

Efficient Searching by Bias and Raising Threshold Algorithm Using Multiple Voting in the Best-of-n Problem

N. H. Phung, M. Kubo, and H. Sato

Abstract—The Bias and Raising Threshold (BRT) algorithm is one of the methods for the best-of-n problem (BSTn) that allows a group of robots to find out the appropriate collective option among a set of n alternatives. This paper improves the BRT algorithm by using multiple voting for shortening the search time. Concretely, each robot is considered that might be able to vote multiple times in a selection. The experimental results revealed that the search time was only dramatically reduced but also the search accuracy was improved, especially in difficult problems where there are a large number of options ($n \geq 2$).

Index Terms—BRT algorithm, multiple voting, the best-of-n problem, complex systems, collective intelligence, group decision-making.

I. INTRODUCTION

The best-of-n problem (BSTn) is an important issue for several robotics scenario focuses on the question of how a group of robots achieves the optimal choice from among a set of n options without a leader [1]. While BSTn problem has been studied for almost two decades, nearly all previous works focused on the best-of-2 problems, such as prey-hunting [2], double-bridge scenario [3], [4] or binary aggregation scenario [5]-[7]. These methods might be sufficient for determining the direction of robots at Gridworld but not able for the decision-making in a complex environment, such as drones flying in three-dimensional space. To tackle this problem, the research on how can deal with a large number of options ($n \geq 2$) has been extensively studied in recent years. For example, in works inspired by the nest-site selection of honeybees [8]-[10], robots can make an agreement with correct option among multiple options. However, the number of options that can be dealt with is limited and there are cases where the proportion of the correct choices is only about 70%. Hence, it is difficult to increase the number of options using exiting methods.

To this problem, we have proposed Bias and Raising Threshold (BRT) algorithm that enables robots to deal with a large number of options in a trial and error manner by introducing a threshold term, that its value increases by the time elapsed, into individual' decision-making process [11]. Each robot changes its option by monitoring the number of supporters of its recurrent option. Robots keep voting until all members make an agreement with the appropriate option. Therefore, the BRT algorithm is potentially effective in

maintaining the homeostasis of groups and finding out the appropriate option with very high accuracy. However, the experimental results showed that it sometimes takes a large amount of time to find out the best option. This is inefficient in the cases where the search time is limited or a short time is desirable.

In this paper, we propose an extension of BRT algorithm to solve this problem by using multiple voting instead of a single vote in the original BRT algorithm. The multiple voting (which is also known as plural voting) is the practice whereby one person might be able to vote multiple times in an election. And it is recommended that the multiple voting produces better political consequences than “one person, one vote” democracy in [12] by J. S. Mill.

Inspired by voting systems (electoral systems), various works have been studied in the context of robotics and opinion dynamics. One of the simplest mechanisms is voter model that leads to particularly accurate collective decisions but has long decision times [13]. Majority rule is also a popular model that allows a group of robots to reach consensus faster than the voter model but also less accurate [14], [15]. Recently, the k-unanimity rule has been proposed by Scheidler et al. [4] and its performance is a compromise between that of the voter model and that of the majority rules [16]. In the present paper, we provide a decision-making algorithm that enables a group of robots to find very quickly the best option with very high accuracy.

The rest of the present paper is organized as follows. In Section II, we will present Iwanaga and Namatame model, which inspired this work, and original BRT algorithm. In Section III, we will describe our proposed method and then, in Section IV, we will present our experimental results. Finally, in Section V, we will present the conclusions.

II. RELATED WORKS

A. Iwanaga and Namatame Model [17]

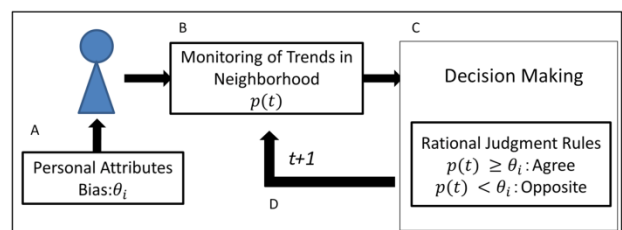


Fig. 1. The role of social skin in individual decision-making [17].

BRT algorithm is inspired by Iwanaga and Namatame model [17] in which a group of agents rapidly becomes bipolar: *Agree* or *Disagree*. Firstly, we explain Iwanaga and

Manuscript received March 15, 2019; revised April 25, 2019.

N. H. Phung, M. Kubo, and H. Sato are with the Computer Science, National Defense Academy of Japan, Hashirimizu 1-10-20, Yokosuka, Kanagawa, Japan (e-mail: ed17006@nda.ac.jp, masaok@nda.ac.jp, hsato@nda.ac.jp).

Namatame model.

They got inspiration from Schelling's critical mass research [18]. As shown in Fig. 1, they stated that individual decision-making does not only depend on personal philosophy and personal preferences, but also on the atmosphere of the whole group ($p(t)$). Each individual shall take the position of agreeing or disagreeing with an opinion and has a personal attribute bias value (Fig. 1A), which is used in this decision-making process (Fig. 1C). Each individual observes the ratio of agreed members to the opinion $p(t)$ (Fig. 1B).

$$\begin{cases} p(t) \geq \theta_i : & \text{Agree} \\ p(t) > \theta_i : & \text{Opposite} \end{cases} \quad (1)$$

As shown in (1), an individual i agrees with an option if the percentage of agents who agree with this option $p(t)$ (the agreement ratio) is higher than the bias value θ_i at time t . After all individuals made their decisions, the agreement ratio $p(t+1)$ at time $(t+1)$ changes (Fig. 1D), so the individual makes its decision again (Fig. 1C).

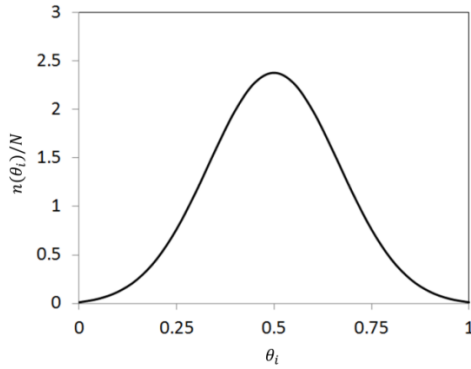


Fig. 2. An example of bias distribution [17].

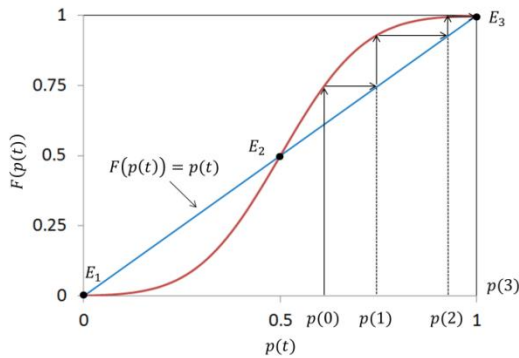


Fig. 3. Cumulative distribution function and Equilibrium point of collective consensus decision (E_1, E_3 are stable points, E_2 is unstable point [17]).

Iwanaga and Namatame proposed a method to analyze the dynamics of the agreement ratio $p(t)$ in this decision-making process. If the distribution of θ is determined as shown in Fig. 2, the cumulative distribution function $F(\theta)$ forms an S-shaped curve as shown in Fig. 3. If the agent makes a decision according to (1), the ratio $p(t)$ at the time t follows the following equation:

$$F(\theta) = \sum_{\theta_i \leq \theta} n(\theta_i) \quad (2)$$

$$p(t+1) = F(p(t)) \quad (3)$$

$$p^* = F(p^*) \quad (4)$$

The solutions of the above equation are called fixed points. There are two types of fixed points: a stable point and 2 unstable points. If the bias value θ_i is distributed in a bell shape as shown in Fig. 2, one unstable point (E_2) and two stable points (E_1, E_3) are generated in the cumulative distribution function, and all members smoothly move their opinions to ultimately agree or disagree (Fig. 3).

B. BRT Algorithm [11]

We have extended Iwanaga and Namatame model to the best-of- n case with $n \geq 2$. To avoid confusion, the number of options is written as M instead of n in following.

We assumed that there is a group of N agents $G = \{P_i: i = 1, \dots, N\}$. They have to find out the best option a_{goal} among a set of M options $A = \{a_j: j = 1, \dots, M\}$ ($a_{goal} \in A$) but they don't know that whether their current options are the best one until they archive agreement state where all members choice the same option. Here, we assumed that there is only one the best option. $A_i(t) \in A$ is the option of agent P_i at time t . $n(a_j)$ represents the number of agents selecting option a_j . The agent P_i has a bias θ_i represent his personal preference ($0 < \theta_i < 1$).

The agent P_i decides his next time option $A_i(t+1)$ at time t as follows. If

$$\frac{n(A_i(t))}{N} \geq \theta_i + \tau \cdot c_i(t) \cdot (t - t_{i,last}) \quad (5)$$

is satisfied, he will keep selecting the current option, i.e. $A_i(t+1) = A_i(t)$. Otherwise, he will stochastically select another option, i.e. $A_i(t+1) = A \setminus A_i(t)$. Here, we define τ is a constant representing the prediction rate of the increase in number of supporters. In addition, $t_{i,last}$ is the time at which the agent P_i last changed its option, and $(t - t_{i,last})$ is the time span that the same option continues to be selected.

$$t_{i,last} = \begin{cases} t & A_i(t) \neq A_i(t-1) \\ t_{i,last} & \text{otherwise} \end{cases} \quad (6)$$

$c_i(t)$ is a function that is equal to 0 or 1 depending on the follows:

$$c_i(t) = c(t) = \begin{cases} 0 & \forall i, A_i(t) \equiv a_{goal} \\ 1 & \text{otherwise} \end{cases} \quad (7)$$

If the ratio of agents who select the same option as theirs is lower than $\theta_i + \tau \cdot c_i(t) \cdot (t - t_{i,last})$, the agent stochastically selects another option. The second term on the right side of (5) is a term with increasing value by the time elapsed.

III. THE PROPOSED METHOD

A. BRT Algorithm Using Multiple Voting

Here, we extend BRT algorithm by using multiple voting instead of single vote. We call this BRT algorithm k -BRT with k is the number of votes that each agent uses to vote at each time step ($M > k > 1$).

We assume that agent P_i has k votes for M options and votes k times at each time step. In original BRT algorithm, since each agent only has a single vote, we adopted only one bias for each agent's decision-making. In the proposed method, since each agent votes to different k options in each time step, different biases should be used for different options. Concretely, the agent P_i has M biases $B_i = \{\theta_{i1}, \dots, \theta_{ij}, \dots, \theta_{iM}\}$ ($0 < \theta_{ij} < 1, j = 1, \dots, M$) for M options. At l^{th} voting, $A_{il}(t) \in A$ is the option voted by agent P_i and the bias related to $A_{il}(t)$: $\vartheta_{il}(t) \in B_i$ at time step t ($l = 1, \dots, k$).

The agent P_i decides his next time l^{th} -option $A_{il}(t+1)$ as follows. If

$$\frac{n(A_{il}(t))}{N} \geq \mathcal{G}_{il} + \tau \cdot c_{il}(t) \cdot (t - t_{il,last}) \quad (8)$$

is satisfied, he will keep selecting the current l^{th} -option, i.e. $A_{il}(t+1) = A_{il}(t)$. Otherwise, he will stochastically select another option, i.e. $A_{il}(t+1) = A \setminus A_{il}(t)$. Similar to the original BRT algorithm, τ is a constant representing the prediction rate of the increase in number of supporters. $t_{il,last}$ is the time at which the agent P_i last changed his l^{th} -option, and $(t - t_{il,last})$ is the time span that the same option continues to be selected.

$$t_{il,last} = \begin{cases} t & A_{il}(t) \neq A_{il}(t-1) \\ t_{il,last}(t-1) & otherwise \end{cases} \quad (9)$$

$c_{il}(t)$ is a function that is equal to 0 or 1 depending on the follows:

$$c_{il}(t) = c(t) = \begin{cases} 0 & \forall i, \exists l, A_{il}(t) \equiv a_{goal} \\ 1 & otherwise \end{cases} \quad (10)$$

In other words, in any of k -voting, if all members vote to a_{goal} then $c_{il}(t)$ becomes 0 and the decision-making process finishes successfully.

B. Distribution of Bias

In this paper, we adopt the bias generating method using a quadratic function proposed in [19]. Concretely, the bias θ_{ij} is sampled from the following distribution:

$$n(\theta_{ij}) = \begin{cases} 0? & \theta_{ij} \leq 0 \\ vN\theta_{ij}^2 & 0 < \theta_{ij} \leq 1/M \\ vN(\theta_{ij} - 2/M)^2 & 1/M < \theta_{ij} \leq 2/M \\ 0? & 2/M \leq \theta_{ij} \end{cases} \quad (11)$$

where N is the number of agents, $n(\theta_{ij})$ represents the number of agents taking θ_{ij} , $\mu = 1/M$ is the expectation of

the distribution, and $v = \frac{3M^3}{2}$ is a normalization term. Thus, we derive following equation:

$$\int_0^1 n(\theta_{ij}) d\theta_{ij} = 1 \quad (12)$$

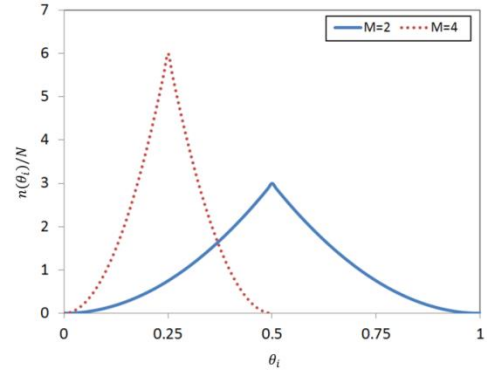


Fig. 4. Examples of distributions of bias [19].

Fig. 4 illustrates some examples of the distributions of bias determined by (11). As can be seen, the shape of the bias distribution is a peak determined by the number of options M only.

C. Dynamics of the Decision-Making Process

Here, by the computer simulations, we show an example of the dynamics of the best option searching process. The parameters were set as follows: $k = 2, M = 5, N = 100, \tau = 0.003, k = 2, a_{goal} \equiv a_5$. At the initial time, assume that all agents randomly select 2 options and vote for them.

Fig. 5 and Fig. 6 show the transition of the number agent voting for each option in 1st and 2nd voting process, respectively. The horizontal axis represents time and the vertical axis represents the number of agents selecting each option. As can be seen, in both voting processes, all agents rapidly reach the agreement state and change the collective option if it is not the best option. At time step round 1050, while the best option has not been found out in voting, in all agents collectively select the best option after 5 agreement times. This result shows that a group of agents can rapidly make the collective decision and find out the best option using the proposed method. By emerging a trial and error process at the macro level, our proposed method is not only efficient in the case of difficult problems where there are a large number of options but also in the case of dynamic environment where the environment and tasks frequently change.

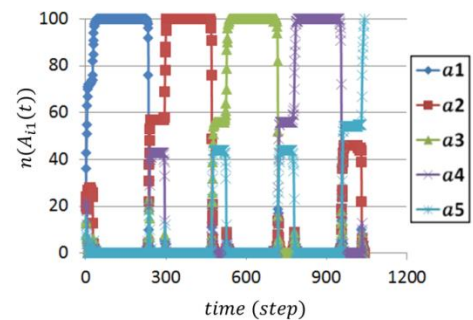


Fig. 5. The transition of number agents selecting each option in 1st voting. $M = 5, N = 100, \tau = 0.003, k = 2$.

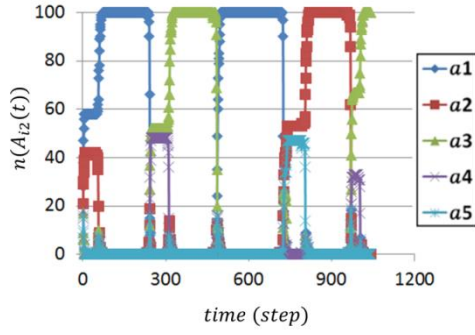


Fig. 6. The transition of number agents selecting each option in 2^{nd} voting. $k = 2, M = 5, N = 100, \tau = 0.003, a_{goal} \equiv a_5$.

D. Analytical Model

In the following, we analytically study the dynamics of the proposed model.

TABLE I: PROBABILITY TO FIND OUT THE BEST OPTION AT AN AGREEMENT STATE

Number of agreements	Original BRT algorithm	k -BRT algorithm
1^{st}	$\frac{1}{M}$	$\frac{M^k - (M-1)^k}{M^k}$
2^{nd}	$\frac{1}{M}(1 - \frac{1}{M})$	$P(1)(1 - P(1))$
3^{rd}	$\frac{1}{M}(1 - \frac{1}{M})^2$	$P(1)(1 - P(1))^2$
x^{th}	$\frac{1}{M}(1 - \frac{1}{M})^{x-1}$	$P(1)(1 - P(1))^{x-1}$

While in the original BRT algorithm, at the first agreement, the probability of finding the best option a_{goal} is $P(1) = 1/M$, in the proposed method, this probability is $P(1) = \frac{M^k - (M-1)^k}{M^k}$. Table I shows the probability to find out the best option at an agreement state. Therefore, the probability at the x^{th} agreement generally is

$$P(x) = \frac{M^k - (M-1)^k}{M^k} \left(1 - \frac{M^k - (M-1)^k}{M^k} \right)^{x-1} \quad (13)$$

Thus,

$$\begin{aligned} \ln(P(x)) &= \ln(P(1)(1 - P(1))^{(x-1)}) \\ &= (x-1)\ln(1 - P(1)) + \ln(P(1)) \\ &= x\ln(1 - P(1)) + \ln(P(1) / (1 - P(1))) \end{aligned} \quad (14)$$

So, we obtain

$$\begin{aligned} P(x) &= \frac{P(1)}{1 - P(1)} \exp(x \ln(1 - P(1))) \\ &= \frac{M^k - (M-1)^k}{(M-1)^k} \exp(x \ln((M-1)^k / M^k)) \end{aligned} \quad (15)$$

Fig. 7 shows the distribution of the number of agreements required to find out the best option. As can be seen, since these distributions are exponential distributions, small number of agreements has a higher probability to find out the best option than the large one. In addition, the bigger k is, the smaller the variance and expected value of the distribution

becomes. In other words, by using multiple voting, it can be expected to dramatically reduce the time required to find out the best option and improve the performance of the decision-making process.

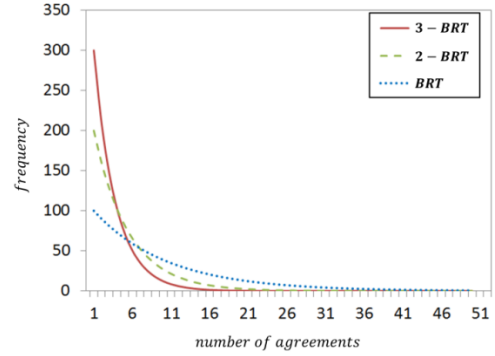


Fig. 7. The distribution of the number of agreements required to find out the best option.

Next, in order to quantitatively evaluate the advantages of the proposed method, we estimate the average times required to find out the best option. As can be seen in Fig. 5 and Fig. 6, the estimating equation must be as follows:

$$\bar{T} = \bar{E} \cdot T_D \quad (16)$$

where \bar{E} is the average times of agreement required and T_D is the time required for one agreement.

Firstly, we obtain the average times of agreement \bar{E} as follows:

$$\begin{aligned} \bar{E} &= \sum_{x=1}^{\infty} xP(x) \\ &= \frac{P(1)}{1 - P(1)} \sum_{x=1}^{\infty} x \cdot \exp(x \cdot \ln(1 - P(1))) \end{aligned} \quad (17)$$

As known that

$$\sum_{x=1}^{\infty} e^{xb} = \frac{e^b}{1 - e^b} = \frac{1}{e^{-b} - 1} \quad (18)$$

Thus,

$$\begin{aligned} \frac{d}{db} \sum_{x=1}^{\infty} e^{xb} &= \frac{d}{db} (e^{-b} - 1)^{-1} \\ \sum_{x=1}^{\infty} b e^{xb} &= \frac{e^{-b}}{(e^{-b} - 1)^2} = \frac{e^b}{(1 - e^b)^2} \end{aligned} \quad (19)$$

So, we obtain

$$\begin{aligned} \bar{E} &= \frac{P(1)}{1 - P(1)} \frac{1 - P(1)}{P(1)^2} = \frac{1}{P(1)} \\ &= \frac{M^k}{M^k - (M-1)^k} \end{aligned} \quad (20)$$

Secondly, in [19], the time required for one agreement T_D has been estimated as follows:

$$T_D \approx \frac{1 - \frac{2}{M}}{\tau} \quad (21)$$

Therefore, the average times required \bar{T} to find out the best option will be

$$\bar{T} \approx \frac{M^k}{M^k - (M-1)^k} \frac{1 - \frac{2}{M}}{\tau} \quad (22)$$

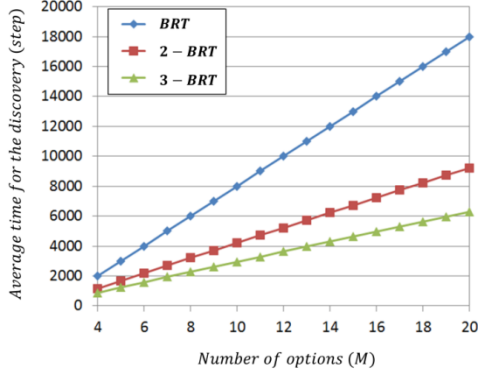


Fig. 8. Estimation of the average time required to find out the best option based on (22) in the case of $\tau = 0.001$.

Fig. 8 shows an estimation of the average time required to find out the best option based on (2) in the case of $\tau = 0.001$. As can be seen, the required average time increases according to a linear function of the number of options M in both of original BRT algorithm and the proposed method. Moreover, the slope of the linear function in the proposed method is smaller than it is in original BRT algorithm, i.e. the average time required to find out the best option will be reduced by using the proposed method dramatically.

IV. EXPERIMENTAL RESULTS

In order to highlight the effectiveness of the proposed method, we will show the results of simulations in *search accuracy* and *search time*.

In all simulations, we assume that all agents randomly select their k options in initial time and finish the simulation if all agents collectively find out the best option in any voting.

A. Improvement of Search Accuracy

Firstly, we set that $N = 100, M = 10, \tau = 0.001$ and do 1000 runs for each following case: original BRT algorithm, the proposed method with $k = 2$ (2-BRT) and $k = 3$ (3-BRT). We count the number of simulations where the best option ($n_{success}$) was found out and measure the time elapsed. Success ratio SR is the proportion of $n_{success}$ in 1000 runs:

$$SR \approx \frac{n_{success}}{1000} \quad (23)$$

Fig. 9 shows the cumulative distribution of success ratio of the discovery. The horizontal axis represents the time elapsed, the vertical axis represents the cumulative distributions of the success ratio. We can see that, when the time is sufficient

(more than 30000 steps), all models show very high success ratios. However, although the original BRT shows very high accuracy in long time run, the proposed method shows much higher accuracy than the original BRT does in short time run. For example, within 5000 steps, while with the original BRT algorithm there is only 50% simulations where all agents can find out the best option, with the proposed method the success ratios are about 74% (2-BRT) and 83% (3-BRT).

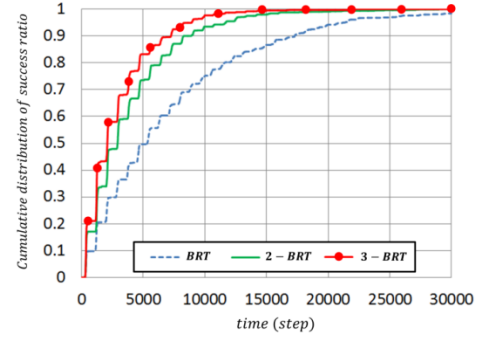


Fig. 9. Cumulative distribution of success ratio of the discovery in 1000 runs. $N = 100, M = 10, \tau = 0.001$.

From the above, we observe that using the proposed method a group of agents can collectively find out the best option among multiple alternatives in very high accuracy even when there is a little time.

B. Improvement of Search Time

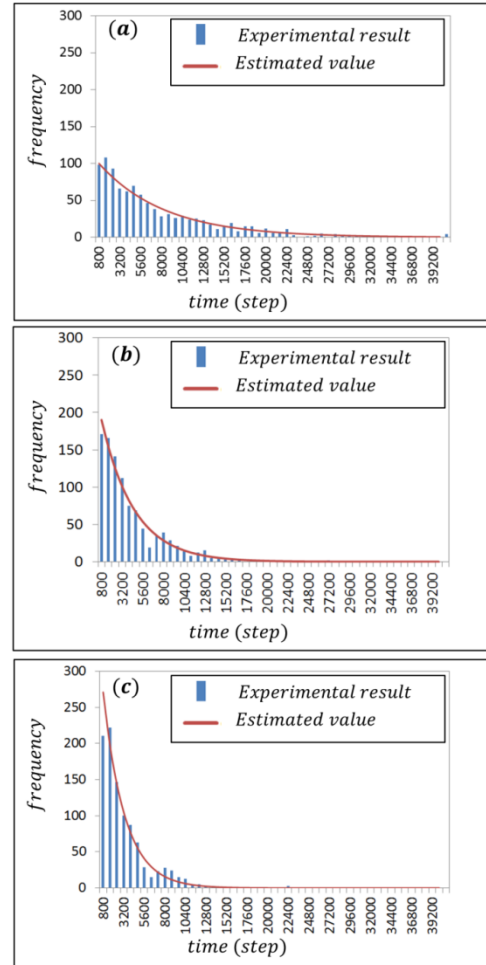


Fig. 10. Distributions of the time required to find out the best option in (a) original BRT model, the proposed method with (b) $k = 2$ and (c) $k = 3$ over 1000 runs. $N = 100, M = 10, \tau = 0.001$.

Here, we focus on clarifying the improvement of search time in the proposed method.

Firstly, in Fig. 10, we show the distributions of the time required to find out the best option over 1000 runs in the cases of the original BRT algorithm, the proposed method: 2-BRT and 3-BRT. The horizontal axis represents the time step and the vertical axis represents its frequency. The experimental result is represented by blue bar, the estimated value using (15) is represented by red line.

From Fig. 10 we can see that, the estimation of (15) fits well in all of 3 cases: the distribution of the time required to find out the best option is an exponential distribution. While the distribution in the case of the original BRT algorithm (Fig. 10(a)) is highly right-skewed and it takes over 40000 steps for a few runs, in the case of the proposed method, the distribution is more highly right-skewed without over 40000 step-run (Fig. 10(b) and (c)). Overall, compared with the results of the original BRT algorithm, the variation of time required to find out the best option using the proposed method 2-BRT is smaller and the smallest is 3-BRT.

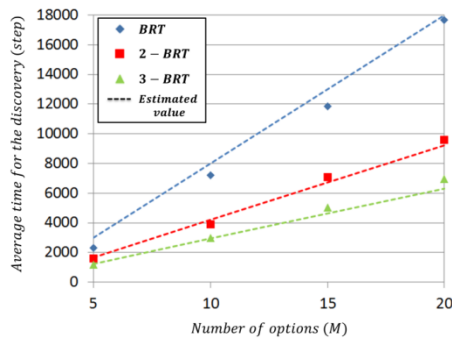


Fig. 11. Average time required to find out the best option in 1000 runs across different number of options M . $N = 100, \tau = 0.001$.

Next, we do the same simulation as above with different the number of options M and investigate time required to find out the best option with each value of M . The result is shown in Fig. 11. The horizontal axis represents the number of options M , the vertical axis represents the average time and dotted line represents the estimated value using (22). The estimation of (22) fits well in all of 3 cases: the average time required to find out the best option is nearly proportional to the number of options. Overall, compared with the original BRT algorithm, it is about 2 and 3 times faster with 2-BRT and 3-BRT in any value of M .

Finally, we observe that the proposed method does not only allow a group of agents to find out the best option among a large number of options faster than the original BRT algorithm but also dramatically reduce the variation of the time required.

V. CONCLUSION

To improve the search capability of BRT algorithm, this paper proposed the additional improvement for the BRT algorithm using multiple votes that each agent votes in each time step. The experimental results show that the proposed method allows a group of agents to solve the best-of-n problem faster with higher accuracy than the original BRT could do. This does not only improve the search capability

but also the applicability in the case of time-limited dynamic environments.

REFERENCES

- [1] 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. 9, no. 9, pp. 1-18, 2017.
- [2] J. Wessnitzer and C. Melhuish, "Collective decision-making and behaviour transitions in distributed Ad Hoc wireless networks of mobile robots: Target-hunting," in *Advances in Artificial Life. ECAL 2003, Lecture Notes in Computer Science*, W. Banzhaf, J. Ziegler, T. Christaller, P. Dittrich, and J. T. Kim, Eds. vol. 2801, 2003, pp. 893-902.
- [3] S. Garnier, F. Tache, M. Combe, A. Grimal, and G. Theraulaz, "Alice in pheromone land: An experimental setup for the study of ant-like robots," in *Proc. 2007 IEEE Swarm Intelligence Symposium*, 2007, pp. 37-44.
- [4] A. Scheidler, A. Brutschy, E. Ferrante, and M. Dorigo, "The k-unanimity rule for self-organized decision making in swarms of robots," *IEEE Transactions on Cybernetics*, vol. 46, no. 4, pp. 1175-1188, 2016.
- [5] S. Garnier, J. Gautrais, M. Asadpour, C. Jost, and G. Theraulaz, "Selforganized aggregation triggers collective decision making in a group of cockroach-like robots," *Adapt. Behav.*, vol. 17, pp. 109-133, 2009.
- [6] G. Francesca, M. Brambilla, V. Trianni, M. Dorigo, and M. Birattari, "Analysing an evolved robotic behaviour using a biological model of collegial decision making," in *Proc. International Conference on Simulation of Adaptive Behavior-From Animals to Animats 12*, vol. 7426, 2012, pp. 381-390.
- [7] H. Hamann, T. Schmickl, H. Wörn, and K. Crailsheim, "Analysis of emergent symmetry breaking in collective decision making," *Neural Comput. Appl.*, vol. 21, pp. 207-218, 2012.
- [8] D. Pais, P. M. Hogan, T. Schlegel, N. R. Franks, N. E. Leonard, and J. A. Marshall, "A mechanism for value-sensitive decision-making," *PLoS One*, vol. 8, no. 9, 2013.
- [9] A. Reina, G. Valentini, C. Fernandez-Oto, M. Dorigo, and V. Trianni, "A design pattern for decentralised decision-making," *PLoS One*, vol. 10, no. 10, 2015.
- [10] A. Reina, J. A. Marshall, V. T. T. Bose, and T. Bose, "Model of the best-of-N nest-site selection process in honeybees," *Physical Review E*, vol. 95, no. 052411, 2017.
- [11] N. H. Phung, M. Kubo, and H. Sato, "Agreement algorithm with trial and error method at macro level," *Artificial Life and Robotics*, vol. 23, no. 4, pp. 564-570, 2018.
- [12] J. S. Mill, *Considerations on Representative Government*, 1st ed. London: Parker, Son, and Bourn, 1861.
- [13] T. Liggett, *Stochastic Interacting Systems: Contact, Voter and Exclusion Processes*, S. S. Chern, B. Eckmann, and P. de la Harpe Eds. Springer, 1999, part 3.
- [14] M. A. M. Oca, E. Ferrante, A. Scheidler, C. Pinciroli, M. Birattari, and M. Dorigo, "Majority-rule opinion dynamics with differential latency: A mechanism for self-organized collective decision-making," *Swarm Intelligence*, vol. 5, no. 3-4, pp. 305-327, 2011.
- [15] P. L. Krapivsky and S. Redner, "Dynamics of majority rule in two-state interacting spin systems," *Physical Review Letters*, vol. 90, no. 23, 2003.
- [16] G. Valentini, *Achieving Consensus in Robot Swarms: Design and Analysis of Strategies for the Best-of-n Problem*, Springer, 2017, ch. 3.
- [17] S. Iwanaga and A. Namatame, "The complexity of collective decision," *Nonlinear Dynamics, Psychology and Life Sciences*, vol. 6, no. 2, pp. 137-158, 2002.
- [18] T. S. Schelling, *Micromotives and Macrobehavior*, New York: W. W. Norton, 1978.
- [19] N. H. Phung, M. Kubo, and H. Sato, "BRT algorithm for the best-of-n problem using short-term experience memory," submitted for publication.



Nhu Hai Phung was born in 1991 and received his B. E degree and M. E degree in computer science of National Defense Academy of Japan in 2015 and 2017. His area of interest includes multi-agent systems, artificial intelligence and swarm intelligence. He is a doctoral student of Department of Computer Science at National Defense Academy of Japan. He has published 2 journal research papers and presented 5 papers in conferences.



Masao Kubo holds the B. E. degree in precision engineering department (Hokkaido University, Japan, 1991), a Ph. D. degree in computer science (Hokkaido University, Japan, 1996). He is an associate professor of Department of Computer Science at National Defense Academy in Japan. His research interest is multi-agent systems.



Hiroshi Sato holds the B. E. degree in Keio University, the M. E. and the Ph. D. degrees in Tokyo Institute of Technology in Japan. He is an associate professor of Department of Computer Science at National Defense Academy of Japan. His research interests include agents-based simulation, evolutionary computation and artificial intelligent.