching for neighbors, a robot will go into state 2 and prepare for the transmission of messages. It is likely that some robots cannot be reached by the network due to maximum communication distance and number of neighbors. Robots in state 2 will perform message passing. It is done similarly to belief propagation algorithm. Message from robot n_1 to n_2 is an array of 10 numbers and defined as the follows.

$$Message_{n \to m} = Op_n \circ \mathit{\Pi}_{x \in \mathrm{Neigh}(n) \backslash \{m\}} Message_{\mathrm{x}} \to n \quad (n \neq 1) \tag{11}$$

In practice, messages are normalized before being sent to avoid underflow. Once a robot sends its message, it switches to state 3 and rest. Message passing starts from the leave nodes and gradually converges towards the leader. The leader will compute the estimate of the whole swarm as follows.

$$Op^* = Op_1 \circ \mathit{\Pi}_{\mathrm{m} \in \ \mathrm{Neigh}(1)} Message_{\mathrm{m}} \to 1 \tag{12}$$

Once Op^* is computed, the leader will switch back to state 0 and send out signals to its neighbors. The neighbors will switch to state 0 and send out signals as well. The process continues until all robots are in state 0. Communications thus stop and will start again at the next dissemination session. There is therefore no need for the robots to be in constant communication with each other. A new topology of the communication network will be constructed at every new dissemination session. The illustration of robot states during a dissemination session is shown in Fig. 1. Op^* will converge to one of the 10 hypotheses eventually. The algorithm stops when one of the hypotheses has a normalized likelihood of at least 0.99. The leader can then report the result to the user or direct the swarm to perform other tasks dependent on the decision.

4 Experiments

In this section, we describe our experiments and analyze the results. The setting of our experiments are largely the same as [2,11,13]. The arena is $2\,\mathrm{m} * 2\,\mathrm{m}$ with 400 tiles. We use 20 simulated e-puck robots [7] to perform the designated task. As for the maximum communication radius, we follow the settings proposed in [2,11] and set it 50 cm. E-pucks can only perceive up to 5 neighbors simultaneously [5] and can communicate with up to 7 neighbors [7], therefore we can set $M_{\max}$ to 5. In addition, they cannot receive multiple messages simultaneously [10], therefore setting $M_{\max}$ to 2 greatly reduces the probability of communication failure. We have tested both limits and the results are shown in Sect. 4.2 and 4.3. The low level random walk routine we used is the same as described in [13]. We use the same Matlab simulation environment as [2] to simulate the robots and arena.

Table 1. Average error using different sampling and dissemination intervals

/10%	Sampling interval T_S/s					
Dissem interval T_D /s	0.1	0.2	0.5	1	2	5
1	0.3	0.216	0.072	0.032	0.026	0.036
2	0.278	0.204	0.064	0.038	0.022	0.028
5	0.208	0.124	0.044	0.028	0.018	0.024
10	0.17	0.128	0.052	0.01	0.018	0.024
20	0.094	0.076	0.026	0.026	0.016	0.006

4.1 Finding the Optimal Sampling and Dissemination Interval

The first set of experiments aims at determining the optimal sampling and dissemination interval, T_S and T_D. It is expected that reducing the sampling interval could provide more samples per unit time. However, these samples will be collected closer to each other, thus highly correlated. It then means that the independence assumption used in Sect. 3 will not closely reflect reality. Therefore the algorithm will produce less accurate result. On the other hand, increasing the

Table 2. Average decision time using different sampling and dissemination intervals

/s	Sampling interval T_S /s					
Dissem interval T_D/s	0.1	0.2	0.5	1	2	5
1	3.09	4.75	9.53	15.80	29.66	65.82
2	3.84	6.13	11.08	18.17	32.72	69.62
5	7.36	9.22	14.98	23.11	39.08	80.05
10	12.43	14.59	21.06	30.06	45.12	87.06
20	22.23	23.87	30.07	39.95	57.40	103.64

dissemination interval means it takes longer on average for the leader to connect to a large number of robots. However, the robots can travel for a longer distance and collect more samples without violating the independence assumption during the time. Therefore the result will be more accurate. To determine the optimal values for T_S and T_D, we run 100 tests each for scenarios with white tile proportions of 0.55, 0.65, 0.75, 0.85, 0.95, and random distribution of white tiles. We compare the average error and consensus time across different T_S and T_D. Error is defined as the difference between the true proportion of white tiles and the computed proportion by the swarm.

Our results (Tables 1 and 2) show that as T_S increase from 0.1 to 1 s, there is a significant decrease in error. This is because the speed of our robots is 0.16 m/s, thus for 1 s, a robot would have traveled for 0.16 m. This is a bit bigger than the width (0.1 m) and diagonal length (0.141 m) of an individual tile, meaning that when collecting the next sample, the robot would have moved to the neighboring tile. Since the distribution of tiles is random, there is a very weak correlation between the colors of adjacent tiles, thus moving to the neighboring tile is enough to reduce the correlation to near zero. This is why when T_S is beyond 1 s, increasing it no longer reduces the error much but still increases the decision time. We thus choose 1 s as the optimal T_S. This also agrees with the findings of Ebert et al. [3], that collecting less correlated samples sparsely can improve the accuracy of decision making. However, when T_S is 5 s, there is an increase in error compared to 2 s, since too high a T_S means too few sample would be collected, thus impeding accurate decision making. At the same time, increasing T_D provides moderate improvements on accuracy and in exchange moderate increase in decision time. A trade-off need to be considered when applying our method to a real problem. Here we choose a T_D of 5 s to be used in later experiments.

4.2 Comparison with Other Collective Perception Algorithms

To determine the effectiveness of our algorithm and how it compares to other state-of-the-art algorithms, we follow the experimental framework of Bartashevich et al. [2] and perform simulations of all considered algorithms for different difficulty ($\rho_b^* = number_{\text{black tiles}}/number_{\text{white tiles}}$) as well as different patterns of black and white tiles. For difficulty, we use the same test cases as in [2,11], $\rho_b^* = [0.52, 0.56, 0.61, 0.67, 0.72, 0.79, 0.85, 0.92]$. White color is always kept in majority. The patterns are selected from the patterns used for matrix visualization and classified according to entropy (E_c) and Moran index (MI). E_c describes the densities of clusters in the pattern and MI describes the level of connectivity between clusters. See [2] for detailed definition. For every ρ_b^* and pattern, the simulation is run 100 times. The performance of the proposed algorithms is measured by the exit probability. It is the probability that the swarm will come to the correct decision that white color is in majority. For DBHT, a result is considered correct when the computed hypothesis for white tile proportion is among 0.55, 0.65 ... 0.95. The test results are shown in Fig. 2. The first row shows an example of each pattern tested. The second row shows the exit probability for every algorithm considered. The shaded area indicates the standard deviation of our measurement of exit probability. It is computed theoretically by treating the exit probability as the mean of 100 Bernoulli trials. Thus the standard deviation is $\sqrt{p(1-p)/100}$, where p is the measured exit probability. The third row shows the mean decision time for every algorithm considered. The shaded area indicates the standard deviation of all the samples.

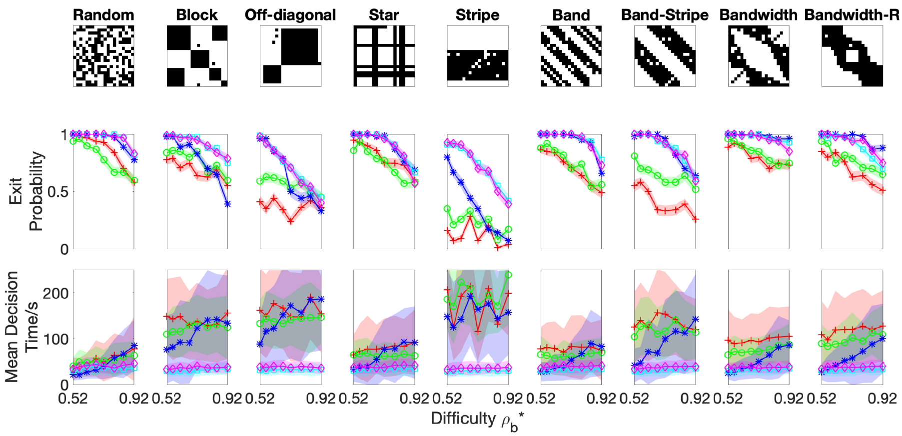

Fig. 2. Exit probability and decision time for all algorithms in 9 patterns over various difficulties. Red+:DMVD, Green○:DMMD, Blue∗:DC, Cyan□:DBHT $M_{\max} = 5$, Magenta◇:DBHT $M_{\max} = 2$ (Color figure online)

It can be observed that maximum neighbor limit only has a small impact on the performance of DBHT algorithm. Exit probability when $M_{\max}$ is 2 and 5 are usually very close. Decision time is slightly longer when the limit is 2.

Random ($E_c \approx 0.5, MI \approx 0$) pattern is the most studied pattern in previous works. All considered algorithms have comparable performance when ρ_b^* is low, with DC having the lowest decision time. As ρ_b^* increase, DMVD and DMMD have a significant drop in accuracy and a rise in decision time. The accuracy of

DBHT and DC is more resilient to the difficulty increase, however, DC also has a significant increase in decision time.

Star ($E_c \approx 0.8, MI \approx 0.4$) and **Band** ($E_c \approx 0.7, MI \approx 0.3$) pattern are observed to be more challenging than random pattern for collective perception. DMVD and DMMD have lower accuracy and higher decision time compared to DC and DBHT. DC and DBHT are comparable in accuracy. They are also comparable in decision time when the difficulty is low, but DC's decision time increases significantly when the difficulty is high.

Bandwidth ($E_c \approx 0.9, MI \approx 0.6$) and **Bandwidth-R** ($E_c \approx 1, MI \approx 0.7$) see DBHT outperforming DMVD and DMMD in both accuracy and decision time. However its accuracy is not as high as DC especially at the highest tested difficulty of 0.92. In terms of decision time, DBHT and DC have similar performance at low difficulty, but the decision time of DC quickly rises as difficulty increases.

Block ($E_c \approx 0.9, MI \approx 0.8$), **Off-diagonal** ($E_c \approx 0.9, MI \approx 0.8$), **Stripe** ($E_c \approx 1, MI \approx 0.8$) and **Band-Stripe** ($E_c \approx 0.9, MI \approx 0.6$) are observed to be the most difficult collective perception scenarios, with highly clustered black tiles. In these scenarios, existing algorithms often becomes very inaccurate with long and volatile decision time. DBHT is able to outperform existing algorithms in terms of accuracy while maintaining a relatively constant decision time.

Overall, DBHT is able to produce higher perception accuracy compared to existing algorithms for the scenarios with high task difficulty. In terms of decision time, DBHT can be slower than existing algorithms, especially DC, when the task is simple. However, it is much faster than existing algorithms when the task is difficult both in terms of high ρ_b^* and clustered feature patterns, as the decision time of DBHT is largely independent of the 2 measurements of difficulty. This is because both DMVD and DMMD make use of modulation of positive feedback to enable decision making. Clustering of features forms local echo chambers among robots with similar opinions, making consensus forming within the swarm difficult. For DC, clustering of features can create robots with extreme quality estimation of its own opinion and thus rarely adopt its neighbors' opinion, therefore disrupting decision making of the swarm. In contrast, DBHT's operation is mostly unaffected by clustering of features. Its opinion fusion is also able to produce a middle ground between robots with highly contrasting estimates. In addition, if collective perception is to be applied in the real world, ρ_b^* and feature patterns are usually unknown. Thus the volatile decision time of the three existing algorithms causes a halting problem. When the algorithm is running for a long time with no clear decision, there is a dilemma whether to keep the algorithm running for even longer time, or to recognize a failed run and restart the process, wasting past progress.

4.3 Estimation Accuracy and Effects of Limiting Maximum Neighbors

The performance of DBHT can also be measured by the difference between the most likely hypothesis computed by the swarm and the correct hypothesis that is closest to the true P_W. Among our test cases, we classify ρ_b^* value of $[0.52, 0.56, 0.61]$ to hypothesis $P_W = 0.65$ and the rest to $P_W = 0.55$. The average errors for all tested scenarios and different $M_{\max}$ are shown in the top row in Fig. 3.

Across all patterns, the error usually spikes at ρ_b^* value of 0.61 or 0.67. These ρ_b^* are in the middle of 2 classes which can cause some error during classification. In addition, the 2 plotted curves are very close to each other and thus a change in $M_{\max}$ in DBHT does not significantly impact the accuracy of collective perception. This is because although a low $M_{\max}$ does reduce the average number of opinions that can be collected as shown in the bottom row in Fig. 3, to meet the likelihood threshold on the final chosen hypothesis of 0.99, the robots have to collect more samples over longer periods of time, as shown in Fig. 3 (middle row). Therefore, the limit on maximum number of neighbors gives a trade-off between decision time, design complexity and robustness of the system.

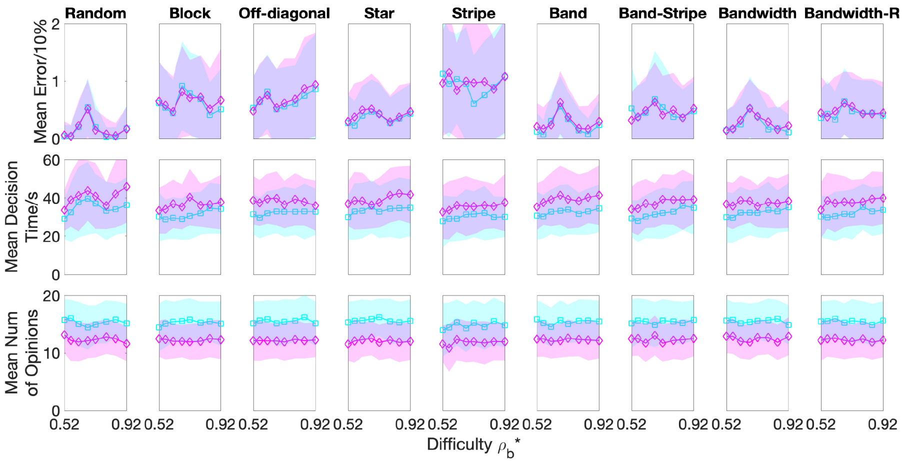

Fig. 3. Mean error, mean decision time, and mean number of opinions collected for DBHT with different $M_{\max}$, Cyan □:DBHT $M_{\max} = 5$, Magenta ⋄:DBHT $M_{\max} = 2$ (Color figure online)

5 Conclusion

In this paper, we have proposed DBHT as a novel collective perception strategy. We have argued that a distributed perception but centralized opinion fusion strategy for decision making is easier to use in the swarm robotics than a fully decentralized decision-making strategy used in most state-of-the-art strategies. After that, we tested the performance of DBHT with different sampling and dissemination intervals. We have concluded that, up to a limit, collecting sparse and uncorrelated samples could increase perception accuracy but also increase the decision time. Changing the dissemination interval presents a similar trade-off. We compared DBHT's performance with that of 3 other state-of-the-art collective decision-making strategies, DMVD, DMMD and DC, in how well they determine which color is in the majority. We have shown that DBHT often has superior performance due to its resilience in high ρ_b^* and feature patterns with large clusters, as well as its stable decision time regardless of the difficulty of the environment. Finally, we examined DBHT's ability to accurately estimate the proportion of colors as well as the effect of limiting the maximum number of neighbors during dissemination. We have found out that the error not only generally increase with ρ_b^* and pattern difficulty, but also tend to spike around boundary cases between classes. Also, the limit on maximum number of neighbors does not have a significant impact on estimation accuracy. The decrease in number of opinions that can be collected is made up by a longer decision time, which means more uncorrelated samples.

In future work, we plan to utilize Bayesian reasoning further to model the correlation between observations. Here we circumvent this issue by having a large sampling interval. Taking correlation between samples into consideration can make decision making more robust when a decision is needed on a short time frame.

References

1. Alanyali, M., Venkatesh, S., Savas, O., Aeron, S.: Distributed Bayesian hypothesis testing in sensor networks. In: Proceedings of the 2004 American Control Conference, vol. 6, pp. 5369–5374. IEEE (2004)
2. Bartashevich, P., Mostaghim, S.: Benchmarking collective perception: new task difficulty metrics for collective decision-making. In: Moura Oliveira, P., Novais, P., Reis, L.P. (eds.) EPIA 2019. LNCS (LNAI), vol. 11804, pp. 699–711. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30241-2_58
3. Ebert, J.T., Gauci, M., Mallmann-Trenn, F., Nagpal, R.: Bayes bots: collective Bayesian decision-making in decentralized robot swarms. In: International Conference on Robotics and Automation (ICRA 2020) (2020)
4. Ebert, J.T., Gauci, M., Nagpal, R.: Multi-feature collective decision making in robot swarms. In: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, pp. 1711–1719. International Foundation for Autonomous Agents and Multiagent Systems (2018)
5. Francesca, G., Brambilla, M., Trianni, V., Dorigo, M., Birattari, M.: Analysing an evolved robotic behaviour using a biological model of collegial decision making. In: Ziemke, T., Balkenius, C., Hallam, J. (eds.) SAB 2012. LNCS, vol. 7426, pp. 381–390. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33093-3_38
6. Hoballah, I.Y., Varshney, P.K.: Distributed Bayesian signal detection. IEEE Trans. Inf. Theory **35**(5), 995–1000 (1989)
7. Mondada, F., et al.: The e-puck, a robot designed for education in engineering. In: Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions, vol. 1, pp. 59–65. IPCB: Instituto Politécnico de Castelo Branco (2009)
8. Morlino, G., Trianni, V., Tuci, E., et al.: Collective perception in a swarm of autonomous robots. In: IJCCI (ICEC), pp. 51–59 (2010)

9. Parker, C.A., Zhang, H.: Biologically inspired collective comparisons by robotic swarms. Int. J. Robot. Res. **30**(5), 524–535 (2011)
10. Salomons, N., Kapellmann-Zafra, G., Groß, R.: Human management of a robotic swarm. In: Alboul, L., Damian, D., Aitken, J. (eds.) TAROS 2016. LNCS, vol. 9716, pp. 282–287. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-40379-3_29
11. Strobel, V., Castelló Ferrer, E., Dorigo, M.: Managing Byzantine robots via blockchain technology in a swarm robotics collective decision making scenario. In: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, pp. 541–549. International Foundation for Autonomous Agents and Multiagent Systems (2018)
12. Trabattoni, M., Valentini, G., Dorigo, M.: Hybrid control of swarms for resource selection. In: Dorigo, M., Birattari, M., Blum, C., Christensen, A., Reina, A., Trianni, V. (eds.) ANTS 2018. LNCS, vol. 11172, pp. 57–70. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00533-7_5
13. Valentini, G., Brambilla, D., Hamann, H., Dorigo, M.: Collective perception of environmental features in a robot swarm. In: Dorigo, M., et al. (eds.) ANTS 2016. LNCS, vol. 9882, pp. 65–76. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44427-7_6
14. Valentini, G., Ferrante, E., Hamann, H., Dorigo, M.: Collective decision with 100 kilobots: speed versus accuracy in binary discrimination problems. Auton. Agents Multi-Agent Syst. **30**(3), 553–580 (2016)
15. Valentini, G., Hamann, H., Dorigo, M.: Efficient decision-making in a self-organizing robot swarm: on the speed versus accuracy trade-off. In: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, pp. 1305–1314 (2015)
16. Valentini, G., Hamann, H., Dorigo, M., et al.: Self-organized collective decision making: the weighted voter model. In: AAMAS, pp. 45–52 (2014)
17. Varshney, P.K., Al-Hakeem, S.: Algorithms for sensor fusion, decentralized Bayesian hypothesis testing with feedback, vol. 1. Technical report, Kaman Sciences Corp, Colorado Springs, CO (1991)