Negative Updating Combined with Opinion Pooling in the Best-of-n Problem in Swarm Robotics

Chanelle Lee[1,2,3], Jonathan Lawry[2], and Alan Winfield[1,3]

[1] Bristol Robotics Laboratory, Bristol, UK c.l.lee@brl.ac.uk,
[2] Department of Engineering Mathematics, University of Bristol, Bristol, UK
[3] University of the West of England, Bristol, UK

Abstract. There is a need for effective collective decision making in decentralised multi-agent and robotic systems. This paper introduces a novel approach to the *best-of-n* decision problem with large n. It utilises negative feedback obtained from direct pairwise comparison of options and evidence preserving opinion pooling. We present agent-based simulation experiments that explore the effects of pool size and the number of options on the speed of consensus. Robotic simulation experiments are then used to investigate the potential of the approach as a method for solving the *best-of-n* decision problem in swarm robotic applications. Overall, the results suggest that the proposed approach is highly scalable with regards to n.

1 Introduction and Background

There is a widely acknowledged and growing need for effective collective decision-making in decentralised multi-agent and robotic systems [1, 13]. Of particular interest is the class of *best-of-n* decision problems [9], where a system needs to achieve consensus on the most desirable option drawn from a number of n distinct possibilities. For example, the choice could be between different nesting sites, foraging locations [15] or which action to perform next [12]. Each option, i, has an associated option quality, ρ_i, which is used by the members of the system to guide the collective decision in favour of the best option. There are three key challenges to this problem. Firstly, the system must reach consensus on a single option based on only local communications. Secondly it needs to ensure that convergence is to the best possible option. Finally, the third challenge is achieving the first two within an application appropriate time frame.

Study of collective decision-making in artificial systems is often heavily influenced by solutions found in nature, such as those of social insects like bees or ants [9, 11]. Scheidler et al. point out in [10] that these solutions tend to be based on positive feedback, i.e. good options are reinforced more than bad ones. For example, in [9] the rate at which an agent recruits others to an option is proportional to the option quality. The greater the quality of an option, the more frequently the agent will advocate for it, thus making it more likely that other agents in the population will be recruited to that option. For many applications, positive feedback with raw values is very successful; however, its effectiveness may be limited in cases where there is little difference in the range of option qualities and as such the best option has insufficient advantage over the others. Valentini et al. notes that there is a lack of research extending into the $n > 2$ cases, leading to the suspicion that this becomes a potentially damaging limitation as n increases and the option quality space becomes more saturated. Furthermore, in the few such examples in the literature, $[10, 4, 2, 8]$, no case larger than $n = 7$ is discussed.

With this in mind, this paper presents a novel approach to the *best-of-n* problem based on negative information obtained from pairwise comparisons. Rather than updating agent opinions using the raw values of the option qualities to inform a positive feedback mechanism, agents instead compare pairs of options and update their opinions based on which is the worst. By means of this direct comparison, agents determine which option is not the best overall and thus acquire negative feedback with which to update their opinions. We show that by combining such negative updating with opinion pooling the system will converge significantly faster than exhaustive comparative search, wherein each agent samples all option qualities and compares them all. This is achieved by using the opinion pooling operators discussed in [6].

This paper is organized as follows: The next section outlines a Bayesian evidential updating method based on negative feedback. We discuss a particular opinion pooling operator in Section 3 and explore the effect of combining evidential updating with opinion pooling on system level consensus and convergence. In the fourth section, we present agent-based simulation results on the speed and reliability of consensus and convergence for the cases of $n = 10, 20, 50$ and 100 with varying pooling sizes. In section five, we present robot simulation experiments with a fixed population size and spontaneous pooling and explore the results as the number of options n is increased. Finally, in section six we give some conclusions and further work.

2 Evidential Updating with Comparisons

We now introduce a mechanism for evidential updating focused on utilising negative feedback from direct option comparisons. The model uses an opinion-based approach as introduced in [14]; extended to the general case of $n > 2$, where agent opinions will be represented as probability vectors across the set of exclusive and exhaustive hypotheses $H = \{\mathcal{H}_i : i = 1, \ldots, n\}$ where $\mathcal{H}_i$ denotes the claim *option i is the best.* As such, an agent A_r, represents their opinion as a probability vector $\mathbf{x}_r$ where $P_{A_r}(\mathcal{H}_i) = x_{ri}$ for $i = 1, \ldots, n$ with $\sum_{i=1}^{n} x_{ri} = 1$, i.e. x_{ri} is the probability with which agent A_r believes $\mathcal{H}_i$ to be true.

An agent samples two options, i and j, and receives qualities ρ_i and ρ_j. Further suppose, without loss of generality, that $\rho_j > \rho_i$. In this case, the agent does not have enough information to know whether j is the best option, but does learn that i cannot be the best possible option, i.e. it receives the evidence

$E_i = \{\mathcal{H}_i\}^c$ the complement of $\{\mathcal{H}_i\}$ with respect to H. The agent can now update their prior belief $\mathbf{x}$ to obtain the posterior $\mathbf{x}|E_i$ using Bayes' theorem as follows:

Definition 1 (Evidential Updating). *Assume we have a set of exclusive and exhaustive hypotheses* $\{\mathcal{H}_i : i = 1, \ldots, n\}$. *Then for* $\mathbf{x} \in [0,1]^n$, $E_i = \{\mathcal{H}_i\}^c$ *and* $\alpha \in [0, \frac{1}{2}]$, *we have,*

$$\mathbf{x}|E_i = \begin{cases} \dfrac{(1-\alpha)x_j}{\alpha x_i + (1-\alpha)(1-x_i)}, & j \neq i, \\ \dfrac{\alpha x_j}{\alpha x_i + (1-\alpha)(1-x_i)}, & j = i. \end{cases}$$

Here α quantifies the agent's belief in the reliability of the evidence source. For $\alpha = 0$, the evidence source is completely reliable evidence source and only provide E_i if i was the worse of the two options. Alternatively, for $\alpha = 0.5$ the evidence source is completely unreliable and so is as likely to provide E_i if i is the best or the worst option.

Unfortunately, even in the best conditions a system using evidential updating on comparisons alone will need all agents to make at least $n - 1$ pairwise comparisons before reaching consensus. Now if we are considering perfect conditions with no noise in the sensed quality values, this is significantly worse than exhaustive comparative search with agents visiting two sites at a time. In the next section we introduce an approach to opinion pooling which allows evidence to be efficiently propagated across the swarm and significantly enhances the effectiveness of negative updating.

3 Combining with Opinion Pooling

In this section, we describe the benefits of combining evidential updating and opinion pooling as we suggested in [6]. We speculate that the use of opinion pooling to propagate evidence between agents in the system will significantly reduce the number of comparisons agents need to make before reaching consensus.

For this study, we limit ourselves to evidence preserving propagation and so use the Product Operator [5, 3]. Below we present an extended version for the case of multiple hypotheses.

Definition 2 (The Multi-Option Product Operator (MProdOP)). *Assume we have a set of exclusive and exhaustive hypotheses* $\{\mathcal{H}_i : i = 1, \ldots, n\}$. *The Product Operator for* k *agents is the function* $c : [0,1]^k \to [0,1]$, *such that for agents* $A_1, \ldots, A_k$ *with opinions* $P_{A_r}(\mathcal{H}_i) = x_{ri}$ *for* $r = 1, \ldots, k$,

$$c(\mathbf{x}_1, \ldots, \mathbf{x}_k) = \frac{\prod_{r=1}^{k} \mathbf{x}_i}{\left\| \prod_{r=1}^{k} \mathbf{x}_i \right\|_1},$$

where $\mathbf{x}_r = [x_{r1}, \ldots, x_{rn}]$ *for all* $r = 1, \ldots, k$, $\prod$ *is the Hadamard product and* $\|-\|_1$ *is the* L_1 *norm.*

Given a pool of k agents with prior beliefs $\mathbf{x}_r$ for $r = 1, \ldots, k$, we suppose that each samples two distinct options and consequently receives evidence E_{i_r}. They then each update their opinion to $\mathbf{x}_r|E_{i_r}$ and aggregate to form the pooled opinion $c(\mathbf{x}_1|E_{i_1}, \ldots, \mathbf{x}_k|E_{i_k})$. Since MProdOP is evidence preserving this is equivalent to $c(\mathbf{x}_1, \ldots, \mathbf{x}_k)|E_{i_1} \ldots |E_{i_k}$.

If we consider the case where all the agents are initialised with uniform opinions where $\mathbf{x}_r = [\frac{1}{n}, \ldots, \frac{1}{n}]$ for $r = 1, \ldots, k$. Furthermore, suppose that the evidence E_i is received by m_i of the agents. Now, without loss of generality, we can also assume that $\rho_1 > \ldots, > \rho_n$, so we have $m_1 = 0$ and $\sum_{i=2}^{n} m_i = k$. This leads to:

$$c(\mathbf{x}_1|E_{i_1}, \ldots, \mathbf{x}_k|E_{i_k})_i = \frac{\alpha^{m_i}(1-\alpha)^{k-m_i}}{\sum_{i=1}^{n} \alpha^{m_i}(1-\alpha)^{k-m_i}}. \tag{1}$$

We can also calculate the probability that an agent receives the evidence E_i. To receive evidence E_i, an agent must sample two options, one of which is i and the other is some $j > i$. There are $2(i-1)$ of these pairings out of a total $n(n-1)$ distinct pairs. Hence, provided distinct pairs of options are selected at random, then $P(E_i) = \frac{2(i-1)}{n(n-1)}$. Thus, the probability that there are m_i occurrences of E_i amongst the k agents for $i = 1, \ldots, n$ is,

$$\frac{k!}{\prod_{i=1}^{n} m_i!} \prod_{i=1}^{n} (\frac{2(i-1)}{n(n-1)})^{m_i}. \tag{2}$$

Combining both of these it follows that,

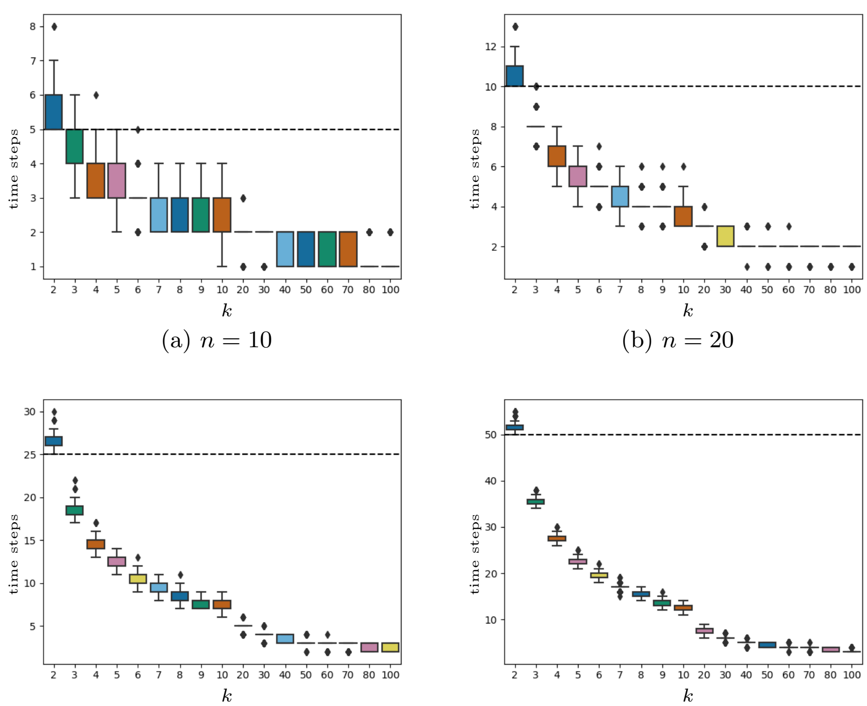
$$\begin{aligned} &\mathbb{E}(c(\mathbf{x}_1|E_{i_1}, \ldots, \mathbf{x}_k|E_{i_k})_i) \\ &= \sum_{\mathbf{m}: m_1=0, \sum_i m_i = k} \frac{k!}{\prod_{i=1}^{n} m_i!} \prod_{i=1}^{n} (\frac{2(i-1)}{n(n-1)})^{m_i} \frac{\alpha^{m_i}(1-\alpha)^{k-m_i}}{\sum_{i=1}^{n} \alpha^{m_i}(1-\alpha)^{k-m_i}}, \end{aligned} \tag{3}$$

giving the expected value of the pooled opinion in option i after a single pooling of the whole population.

4 Agent-based Simulations Experiments

For initial simulation experiments, we present a simple event based multi-agent model exploring the consensus attainment properties of the decision making algorithm proposed in Sections 2 & 3. Specifically, we are interested in the performance of our proposed algorithm versus an exhaustive comparison of all options. We hypothesise that there will be an optimal pool size k^* for each n value, below which our algorithm will take longer on average, and above which it will be faster. The simulation has no physical representation of the environment and as such the options have no associated cost. However, the spatial distribution of the population is represented and at every time step the agents are shuffled to emulate random movement, this approach being consistent with the well-stirred assumption as described in [9].

We assume that a population of N agents begin with no prior knowledge of the option qualities and all opinions are initialised uniformly with probabilities $P(\mathcal{H}_1), \ldots, P(\mathcal{H}_n) = 1/n$ to ensure no initial bias. At every iteration, each agent makes a weighted random choice of two options, i and j say, based on their current probability distribution. The agent compares the qualities of this pair, then uses the updating method as described in Definition 1 to update on the evidence $E = \{\mathcal{H}_i\}^c$ where $\rho_i < \rho_j$. We set $\alpha = 0$ to indicate that agents have total trust in the evidence, a not unreasonable assumption as there is currently no noise. We assign the qualities $\rho_i = \frac{(n-1)-(i-1)}{n-1} \in [0, 1]$ to the options $i \in \{1, \ldots, n\}$, assuming that option 1 is the best with maximal quality, i.e. $\rho_1 > \rho_i \; \forall i \in \{1, \ldots, n\}$.



(a) $n = 10$ (b) $n = 20$

(c) $n = 50$ (d) $n = 100$

Fig. 1: Box and whisker plots showing the time to consensus plotted against population size with (a) $n = 10$, (b) $n = 20$, (c) $n = 50$ and (d) $n = 100$. The dashed lines at $y = \frac{n}{2}$ show the number of time steps $\hat{t}$ needed for the agents to sample every option.

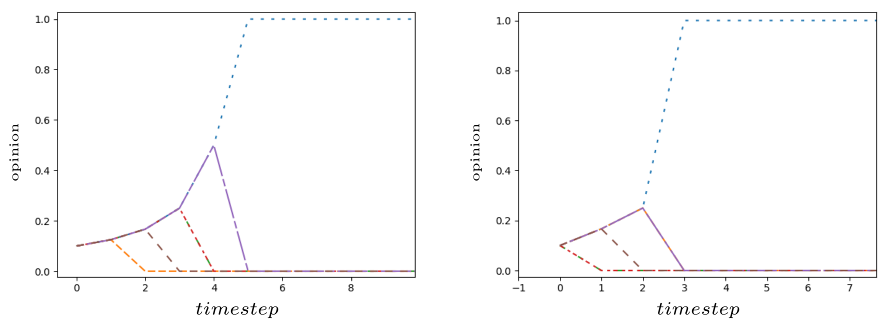
In addition, all agents in the population pool their opinions using the MProdOp operator from Definition 2 with every agent adopting the resulting pooled opinion. Thus pooling size k is fixed and equal to the population size N. For an embodied system, this set-up could be envisaged as a population of robots visiting potential new nest sites, receiving some sensory data indicating that site's quality and updating on this comparison before returning to the original nest to pool opinions. It is thus reminiscent of many similar experiments in the *best-of-n* literature [13]. For each set of parameter values, 1,000 independent runs are carried out with each lasting for 100 iterations. We judge that consensus has been attained in a run once all agents in the population have $P(\mathcal{H}_1) = 1$, i.e. $x_{r1} = 1 \forall r$. Each agent is assumed to be able to sample a pair of options every time step and the number of time steps needed to reach consensus is recorded for each run. This ensures that the two conditions of effective collective decision making are met; the population has converged to a consensus on a single option and that option is the best one possible. If the run fails for either of these conditions, its consensus time is recorded as 100 iterations for ease of comparison. Results are averaged over all 1,000 runs, giving a consensus time for each set of parameter values.

Figures 1(a)-(d) show the number of time steps needed for the system to reach consensus for varying populations size and when the number of site is $n = 10$, $n = 20$, $n = 50$ and $n = 100$ respectively. This is compared with the number of time steps that the system would need if each agent were to sample the quality of every option two at a time, i.e. $\hat{t} = \frac{n}{2}$ time steps. As expected the number of time steps decreases as the pooling size k increases, this effect plateaus once the best performance of a single time step is reached. We show that our hypothesis was correct, that for each different n there would be some optimum k^*_n where performance is better than or equal to $\hat{t}_n$; with $k^*_{10} = 5$ and $k^*_{20} = k^*_{50} = k^*_{100} = 3$. We can see that this would be the case, as if we substitute our requirement that $\alpha = 0$ into Equation 1 gives us:

$$c(\mathbf{x}_1|E_{i_1}, \ldots, \mathbf{x}_k|E_{i_k})_i = \begin{cases} 0, & m_i \geq 1, \\ \dfrac{1}{|\{m_i : m_i = 0 \text{ for } i = 2, \ldots, n\}| + 1}, & m_i = 0. \end{cases} \quad (4)$$

Thus only one agent in the pool needs to have received evidence E_i for all agents to completely disregard the option, i.e. $x_{ri} = 0$ and this becomes more likely with larger pooling sizes. This effect can be seen in Figure 2.

The values for k^* are surprisingly small with $k = 3$ performing optimally for almost all n tested. For example, with $n = 100$ for $k = 3$ consensus was achieved on average within 35 time steps, an improvement of 15 time steps when compared to the 50 that would be needed for each agent to visit every site. Furthermore, considering the $n = 100$ results again, when $k = 5$ the system achieves consensus within on average 23 time steps, this is less than half $\hat{t}_{100}$ with only a $1 : 20$ ratio between pooling agents and the number of options. These results suggest that the evidence propagation is very effective and that our method is highly scalable to



(a) $k = 2$ (b) $k = 5$

Fig. 2: Plots showing a single agent's opinion of different options changing over time when (a) $k = 2$ and (b) $k = 3$.

large n. Unexpectedly, k^*_{10} was greater than the optimal pooling sizes for larger n. This could be due to the accumulative effect of evidence preservation within the system. Every time an agent pools its opinion, it receives all the evidence that every other agent in the pool has. For example, if every agents $A_1, \dots, A_{k-1}$ all update on evidence E_i and agent A_k updates on some different evidence E_j then we would have $c(\mathbf{x}_1|E_i, \dots, \mathbf{x}_{k-1}|E_i, \mathbf{x}_k|E_j) = c(\mathbf{x}_1, \dots, \mathbf{x}_k)|E_i|E_j$. A secondary explanation for this is that there is a greater chance for diversity with larger n and so less instances of 'redundant pooling', i.e. when all agents have the same evidence and hence receive no gain from pooling.

5 Robot-based Simulation Experiments

In this section, we present robot-based simulation experiments where we have a fixed population size, spontaneous pooling sizes k and varying n in order to test the feasibility of our approach in a swarm robotics scenario. We use e-puck robots [7] since they are small, mobile and equipped with a range of sensors making them well-suited to small scale swarm experiments. Experiments are conducted in the V-Rep [4] simulation environment which models many of the required physical characteristics of the e-pucks, such as motion, communication and sensory feedback. Figure 3 shows the experimental arena consisting of n sites equally spaced around a 1.5m disc with a central 'nest' site. Each site is coloured a different shade of red or blue indicating site quality. Site i is given quality $\rho_i = \frac{(n-1)-(i-1)}{n-1} \in [0, 1]$. Sites are coloured a proportion of blue equal to $1-\rho_i$ to help visually distinguish between sites. Noise has not been added into the simulation and physical communication limitations have not been considered.

[4] http://www.coppeliarobotics.com/

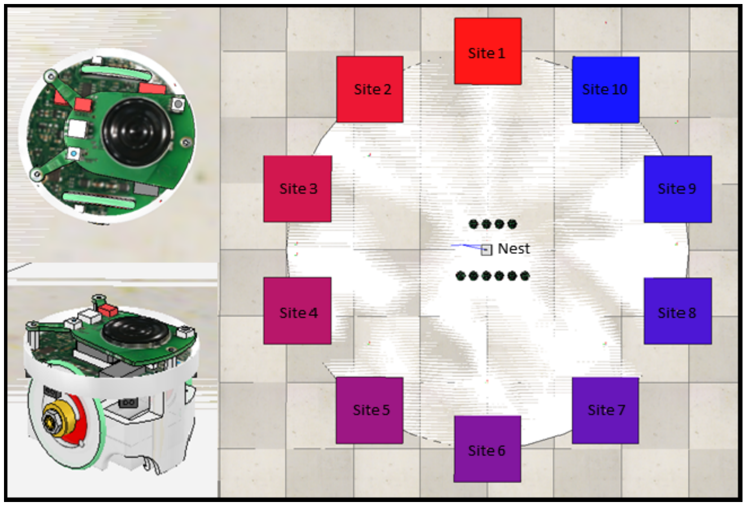
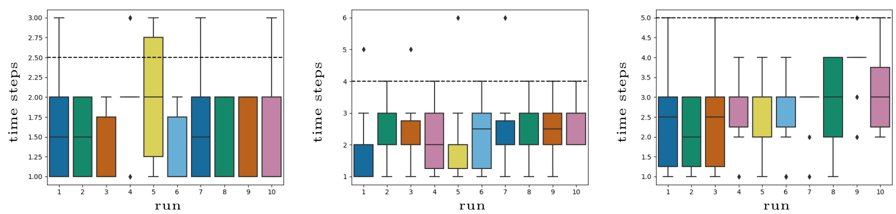


Fig. 3: Top-down and side close-ups of the e-puck model and the arena set-up for $n = 10$.

As in Section 4, the swarm is initialised with uniform probabilities for all sites and each robot makes a weighted random choice of two sites to visit. The robots are given the locations of all sites and use a simple path planning algorithm to travel between sites and the nest. At each site the robots use their colour camera to return a value indicating the amount of red visible, i.e. the site quality. They then compare qualities and update on this negative feedback using Equation 1, before returning to the nest site for aggregation. Once a robot has reached the nest site, it will listen for a message from the transceiver located there to confirm it has reached the nest. This ensures that all robots who have returned for aggregation are within communication range of each other. The robot broadcasts its opinion, while also listening for any neighbouring robots broadcasting their own opinions. As the system has no centralised controller, pooling between the robots is spontaneous and so the pooling size k could range from zero to the whole population, depending on which robots happen to be at the nest site. This differs from the agent-based simulation experiments where k was fixed, and allows us to investigate the effect pooling size variance has on the system. To reduce the communication requirements between robots we employ neighbourhood-based pooling, wherein each robot has their own pool of opinions on which only they update. The robot then uses its updated belief to make a weighted random choice about the next two sites to visit. This process is repeated until $x_{ri} = 1$ for some i, at which point they move to their chosen site and stop. Each parameter set of ten robots and $n \in [5, 8, 10]$ was run ten times, with the pooling size of each aggregation and the number of time steps needed for each agent to reach convergence being recorded.

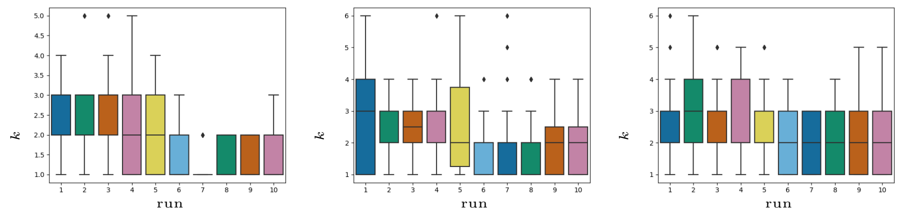


(a) $n = 5$ (b) $n = 8$ (c) $n = 10$

Fig. 4: Box and whisker plots showing the spread of the number of time steps to consensus for individual robots in a population of size $N = 10$ for ten different runs with number of sites (a) $n = 5$, (b) $n = 8$ and (c) $n = 10$. The dashed lines at $\frac{n}{2}$ show the number of time steps $\hat{t}$ necessary for a robot to sample every option.

For all runs the swarm was successful with respect to the first two key challenges of the *best-of-n* decision problem; all robots reached consensus on a single option and that option was the best possible one. Figure 4 gives the time frame for reaching consensus, showing for each run the number of time steps the swarm needed before reaching a final decision. As in Section 4, we use the number of time steps that a robot would need to visit all sites, two at a time, as a benchmark for performance. Many of the runs across all values of n achieved consensus within $\hat{t}$ time steps, with the best performance for $n = 10$, where the swarm reached consensus within $\hat{t}$ time steps for all runs. Moreover, in 60% of the runs, the swarm achieved consensus in less than $\hat{t}$ and, in particular, in Run 2 consensus is achieved in just three time steps. This is consistent with our findings in Section 4 that increasing n can have a positive effect on pooling due to the decreased likelihood of redundant pooling. These results have also been achieved with relatively low pooling sizes with Figure 5 showing that no run achieved an average pool size greater than three. This suggests that alterations in the control architecture leading to a higher average pooling size, such as increasing the time the swarm spends sharing opinions with neighbouring robots during aggregation, could lead to even faster consensus times.

One of the worst results is Run 5 for $n = 8$ where the swarm took six time steps to reach consensus, two more than $\hat{t}$. A closer look at this result shows that this was caused by a single outlying agent, with the rest of the swarm achieving consensus within three time steps. Consideration of the average pooling sizes in this run, as seen in Figure 5(b), reveals that while there were some very large pools with six robots, the average was much lower at only two robots. From this we conjecture that some robots who were part of the larger pools converged very quickly, and thus essentially removing themselves and the evidence they had gained from the system too early for the other robots to benefit. This suggests that while larger k will give faster consensus in general, care has to be taken with the potential variance of k values so as not to isolate robots from the system. A



(a) $n = 5$ (b) $n = 8$ (c) $n = 10$

Fig. 5: Box and whisker plots showing range of pooling sizes k for robots in a population of size $N = 10$ for ten different runs with (a) $n = 5$, (b) $n = 8$ and (c) $n = 10$.

way of alleviating this effect in future work could be to have a period of opinion broadcasting after a robot has reached their decision for the benefit of other robots. This could also have the additional benefit of reducing the range of time steps needed for the swarm to reach consensus, which in the run above was as high as six and as low as one.

6 Conclusion

In this paper, we have introduced a novel approach to solving the *best-of-n* decision problem that uses evidence updating from negative feedback combined with opinion pooling. We present an evidential updating method that utilises negative feedback obtained from direct pairwise comparisons of options. We then introduced the multi-option pooling operator MProdOp, with the expectation that its evidence preserving property would efficiently propagate evidence throughout a swarm. In simulation experiments, we explore the effect of pooling size on the time to consensus and the scalability of our approach to increasing values of n. Finally, we investigated our approach in a typical swarm robotics scenario in simulation to test its applicability.

The simulation experiments presented in Section 4 suggest that our approach is highly scalable with regards to n. Indeed, successful and effective consensus was reached even with $n = 50$ and $n = 100$ options. We also found that although performance improves with larger k, the system can achieve consensus faster than exhaustive comparison even with very small k. For example, with $n = 100$ for $k = 3$ consensus was achieved on average within 35 time steps, a considerable improvement on the 50 time steps that would be needed for each agent to visit every site.

Overall, the robot simulation experiments indicate that our approach has potential as a method for solving the *best-of-n* decision problem in swarm robotics applications. We have presented a simplified scenario where the swarm needed to pick the reddest of n sites with $n = 5, 8$, and 10. The first two key challenges facing the *best-of-n* problem were met in all runs. Additionally, for a majority of runs the swarm was able to achieve consensus faster than exhaustive comparative search. This demonstrates a level of robustness to pooling size, with possible improvements if average pooling sizes could be increased. Furthermore, as not all agents had to visit all sites to achieve consensus, this approach could potentially work in a scenario where each agent is unable to visit all sites in the environment. We also saw improvements with an increase in n suggesting that our approach will scale well with large n in the swarm robotics environment.

An observed limitation of our proposed method is that it currently only works in environments for which both n and the location of sites is known. For many possible applications, such as a search a rescue site checking task, this is of minor significance as all sites would be known; however, it does restrict the adaptability of the approach in uncertain or changing environments. Further work would look to address this by introducing the ability to increase n upon the discovery of new sites.

Parker and Zhang argue in [9] that agents should not be performing such direct comparisons of options as it can leave the system exposed to potential stagnation from evaluation errors and hence in future work we will investigate how our algorithm performs in the presence of noise, e.g. in sensed quality values. We hypothesise that by introducing distrust, both by setting $\alpha > 0$ and using a diluting pooling operator [6], our system could be robust to such noise. Furthermore, we intend to explore the robustness of the system in a dynamic environment where the best option may change, much as in [11]. In addition, we plan to replicate our experiments on a physical robotic platform and investigate what happens with much larger swarms, for example when $N > 500$.

Acknowledgements. This research was partially funded by an EPRSC PhD studentship as part of the Centre for Doctoral Training in Future Autonomous and Robotic Systems (grant number EP/L015293/1). The authors would like to thank Michael Crosscombe for many useful discussions and valuable comments. All underlying data is included in full within the paper.

References

1. Brambilla, M., Ferrante, E., Birattari, M., Dorigo, M.: Swarm robotics: a review from the swarm engineering perspective. Swarm Intelligence **7**(1), 1–41 (2013)
2. Crosscombe, M., Lawry, J.: Exploiting vagueness for multi-agent consensus. In: Multi-agent and Complex Systems. pp. 67–78. Springer (2017)
3. Easwaran, K., Fenton-Glynn, L., Hitchcock, C., Velasco, J.D.: Updating on the credences of others: Disagreement, agreement, and synergy. Philosophers' Imprint **16**(11), 1–39 (2016)
4. Garnier, S., Combe, M., Jost, C., Theraulaz, G.: Do ants need to estimate the geometrical properties of trail bifurcations to find an efficient route? a swarm robotics test bed. PLoS computational biology **9**(3), e1002903 (2013)

5. Genest, C., Zidek, J.V.: Combining probability distributions: A critique and an annotated bibliography. Statistical Science pp. 114–135 (1986)
6. Lee, C., Lawry, J., Winfield, A.: Combining opinion pooling and evidential updating for multi-agent consensus. International Joint Conference on Artificial Intelligence p. [Accepted] (2018)
7. Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A., Magnenat, S., Zufferey, J.C., Floreano, D., Martinoli, A.: 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. Parker, C.A., Zhang, H.: Active versus passive expression of preference in the control of multiple-robot decision-making. In: Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on. pp. 3706–3711. IEEE (2005)
9. Parker, C.A., Zhang, H.: Cooperative decision-making in decentralized multiple-robot systems: The best-of-n problem. IEEE/ASME Transactions on Mechatronics **14**(2), 240–251 (2009)
10. Scheidler, A., Brutschy, A., Ferrante, E., Dorigo, M.: The k-unanimity rule for self-organized decision-making in swarms of robots. IEEE transactions on cybernetics **46**(5), 1175–1188 (2016)
11. Schmickl, T., Thenius, R., Moeslinger, C., Radspieler, G., Kernbach, S., Szymanski, M., Crailsheim, K.: Get in touch: cooperative decision making based on robot-to-robot collisions. Autonomous Agents and Multi-Agent Systems **18**(1), 133–155 (2009)
12. Seth, A.K., Bryson, J.J.: Natural action selection, modeling. In: Pashler, H. (ed.) Encyclopedia of the mind, pp. 557–559. Sage (2013)
13. Valentini, G., Ferrante, E., Dorigo, M.: The best-of-n problem in robot swarms: Formalization, state of the art, and novel perspectives. Frontiers in Robotics and AI **4**, 9 (2017)
14. Wessnitzer, J., Melhuish, C.: Collective decision-making and behaviour transitions in distributed ad hoc wireless networks of mobile robots: Target-hunting. In: European Conference on Artificial Life. pp. 893–902. Springer (2003)
15. Winfield, A.F.: Foraging robots. In: Encyclopedia of complexity and systems science, pp. 3682–3700. Springer (2009)