Discrete Collective Estimation in Swarm Robotics with Ranked Voting Systems

Qihao Shan
Faculty of Computer Science
Otto von Guericke University
Magdeburg, Germany
Email: qihao.shan@ovgu.de

Alexander Heck
Faculty of Computer Science
Otto von Guericke University
Magdeburg, Germany
Email: alex-heck1991@web.de

Sanaz Mostaghim
Faculty of Computer Science
Otto von Guericke University
Magdeburg, Germany
Email: sanaz.mostaghim@ovgu.de

***Abstract*—The best-of-n problem has been a popular research topic for understanding collective decision-making in recent years. Researchers aim to enable a swarm of agents to collectively converge to a single opinion out of a series of potential options, using only local interactions. In this paper, we investigate the viability of decision-making via majority rule using ranked voting systems in multi-option scenarios where n>2. We focus on two ranked voting systems, single transferable vote (STV) and Borda count (BC). The proposed algorithms are tested in a discrete collective estimation scenario, and compared against two benchmark algorithms, direct comparison (DC) and majority rule using first-past-the-post voting (FPTP). We have analyzed the experimental results, focusing on the trade-off between accuracy and speed in decision-making. We have concluded that ranked voting systems can significantly improve the performances of collective decision-making strategies in multi-option scenarios. Our experiments show that BC is the best performing algorithm in the studied scenario.**

I. Introduction

Collective decision-making is a long-standing area of study within swarm intelligence. The aim of this field is to understand the decision-making mechanisms of naturally existing intelligent swarms, as well as construct decision-making strategies for artificial swarm intelligence systems. Researchers especially seek to imitate the decentralized and localized decision-making process of natural swarm intelligence such as insect swarms or bird flocks, who exhibit complex behaviors via no centralized control mechanism [1].

Within the field of collective decision-making, best-of-n problems encompass the scenarios where the individual agents need to form a consensus out of a discrete list of options collectively [2]. Various collective decision-making strategies have been proposed to enable decision-making in such scenarios. A popular inspiration for collective decision-making strategies is the behavior of natural intelligent swarms, such as bees or ants. An example of such algorithms is the biologically inspired collective comparison proposed by Parker et al. [3]. Such algorithms are characterized by an explicit preference by the agents for one of the options, while the agents try to recruit other agents to their preference during the decision-making process. This approach has been continuously improved when applied to artificial agents. More recent approaches in this class include weighted voter model, represented by the algorithm Direct Modulation of Voter-based Decisions (DMVD) [4], and majority rule, represented by the algorithm Direct Modulation of Majority-based Decisions (DMMD) [5].

In this paper, we pay close attention to decision-making via majority rule, where the agents decide among available options via local majority voting. This process is akin to how voters choose among political candidates in real-world elections. DMMD has been extensively studied in various binary collective decision-making scenarios, including site selection [6] and collective perception [7]. In such binary decision-making scenarios, agents can decide on the optimal option via a simple majority vote. In recent years, there has been a trend to move beyond simple binary collective decision-making scenarios and towards multi-option scenarios. Many approaches have been proposed and tested in multi-option site selection scenarios, such as opinion pooling [8] and cross inhibition [9]. We thus seek to extend the classical majority rule decision-making strategy into a multi-option scenario, where an absolute majority cannot be easily reached. Therefore, we take inspiration from real-world ranked voting systems, which are used as a method to decide among multiple candidates by many democratic countries around the world.

In this study, we focus on two popular real-world ranked voting systems, namely single transferable vote (STV) and Borda count (BC), and investigate their performances when applied to the majority rule decision-making processes of artificial collective intelligence. We have used a simple first-past-the-post (FPTP) voting as a benchmark. We have also compared the performances of majority rule decision-making strategy with a popular benchmark algorithm in collective decision-making, namely direct comparison (DC). The considered algorithms are investigated using a discrete collective estimation scenario. This scenario is an extension of the classical collective perception scenario [10] into a multi-option scenario.

The structure of this paper is as follows. In Section 2, we describe the investigated scenario in detail and cover related works on similar problems and related decision-making strategies proposed in literatures. In Section 3, we present the proposed decision-making strategies based on majority voting and the underlying agent behaviors. Section 4 includes the experiment setup and the results. In Section 5, we discuss our findings and compare them with related works. And finally, Section 6 is the conclusion.

II. Problem Statement & Related Works

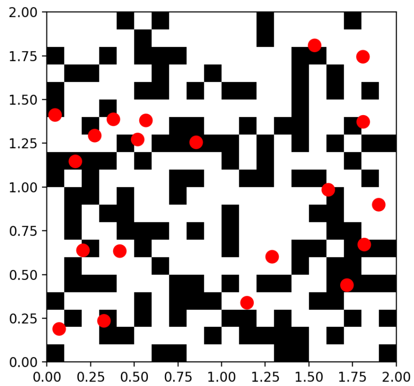


Fig. 1. Example illustration of the investigated discrete collective estimation scenario. The arena is a $2m \times 2m$ square covered by 400 tiles in black & white. Red dots are mobile robots that roam the arena.

We investigate the considered decision-making strategies using a discrete collective estimation scenario as shown in Figure 1. This scenario is inspired by the extensively studied collective perception scenario proposed by Valentini et al. [10], and serves as an extension of it into a multi-option decision-making situation. A continuous collective estimation scenario has also been studied by Strobel et al. [11].

The settings of the decision-making task here are as follows. There is an arena as shown in Figure 1, which is covered in black and white tiles of with a particular ratio. A number of mobile robots are tasked with collectively determining the ratio of area in the arena covered in black, which is referred to as the fill ratio, out of a discrete list of hypotheses. The robots have limited communication, sensory and processing capabilities, and they also only have simple reactive behaviors.

The classical binary collective perception scenario, where the robots determine which color occupies the majority of area, has been well studied in previous literatures. Valentini et al. [10] have investigated the performances of various decision-making strategies, including DMMD, DMVD and DC. Strobel et al. [12] have examined the same strategies under a series of increasingly more difficult fill ratios, which progressively approach 50%. Bartashevich et al. [7] have researched the performances of these algorithms in environments with concentrated distributions of features.

Multi-option collective decision-making has been a hot research topic these few years. Cross inhibition [13] is a nature-inspired algorithm that is shown to perform well in site selection scenarios [9]. Opinion pooling is another approach that received much attention, where the belief masses of options are transmitted and fused directly. Such approaches have been investigated in site selection scenarios [8] and collective perception scenarios [14], [15]. In addition, a similar fusion approach has been attempted in pooling rankings of options and achieving collective learning of the true preference order of the options [16].

The ranked voting algorithms investigated in this paper are an extension to the binary DMMD algorithm proposed in [5]. The base version of DMMD functions as follows. An individual agent alternates between two behaviors, exploration and dissemination. During the exploration phase, the agent performs random walk in the arena and compute the quality of its current chosen option from its observations on the environment. During the dissemination phase, the agent exchanges its choice with its peers, and by the end of the dissemination phase, the agent switches to the choice which has received the most votes among its neighbors. The length of the dissemination phase is proportional to the quality of the current option, which is computed during the exploration phase. This feature enables the swarm to collectively converge to the correct option.

When applying this decision-making process to a multi-option scenario, the probability of any single option achieving absolute majority is low. Thus, we need to introduce a voting system to resolve the deadlock in the decision-making process when no option holds an absolute majority. We therefore take inspiration from real-world elections and apply the following voting systems to the aforementioned process, namely first-past-the-post (FPTP), single transferable vote (STV) and Borda count (BC). FPTP is the most straightforward voting system when more than two options are concerned, where the candidate with the most votes wins regardless of whether absolute majority is achieved. It is a widely used and long-standing voting system in many democratic countries around the world. It has received some criticisms from political scientists [17], and many alternative voting systems have been proposed.

In this paper, we focus on two ranked voting systems, STV and BC. STV [18] is a popular alternative to FPTP. It was proposed in the 19th century by Thomas Wright Hill. In STV, voters would rank the candidates according to preference. The ballots are initially tallied according to the first preferences. The candidate with the least votes will then be eliminated, whose corresponding votes will be redistributed to other candidates according to the voters' rankings. The process continues until a single candidate holds an absolute majority. BC is another ranked voting system, which is less commonly adopted in political processes than the previous two. It has been independently proposed several times, most notably in the 18th century by Jean-Charles de Borda [19]. In BC, each voter's ballot gives points to the candidates according to the voter's ranking. The least preferred candidate receives 1 point, the next least preferred candidate receives 2 points, and the most preferred candidates receives the most points. The winner is the candidate who received the most points in total from all ballots.

In addition, we have also implemented direct comparison (DC) decision-making strategy as a benchmark for the aforementioned algorithms based on majority rule. DC is a popular

1: **Algorithm 1: Decision making behavior of an agent using majority rule strategies**
2: **Input:** Initialized belief: $\underline{\rho}_n$, Initialized decision: d_n
3: **Output:** Converged decisions: d_n
4: Set parameters η, σ, τ
5: Ballot box $R_n = \emptyset$, Individual ballot r_n
6: Initialize d_n with random valid values
7: $state_n = 0$, $timer_n = 0$
8: **while** Decisions in swarm have not converged **do**
9: $\quad timer_n = timer_n - 1$
10: $\quad$ **if** $state_n = 0$ **then**
11: $\quad\quad$ #Exploration State
12: $\quad\quad \underline{ob} = CollectObservation$
13: $\quad\quad \underline{\rho}_n = Normalize(\underline{\rho}_n \circ (K \cdot \underline{ob}))$
14: $\quad\quad$ **if** $timer_n < 0$ **then**
15: $\quad\quad\quad state_n = 1$
16: $\quad\quad\quad timer_n = Sample(exp(\sigma \rho_n[d_n]))$
17: $\quad\quad$ **end if**
18: $\quad$ **else**
19: $\quad\quad$ #Dissemination State
20: $\quad\quad R_n = R_n \cup \underline{r}_m$ if $|R_n| < \eta$, m is the index of a neighboring robot
21: $\quad\quad \underline{r}_n = ComputeBallot(\underline{\rho}_n, d_n)$
22: $\quad\quad$ Compute and broadcast $\underline{r}_n$
23: $\quad\quad$ **if** $timer_n < 0$ **then**
24: $\quad\quad\quad R_n = R_n \cup \underline{r}_n$
25: $\quad\quad\quad d_n = VoteTally(R_n)$
26: $\quad\quad\quad d_n$ = RandomChoice($[(d_n + 1)$, $(d_n - 1)]$, τ)
27: $\quad\quad\quad state_n = 0$
28: $\quad\quad\quad timer_n = Sample(exp(\sigma))$
29: $\quad\quad$ **end if**
30: $\quad$ **end if**
31: **end while**

benchmark algorithm for collective decision-making strategies and has been used frequently in collective perception scenarios [10]. It has also been used in multi-option scenarios [9].

III. Methodology

In this section, we describe the decision-making strategies in detail, together with the underlying behaviors of the robots.

A. Basic Underlying Decision-Making Behaviors

The basic underlying decision-making behaviors of the investigated majority rule strategies are the same and shown in Algorithm 1. We have largely kept the overall decision-making mechanisms of binary DMMD in collective perception scenarios.

In the exploration state (line 11-17), the agent keeps making observations of the color of the arena floor at its current position, and modifies its belief of the fill ratio $\underline{\rho}_n$. We utilize the same mechanism to compute the option qualities for all considered decision-making strategies. The method we used is based on Bayesian statistics and similar to our previous work [14], and is as follows.

We consider $H = 10$ hypotheses on the fill ratio of the arena, expressed in the following matrix K.

$$K = \begin{bmatrix} 0.05 & 0.95 \\ 0.15 & 0.85 \\ \cdots & \\ 0.95 & 0.05 \end{bmatrix} \quad (1)$$

The first column represents the proportion of black tiles P_B, and conversely, the second column represents the proportion of white tiles $1 - P_B$.

When the robot makes an observation on the color of the ground beneath them, the observation is expressed as either $\underline{ob} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ when the ground is black, or $\underline{ob} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ when the ground is white. The belief $\underline{\rho}_n$ is then computed iteratively in line 13 via the element-wise product of the old $\underline{\rho}_n$ value and $K \cdot \underline{ob}$ as follows.

$$\underline{\rho}_n = Normalize(\underline{\rho}_n \circ (K \cdot \underline{ob})) \quad (2)$$

In the dissemination state (line 19-30), the agent exchanges opinions with its neighbors. At the end of the dissemination state, it determines its new chosen decision based on its observations and the information obtained from its neighbors, using a particular majority voting system. The lengths of the dissemination state are modified in line 16 based on the computed quality of the chosen decision $\rho_n[d_n]$, similar to in previous implementations of DMMD.

This decision-making process is tuned by three parameters, maximum number of neighbors η, mean length of exploration/dissemination state σ, and mutation rate τ. It has been observed that in binary decision-making problems, DMMD's performance is heavily affected by the maximum number of neighbors it is allowed to receive opinions from during a dissemination period [6]. In our implementation, η represent the same parameter and is used as a cap on the number of received messages in line 20. Similarly, such decision-making algorithms are also affected by the average lengths of exploration and dissemination periods [20], which is controlled by the parameter σ in our implementation. Different from binary implementations of DMMD, we have added a mechanism to enforce diversity of opinions among the agents and prevent premature convergence to a wrong option. At the end of dissemination periods, an agent will mutate its chosen decision to a neighboring decision on the list of hypotheses with the probability of τ (line 26).

B. Collective Decision Making via Majority Rule

In this subsection, we will introduce the majority voting mechanisms investigated in this paper.

Algorithm 2 shows how agent n computes ballots $\underline{r}_n$ (line 21 in Algorithm 1) for both STV and BC. The agent would rank the available options according to its computed qualities. Its current chosen decision d_n will always be ranked $1st$ regardless of quality.

Algorithms 3,4 and 5 are the vote tallying mechanisms of the voting systems investigated in this paper, and are used in line 25 of Algorithm 1.

Algorithm 2: Computing Ranked Ballots
$ComputeBallotRanked(\underline{\rho}_n, d_n)$
Input: Belief $\underline{\rho}_n$, decision d_n
Output: Ranked ballot $\underline{r}_n$
$\underline{\rho}^*_n = \underline{\rho}_n$
$\underline{\rho}^*_n[d_m] = MaxValue$
$\underline{r}_n = argsort(argsort(-\underline{\rho}^*_n))$

Algorithm 3: Single Transferable Vote
$VoteTallySTV(R_n)$
Input: Collected ballots $R_n = \{\underline{r}_m, m = 1..M\}$
Output: Winning hypothesis index $h*$
$\underline{v} : v_h, h = 1..H$ is vote tallies of all considered hypotheses based on 1st preference
while $max(\underline{v}) \leq sum(\underline{v})/2$ **do**
  Eliminate hypothesis $\hat{h} = argmin(\underline{v})$
  Votes for $\hat{h}$ are transferred to their next best choice
end while
$h* = argmax(\underline{v})$

Algorithm 3 shows the vote tallying mechanism of STV. Here the agent collects M ballots from its peers and itself. Vector $\underline{v}$ tallies the votes for every option according to the 1st preferences on the ballots. The agent then iteratively eliminate the least popular option $\hat{h}$ and redistribute its associated votes to their next best preferences.

Algorithm 4: Borda Count
$VoteTallyBC(R_n)$
Input: Collected ballots $R_n = \{\underline{r}_m, m = 1..M\}$
Output: Winning hypothesis index $h*$
$\underline{v} = \Sigma_{m=1..M}(H - \underline{r}_m)$
$h* = argmax(\underline{v})$

Algorithm 4 shows the vote tallying mechanism of BC. The agent collects the same M ballots as before. The vote tally $\underline{v}$ is calculated by summing up the corresponding points for each option according to the ranking on every ballot. The winning option is the one with the maximum number of points.

Algorithm 5: First Past the Post
$VoteTallyFPTP(R_n)$
Input: Collected ballots $R_n = \{r_m, m = 1..M\}$
Output: Winning hypothesis index $h*$
$v_h, h = 1..H$ is vote tallies of all considered hypotheses
$h* = argmax(v_h)$

Finally, Algorithm 5 shows the vote tallying mechanisms of FPTP. FPTP is not a ranked voting system, and thus the ballots are scalars indicating only the chosen decision d_m. The tallying process is also simple, as the option with the highest number of votes wins regardless of whether absolute majority is achieved. Compared to ranked voting systems described above, FPTP uses less communication bandwidth and computational power. It serves as a naive implementation of majority rule decision-making on multi-option scenarios, and a benchmark algorithm.

C. Benchmark Algorithm: Direct Comparison (DC)

We also use the decision-making strategy of DC as a benchmark for the aforementioned strategies based on majority rule. The decision-making process of DC works similar to the one shown in Algorithm 1. The key difference is that the agents exchange their chosen options and the corresponding computed qualities during the dissemination periods. The agent then switch to the option with the highest corresponding quality among the recorded messages. The lengths of dissemination periods also do not scale with the option qualities.

It has been commented in previous literatures on binary decision-making scenarios that the direct sharing of option qualities increases the communication bandwidth required [10]. In multi-option scenarios, this practice is also observed to propagate inaccurate quality estimates among the agents and distort the decision-making process [9]. However, DC remains a popular benchmark algorithm for collective decision-making strategies, and we use it to gauge the viability of the majority rule strategies proposed above.

IV. Experiments & Results

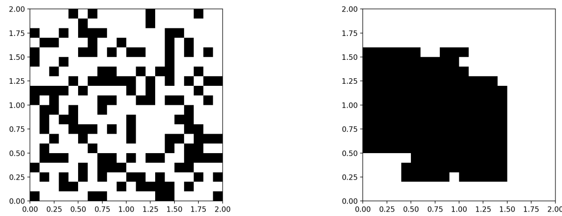
In this section, we will show our experimental setup and results to determine the performances of the considered decision-making strategies.

A. Experimental Setup

In our experiments, we simulate 20 mobile robots with the specification of e-puck [21]. A robot has a linear speed of $0.16m/s$ and a rotational speed of $0.75rad/s$. During an experimental instance, they perform random walk continuously in a $2m \times 2m$ arena. The arena is filled with 400 tiles of a particular fill ratio and pattern. An illustration is shown in Figure 1.

The low-level control mechanism directing the random walk is similar to the one used in [10], and is as follows. The robots alternate between two behaviors, namely going forward in a straight line and rotating in place in a random direction. The durations of the two behaviors are sampled from two random variables, $exp(40)$ and $unif(0, 4.5)$ respectively. To avoid collision, an agent moving forward will stop when the edge of the arena or another robot is detected in front of it and start rotating.

The lengths of the control loop for all decision-making strategies are set to $1s$ for all considered algorithms. As we are using a Bayesian statistics-based technique in computing the option qualities, it has been observed in other literatures that use a similar technique that too small an interval between collecting observations can lead to a reduction in the decision-making accuracy [22]. It has also been deduced in our previous work that an interval of $1s$ is suitable for this environment [14], we thus continue with this setting.



(a) Random Pattern (b) Block Pattern

Fig. 2. Example illustrations of the two patterns of feature distribution investigated

It has been observed that the patterns of feature distribution have a significant impact on the performances of decision-making strategies in collective perception [7]. Most decision-making strategies experience a drop in performance when the environmental features are more concentrated compared to a purely random distribution. This has also been confirmed in our previous work [14]. Therefore, we conduct experiments in both environments as shown in Figure 2.

B. Experimental Results

We first conduct experiments in arenas with randomly distributed black and white tiles, as shown in Figure 2a. Each experimental instance uses a randomly generated arena environment. We have tested the performances of considered algorithms in fill ratios of 0.05, 0.15, 0.25, 0.35 and 0.45, each with 20 independent experimental instances. We pay attention to 3 aspects of the performances, namely accuracy, speed and reliability. We use three metrics to measure them, mean absolute error, mean consensus time and failure rate respectively. These metrics are computed from all experimental instances of a particular parameter configuration.

In related literatures on binary collective perception, performances are usually measured by the probability of making the correct decision and the mean consensus time [7], [10], [12]. However, in the multi-option scenario here, agents have a possibility of reaching a decision-making deadlock and not able to form a consensus at all. Therefore, we set a maximum time limit for our experiments to be $1200s$, which is largely beyond the consensus time of a typical instance, and terminates the experimental instance if the agents fail to reach a consensus by that time. The proportion of experimental instances terminated in this way is referred to as the failure rate.

1) Parameter settings of STV: The performances of STV at various parameter settings are shown in Figure 3 and 4. Figure 3 shows the mean consensus time vs the mean absolute error performances together with their corresponding failure rate and maximum neighbor settings η. The Pareto frontiers of the consensus time vs absolute error trade-off at different η settings are shown in different colors and line styles. It can be observed that the performances of STV is not significantly affected by the limit on maximum number of neighbors, as the Pareto frontiers are very close together.

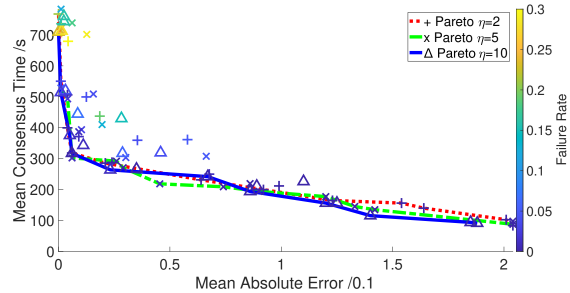


Fig. 3. Performances of STV in environments with random feature distribution with respect to the maximum limit on number of neighbors η. $+$-red dotted:$\eta = 2$, x-green dash-dot:$\eta = 5$, Δ-blue solid:$\eta = 10$. Color codings of markers show failure rate

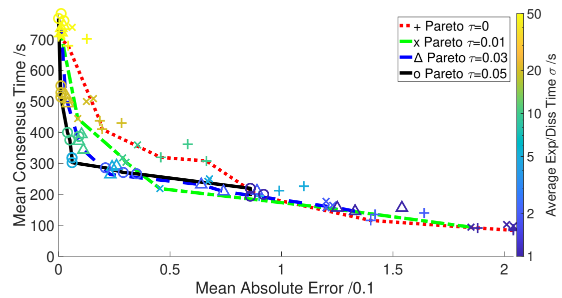


Fig. 4. Performances of STV in environments with random feature distribution with respect to the mutation rate τ and mean exp/diss time σ. $+$-red dotted:$\tau = 0$, x-green dash-dot:$\tau = 0.01$, Δ-blue dashed:$\tau = 0.03$, o-black solid:$\tau = 0.05$. Color codings of markers show the corresponding σ settings

The same data points are shown in Figure 4 regarding the other two parameters, namely the mutation rate τ and the mean time of exploration and dissemination periods σ. Here, both parameters have significant impacts on the performances of STV. As shown via the Pareto frontiers at different τ settings, as τ increases, there is a reduction in error and a slight increase in consensus time, resulting in a high τ setting of 0.05 being able to achieve a good performance of 0.0606 error and $302s$ consensus time at the bottom left. On the other hand, as shown by the color coding of markers, an increasing σ raises the mean consensus time significantly and also reduces the error produced. The performances also become insensitive to other parameters if σ is large, as shown in the clustering of data points when σ is $20s$ or $50s$.

2) Parameter settings of BC: The same plots are made for BC in Figure 5 and 6. When comparing the performances of BC here with those of STV in Figure 3 and 4, it can be noticed that BC can achieve much lower errors and failure rates, while being able to come to a faster consensus. As shown in Figure 5, the performance of BC is slightly affected by the parameter η. Specifically, at the left-hand side of the Pareto frontiers, a higher η raises the consensus time needed to reach an error of 0. In Figure 6, it can be seen that σ has similar effects compared to STV. However, increasing τ does not straightforwardly increase accuracy as before. Notably,

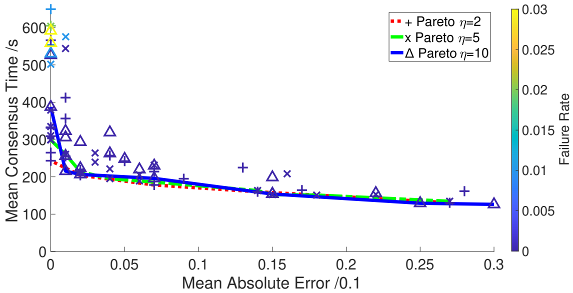


Fig. 5. Performances of BC in environments with random feature distribution regarding the maximum limit on number of neighbors η. $+$-red dotted:$\eta = 2$, x-green dash-dot:$\eta = 5$, Δ-blue solid:$\eta = 10$. Color codings of markers show failure rate

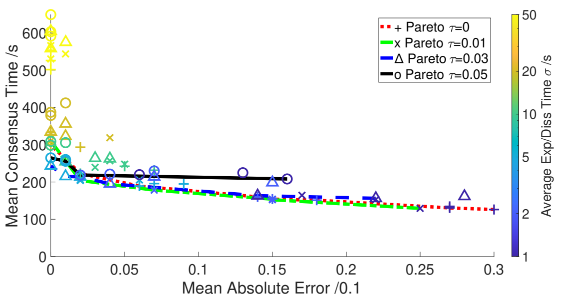


Fig. 6. Performances of BC in environments with random feature distribution regarding the mutation rate τ and mean exp/diss time σ. $+$-red dotted:$\tau = 0$, x-green dash-dot:$\tau = 0.01$, Δ-blue dashed:$\tau = 0.03$, $\circ$-black solid:$\tau = 0.05$. Color codings of markers show the corresponding σ settings

the Pareto frontier produced when $\tau = 0.05$ is completely dominated by that produced when $\tau = 0.03$. On the other hand, increasing τ has the similar effects of reducing the lower bound of error obtained and increasing the lower bound of consensus time as in STV.

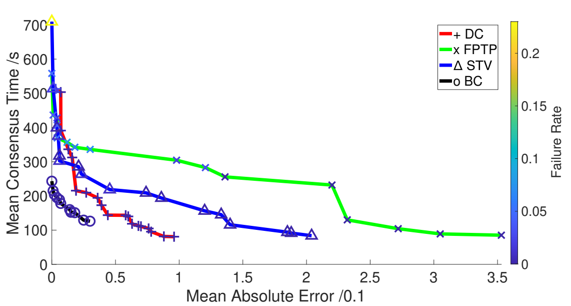


Fig. 7. Pareto frontiers of consensus time vs absolute error for all considered algorithms. $+$-red:DC, x-green:FPTP, Δ-blue:STV, $\circ$-black:BC. Color codings of markers show the failure rates

3) Comparison of considered algorithms in random environments: The performances of the 4 decision-making strategies considered in this paper are compared against each other in Figure 7. All considered strategies exhibit trade-offs between consensus time and absolute error. As a naive implementation of majority rule, FPTP outperforms DC at high consensus times of beyond $400s$. It, however, produces slightly higher failure rates, as shown by the color of the data points. In addition, when a decision need to be reached quickly, the errors produced by FPTP increase rapidly at the bottom right portion of the Pareto frontier.

Compared to the two benchmark algorithms, STV can achieve superior performance in the middle of its Pareto frontier, around the consensus time of $300s$. However, at very high consensus time, its performance is dominated by that of FPTP, and at very low consensus time, its performance is dominated by DC. In addition, it can achieve lower failure rates than FPTP when consensus time is high. On the other hand, BC can dominate most results produced by the other algorithms, except at very low consensus time.

Additionally, we need to consider that the amount of required communication in these decision-making strategies is different. In DC, agents exchange both the chosen option and the corresponding quality estimate. In FPTP, agents exchange only the chosen option. While in the STV and BC, agents exchange the rankings of all options. Therefore, the communication bandwidth required of the 4 considered algorithms would have the relationship of $FPTP < DC < STV = BC$. With this in mind, it can be seen that STV only provides a situational improvement in performances over both benchmarks. In contrast, BC can display superior performance and reach an error of 0 at far lower consensus time than the others.

4) Concentrated environmental feature - block pattern: In the following, we examine the performances of considered algorithms in environments with concentrated feature distributions, specifically arenas where black tiles are arranged in blocks, as shown in Figure 2b. The experimental environments are generated as follows. We fix the fill ratio of these generated environments all at 0.45. Square blocks of black tiles with a random width is sampled from $\mathcal{N}(12, 1^2)$ and first placed in random positions in the arena. The placements happen until the fill ratio is close to the target ratio of 0.45. If the fill ratio is not at 0.45, random individual black tiles will be added or removed from the edge of a square until the targeted fill ratio is reached.

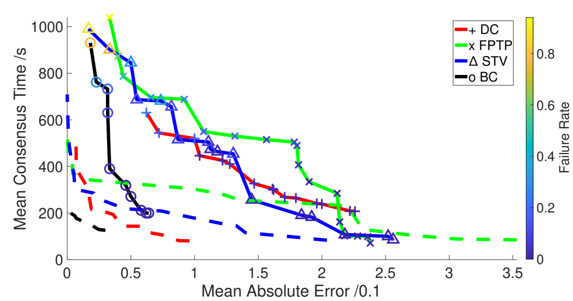


Fig. 8. Pareto frontiers of consensus time vs absolute error for all considered algorithms in environments with concentrated feature distribution. $+$-red:DC, x-green:FPTP, Δ-blue:STV, $\circ$-black:BC. Color codings of markers show the failure rates. Dashed lines show the Pareto frontiers achieved previously in random environments

The Pareto frontiers of all considered algorithms' performances in environments with concentrated feature distribution are shown in Figure 8. The Pareto frontiers obtained earlier in random environments are shown in dashed lines. There is a significant drop in performances for all considered algorithms compared to in random environments. Both FPTP and STV experience an increase in primarily the error. However, fast convergences can still be achieved, as shown on the bottom right end of the Pareto frontiers. On the other hand, DC and BC experience an increase in both decision time and error.

Among the considered algorithms, BC's performances are the least elastic in terms of error, and extending the decision time has very little effects in reducing the error, while for all other algorithms, there are apparent and linear trade-offs between the two metrics. On the other hand, BC still has superior performance compared to the other algorithms. However, at higher decision times, the performances of BC come very close to those of FPTP and STV.

V. Discussion

In the experimental results above, it is demonstrated that BC is a promising technique in multi-option collective decision-making problems. It can significantly outperform our benchmarks in the scenarios investigated in this paper. There is a parallel between the decision mechanism of BC used here and opinion fusion techniques such as in [23] and [15], where the option qualities are combined to update the agents' beliefs. The decision mechanism of BC achieves a limited form of opinion fusion with a predetermined set of beliefs, which are the point allocation used in the tallying process. Compared with full opinion fusion, this design choice has two advantages. First, transmitting the ranking of options takes up less communication bandwidth than transmitting the associated qualities, thus can be achieved with cheaper equipments. Second, limiting the propagation of option qualities can minimize the impact of extreme or faulty estimations to the whole swarm, as indicated in [9].

On the other hand, STV fails to significantly outperform the benchmarks while using more communication bandwidth. It is caused by the stochasticity in the decision-making process of STV. In real-life elections, STV rarely deals with situations with more candidates than voters, which is frequently the case in the scenario in this paper. When multiple options receive no first preferences during voting, the elimination process will eliminate a random option among them. This can cause valid options to be prematurely eliminated. In a typical swarm intelligence setting, the decision-making strategy needs to form a decision based on the information in a small locality and thus STV struggles in such environment. It is however capable of faster convergence than BC, as it is better at eliminating unfavorable options quickly.

FPTP is frequently the worst performing algorithm among the 4 considered. In FPTP, the chosen options are only selected from the first choices of the voters, causing inadequate information transfer among the agents. However, it uses the least communication bandwidth, and therefore should only be considered a viable algorithm when the communication needs to be minimized. Otherwise, a ranked voting system should be utilized in a similar collective decision-making scenario.

In our experiments, the maximum number of neighbors η only has a minimal impact on the performances of the considered algorithm. This runs contrary to the findings in [6]. The reason is that our experimental scenario has a relatively small swarm size. Thus, agents frequently only communicate with a few agents during a dissemination period. It is therefore worth investigating similar scenarios with bigger swarms and observe how that impact the performances.

VI. Conclusion

In this paper, we investigated the viability of majority rule collective decision-making strategies in multi-option best-of-n scenarios. We have examined two ranked voting algorithms, single transferable vote (STV) and Borda count (BC), as well as first-past-the-post (FPTP). We have also utilized direct comparison (DC) as a benchmark algorithm. The considered algorithms are tested in a discrete collective estimation problem with both random environments as well as environments with concentrated feature distributions. The performances of considered algorithms are analyzed using a bi-objective framework to fully describe the speed vs accuracy trade-off.

We have concluded that adopting a ranked voting system, although increasing the communication bandwidth, can significantly improve the performances of majority rule decision-making strategies. Among our examined voting systems, BC has the best performance in our experimental scenario, while STV fails to significantly outperform the benchmarks.

In future works, we aim to examine these ranked voting systems' performances in other multi-option scenarios, such as site selection or multi-color collective perception. We would also study the viability of alternative point allocation methods for BC, such as the Dowdall system.

References

[1] S. Camazine, J.-L. Deneubourg, N. R. Franks, J. Sneyd, E. Bonabeau, and G. Theraula, *Self-organization in biological systems*. Princeton university press, 2003.

[2] G. Valentini, E. Ferrante, and M. Dorigo, "The best-of-n problem in robot swarms: Formalization, state of the art, and novel perspectives," *Frontiers in Robotics and AI*, vol. 4, p. 9, 2017.

[3] C. A. Parker and H. Zhang, "Biologically inspired collective comparisons by robotic swarms," *The International Journal of Robotics Research*, vol. 30, no. 5, pp. 524–535, 2011.

[4] G. Valentini, H. Hamann, and M. Dorigo, "Self-organized collective decision making: the weighted voter model." in *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems*, 2014, pp. 45–52.

[5] ——, "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*, 2015, pp. 1305–1314.

[6] G. Valentini, E. Ferrante, H. Hamann, and M. Dorigo, "Collective decision with 100 kilobots: Speed versus accuracy in binary discrimination problems," *Autonomous Agents and Multi-Agent Systems*, vol. 30, no. 3, pp. 553–580, 2016.

[7] P. Bartashevich and S. Mostaghim, "Benchmarking collective perception: new task difficulty metrics for collective decision-making," in *EPIA Conference on Artificial Intelligence*. Springer, 2019, pp. 699–711.

[8] C. Lee, J. Lawry, and A. Winfield, "Negative updating combined with opinion pooling in the best-of-n problem in swarm robotics," in *International Conference on Swarm Intelligence*. Springer, 2018, pp. 97–108.

[9] M. S. Talamali, J. A. Marshall, T. Bose, and A. Reina, "Improving collective decision accuracy via time-varying cross-inhibition," in *2019 International conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 9652–9659.

[10] G. Valentini, D. Brambilla, H. Hamann, and M. Dorigo, "Collective perception of environmental features in a robot swarm," in *International Conference on Swarm Intelligence*. Springer, 2016, pp. 65–76.

[11] V. Strobel and M. Dorigo, "Blockchain technology for robot swarms: A shared knowledge and reputation management system for collective estimation," in *Swarm Intelligence: 11th International Conference, ANTS 2018, Rome, Italy, October 29–31, 2018, Proceedings*, vol. 11172. Springer, 2018, p. 425.

[12] V. Strobel, E. Castello Ferrer, and M. Dorigo, "Managing byzantine robots via blockchain technology in a swarm robotics collective decision making scenario," in *Proceedings of 17th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2018, pp. 541–549.

[13] A. Reina, G. Valentini, C. Fernández-Oto, M. Dorigo, and V. Trianni, "A design pattern for decentralised decision making," *PloS one*, vol. 10, no. 10, p. e0140950, 2015.

[14] Q. Shan and S. Mostaghim, "Collective decision making in swarm robotics with distributed bayesian hypothesis testing," in *International Conference on Swarm Intelligence*. Springer, 2020, pp. 55–67.

[15] P. Bartashevich and S. Mostaghim, "Multi-featured collective perception with evidence theory: tackling spatial correlations," *Swarm Intelligence*, pp. 1–28, 2021.

[16] M. Crosscombe and J. Lawry, "Collective preference learning in the best-of-n problem," *Swarm Intelligence*, pp. 1–26, 2021.

[17] A. Blais, *To keep or to change first past the post?: the politics of electoral reform*. OUP Oxford, 2008.

[18] N. Tideman, "The single transferable vote," *Journal of Economic Perspectives*, vol. 9, no. 1, pp. 27–38, 1995.

[19] J. d. Borda, "Mémoire sur les élections au scrutin," *Histoire de l'Academie Royale des Sciences pour 1781 (Paris, 1784)*, 1784.

[20] D. Brambilla, "Environment classification: an empirical study of the response of a robot swarm to three different decision-making rules," Master's thesis, Polytechnic University of Milan, Piazza Leonardo da Vinci, 32, 20133 Milano MI, Italy, 9 2015.

[21] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano, and A. Martinoli, "The e-puck, a robot designed for education in engineering," in *Proceedings of the 9th conference on autonomous robot systems and competitions*, vol. 1, no. CONF. IPCB: Instituto Politécnico de Castelo Branco, 2009, pp. 59–65.

[22] J. T. Ebert, M. Gauci, F. Mallmann-Trenn, and R. Nagpal, "Bayes bots: Collective Bayesian decision-making in decentralized robot swarms," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7186–7192, 2020.

[23] M. Crosscombe, J. Lawry, and P. Bartashevich, "Evidence propagation and consensus formation in noisy environments," in *International Conference on Scalable Uncertainty Management*. Springer, 2019, pp. 310–323.