



Research Article

Probabilistic consensus decision making algorithm for artificial swarm of primitive robots

Yang Liu¹ · Kiju Lee^{2,3} 

Received: 20 August 2019 / Accepted: 4 December 2019 / Published online: 16 December 2019
© Springer Nature Switzerland AG 2019

Abstract

This paper presents a consensus algorithm for artificial swarms of primitive agents, such as robots with limited sensing, processing, and communication capabilities. The presented consensus algorithm provides solutions of collective decision making for a connected network of robots. The decisions are considered abstract choices without difference, thus the algorithm can be “programmed” for a broad range of applications with specific decisions. Each robot in the swarm is considered a probabilistic finite state machine, whose preferences towards a set of discrete states are defined as a probabilistic mass function. Then, the individual preferences are updated via local negotiation with directly connected robots, followed by a convergence improvement process. The presented algorithm is evaluated for the effects of network topology and scalability (i.e., the number of decisions and the size of the swarm) on convergence performance.

Keywords Swarm intelligence · Consensus decision making · Probabilistic finite state machine

List of symbols

\mathbf{R}	Set of all robot indices in a network
m	Quantity of robots in a network, i.e. size of \mathbf{R}
\mathbf{Q}	Set of all decision indices
n	Quantity of decisions, i.e. size of \mathbf{Q}
R_k	Robot k
$P_k(j)$	Robot k 's preference probability toward decision or role j
$s(j)$	Position of $P_k(j)$ in descending order of P_k
C_k	Robot k 's local connection group
D_k	Robot k 's local consensus group
N_k	Size of group D_k
λ_k	Maximum distribution difference in C_k
\mathcal{L}_k	Linear multiplier for robot k
H_k	Discrete entropy of P_k
\mathcal{D}_{rel}	Relative holistic dependency of a network

1 Introduction

A robotic swarm aims to achieve physical and computational flexibility and increased system robustness in multi-robot tasks, such as localization, mapping, and navigation in an unknown, possibly dynamic, environment. The main characteristics of a swarm robotic system includes the following, while slightly different definitions exist across literature [1–3]:

- *Autonomy*: Robots are autonomous. Individual robots and the entire swarm exhibit different levels of autonomy.
- *Localized sensing and communication*: Each robot's sensing and communication capabilities are local.
- *Decentralized control*: Individual robots do not have access to centralized control and global knowledge.

This work was done while K. Lee was with the Department of Mechanical and Aerospace Engineering at Case Western Reserve University, Cleveland, Ohio, USA.

✉ Kiju Lee, kiju.lee@tamu.edu; Yang Liu, yx11450@case.edu | ¹Department of Mechanical and Aerospace Engineering, Case Western Reserve University, Cleveland, OH, USA. ²Department of Engineering Technology and Industrial Distribution, Texas A&M University, College Station, TX, USA. ³Department of Mechanical Engineering, Texas A&M University, College Station, TX, USA.



SN Applied Sciences (2020) 2:95 | <https://doi.org/10.1007/s42452-019-1845-x>

- **Cooperative action:** Robots cooperate with each other to perform an intended task.

These characteristics uniquely qualify robotic swarms to perform certain types of tasks effectively, such as (a) large area coverage within a short time; (b) tasks in dynamic, uncertain, and unstructured environments; (c) tasks that require scaling up or down within the task at hand; and (d) tasks requiring redundancy in information.

Despite these potentials, several engineering challenges must be addressed in order for a swarm robotic system to be applied to tackle real-world problems. One primary challenge is in achieving swarm behavior without centralized control. Most of these previous works have focused on achieving a specific global swarm behavior based on relatively simple rules executed by the individual robots. Target global behaviors included swarm aggregation [4–8], shape/pattern formation [9–16], cooperation on construction works [17, 18], collective transportation [19], structural damage detection [20, 21], and navigation [22–26]. To tackle real-world problems, a robotic swarm must be capable of sequentially performing several of these global behaviors. For example, localizing and retrieving an object located in a narrow tunnel may require the swarm to aggregate, form a line, navigate, and cooperate on carrying the object back to the base location.

In this paper, the problem is formulated as a consensus decision making process given a finite number of choices for individual robots. Individual robots are modeled as Probabilistic Finite State Machines (PFSMs), where their finite states are defined by a set of executable distinctive behavioral rules. A successful global behavior emerges when the majority, if not all, of the robots execute the same rule-set simultaneously. Since local sensing/communication and decentralized control are assumed, decision making must also take place at the individual robot level while consensus in the individual decisions is sought for achieving a global behavior at the swarm level. Each robot's preference towards n possible choices is defined as a Probability Mass Function (PMF). The choice with the highest preference is called the “exhibited decision” of the robot. The presented method aims to achieve consensus based on local communication among the nearby robots and internal processing of the individual preferences, as illustrated in Fig. 1:

1. Modeling individual robots as PFSMs by generating initial preference distributions over given choices;
2. Updating each robot's preferences based on its own and locally connected robots' preferences; and
3. Accelerating convergence and conflict resolving by increasing confidence toward the exhibited decision.

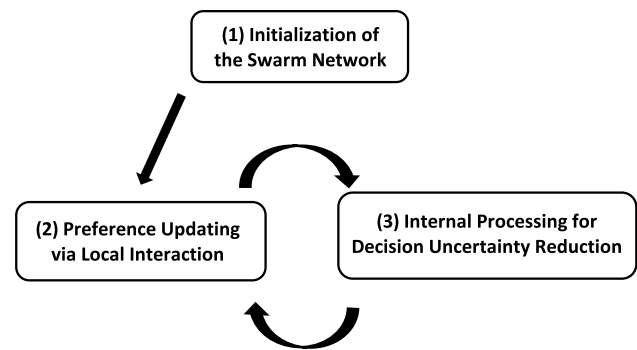


Fig. 1 Flow chart of the presented collective decision making algorithm

The presented algorithm focuses on achieving *guaranteed* consensus over a finite set of abstract decisions for a group of robots in a single network. It achieves consensus in a swarm network regardless of its connectivity density, i.e., consensus can be reached when the network is fully connected or even when sparsely connected. Given a fully connected network condition (i.e., each robot communicates with every other robots), a simple majority rule would be sufficient [27–29], and presented algorithm would fall back to the same majority rule. However, if the network is sparsely connected and only highly localized communication is available, the presented algorithm would more effectively resolve the conflicting decisions within the network than the methods based on the majority rule. The robots in majority rule based methods aim to gather direct information from as many other robots as possible, which is difficult in a sparsely connected network. By targeting consensus in a standalone network, our problem setup and algorithm are well positioned for further designing complex sequential swarm robot behaviors.

In addition to the methods based on the majority rule described above, there exist several other relevant algorithms for collective decision making in a robotic swarm. In [30], a cooperative decision making method for micro-robots with light sensors was used for localization of an area with a higher illuminance based on robot-to-robot collision and onboard sensing. Another approach demonstrated convergence towards the majority decision (i.e., color) between the two randomly initialized choices based on localized interaction among the robots [31]. An ODE-model based method for decision making in a self-organized systems based on the weighted voter model achieved increasing decision accuracy with increasing system size and consensus time [32].

Since the consensus process is often referred to as the *best-of- n* problem—as in this paper, PFSM-based modeling methods have been widely applied for different swarm formation problems, such as aggregation [4]

and chain formation [10]. Another PFSM-based method proposed a probabilistic, threshold-based approach to recruit an approximate number of robots to accomplish a task via collaboration [28]. Biological inspiration has also played an important role in many swarm decision making algorithms. Self-organized aggregation behavior in a robotic swarm was modeled based on cockroach's group behavior [33–35]. For example, cockroach groups decide on shelter selection with limited information and signaling exchanged among individual cockroaches [36]. It is further concluded that such way of interaction resulted in self-amplified aggregation in cockroaches, where the winning choice is the one with a faster self-amplification process [37]. Waggle dance and antennal contacts in honeybees carry data of distance and direction of the feeding site relative to the hive [38]. Collective decision making algorithms utilizing direct verbal communication inspired by such swarm insects were developed for task such as target hunting [27], or resource foraging [39], or nest site comparison [40].

The rest of the paper is organized as follows. Section 2 introduces the presented consensus algorithm; Sect. 3 evaluates the performance of the algorithm with regards to the network size, number of decisions, and network typologies; and finally Sect. 4 discusses the limitations and potential of the presented method.

2 The algorithm

The presented method focuses on achieving consensus in a swarm of simple robots, and thus the following *constraints* are considered:

- Individual robots are primitive with limited sensing, communication, and processing capabilities.
- Communication in the swarm is local; each robot can communicate only with nearby robots within the communication range.
- Robots have no temporal memory (i.e., no log of history data) and function like finite state machines.

It is further assumed that the network topology does not change during the decision making process. If the decision making process is relatively fast enough compared to the robots' physical movements, the change in the network topology would remain trivial while the physical locations of the robots may change during this process.

The overall collective decision making algorithm, illustrated in Fig. 1, consists of three parts: (1) initializing the swarm network with random preference distributions; (2) updating individual preference distributions based on

local interactions; and (3) improving confidence in the exhibited decision. Each node representing a primitive robot follows these steps until a consensus in the swarm is reached. Below describes each of the above three steps in detail.

2.1 Initialization of the swarm network

Let $\mathbf{R} = \{1, 2, \dots, m\}$ be an index set of m robots in the swarm and $\mathbf{Q} = \{1, 2, \dots, n\}$ be the index set of n distinct choices, corresponding to global swarm behaviors. Individual robot is referred to as R_k for $k = 1, \dots, m$. Each robot's preference towards n choices is modeled as a Probability Mass Function (PMF), such that $\sum_{j=1}^n P_k(j) = 1$, where $P_k(j)$ indicates the R_k 's preference toward the choice j . This probability distribution is hereinafter referred to as the *preference distribution*. Each robot exhibits one decision at a time, determined by the corresponding index of $\max\{P_k(1), \dots, P_k(n)\}$. Initial values of these preference distributions are randomly generated for the initial set up of the swarm network.

2.2 Preference updating via local interaction

Each robot updates its own preference by interacting with its neighboring robots within the communication range. For R_k , all neighboring robots of R_k and itself forms a *local connection group*, denoted by C_k . Each robot holds IDs of all members within its local connection group. Robots within the same connection group exchanges their preference distributions. A *local consensus group*, D_k is defined as a non-empty set of the robots connected to R_k that exhibit the same decision as R_k . Each robot within a local consensus group shares information of the IDs of all members, but not their the preference distributions. When a robot decides to join or leave a local consensus group, its closest neighbors in the group can detect the change, and this information is broadcasted within the local consensus group.

Figure 2 shows an example of how these two groups are determined for a network of 8 robots. The node colors indicate the exhibited decision of the robots. When two connected robots exhibit the same decision, the connecting lines are also visualized with the color of the decision. For R_4 , the local connection group is defined by $C_4 = \{2, 3, 4, 5, 6, 7\}$; and the local consensus group—showing the “red” decision—is $D_4 = \{4, 5, 6, 7, 8\}$. As shown in the figure, members of D_4 may not be directly connected to R_4 , but forms a connected network including R_4 . It is noted that R_4 shares IDs and preference distribution with the members of C_4 while sharing only the IDs with the members of D_4 .

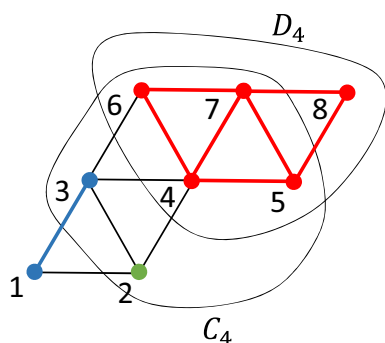


Fig. 2 A network of 8 robots, showing R_4 's local consensus group D_4 and local connection group C_4

Each robot updates its own preference distribution by taking account of preferences of other robots in its local connection group:

$$P_k(j) = \frac{\sum_i N_i P_i(j)}{\sum_i N_i}, \quad i \in C_k, j \in \mathbf{Q} \tag{1}$$

where $N_i = |D_k|$ is the size of the R_i 's local consensus group. This equation is a weighted average of the preference distributions among all directly connected robots, where the weights are determined by the size of the decision group. In the equation, each robot compromises its preference by considering its neighbors' preferences, where the decision agreed by more robots carry a larger weight in this process. If C_k is a subset of D_k , then the decision has been locally converged at R_k , and (1) results in an equally weighted average. The weights proposed in (1) help resolve potential conflicts among all the local consensus groups by favoring large-sized local consensus groups.

2.3 Internal processing for decision uncertainty reduction

Once the following two conditions are satisfied, R_k is considered *locally converged*: (1) local consensus is achieved (i.e., $C_k \subset D_k$) and (2) the maximum difference of the preference distributions within C_k is below a threshold. The maximum difference in the preference distributions among the members of C_k is defined as λ_k and calculated by

$$\lambda_k = \max_{k1, k2 \in C_k, k1 \neq k2} \left(\sum_{j=1}^n |P_{k1}(j) - P_{k2}(j)| \right). \tag{2}$$

λ_k is a measure of the degree of divergence in C_k . If $\lambda_k < \lambda_T$ is satisfied for an empirical threshold value λ_T , then R_k is considered being confident about its own exhibited decision.

Once the above conditions are satisfied, R_k 's preference distribution is further updated to accelerate the

convergence process by multiplying a linear multiplier, $\mathcal{L}_k(j)$, for $j \in \mathbf{Q}$:

$$P_k^{new}(j) = P_k(j) \cdot \mathcal{L}_k(s(j)) \tag{3}$$

where $s(j)$ is introduced to rearrange the probabilities in P_k in a descending order, such that $P_k(j)$ is in $s(j)$ position in the new order. \mathcal{L}_k is constructed as follows:

$$\mathcal{L}_k(j) = L_l \frac{j-1}{n-1} + L_u \frac{n-j}{n-1}, \quad j \in \mathbf{Q} \tag{4}$$

where

$$L_l = \frac{1}{n} \left(\frac{\lambda_k}{\lambda_T} \right)^{0.3}; \quad L_u = \frac{2}{n} - L_l.$$

L_l and L_u are the lower and upper ends of the linear multiplier. This process reduces the uncertainty on the exhibited decisions of individual robots by increasing the preference values of the highly preferred choices and further reducing the preferences values of the less preferred choices. This process of convergence improvement is explained with an example in Fig. 3.

3 Algorithm evaluation

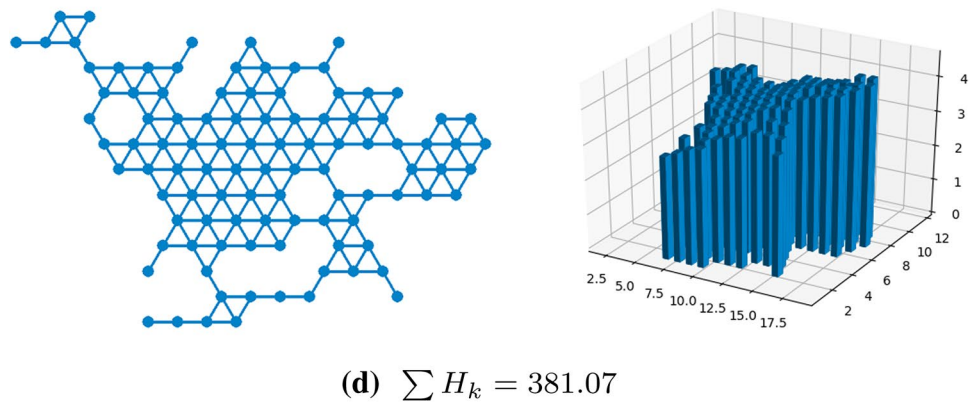
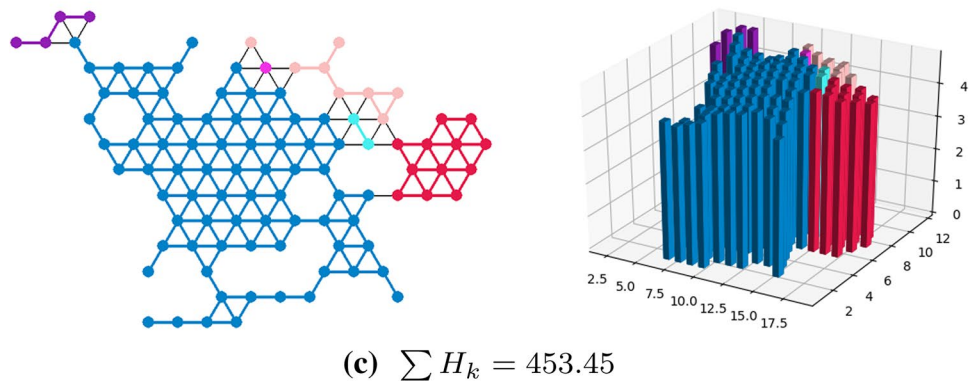
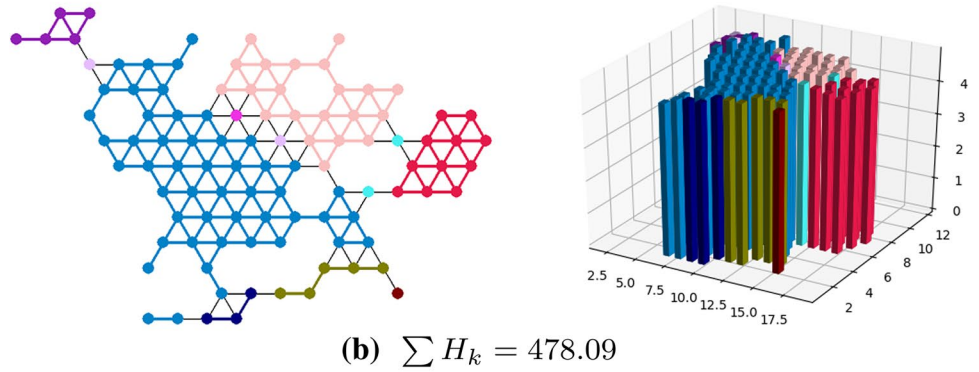
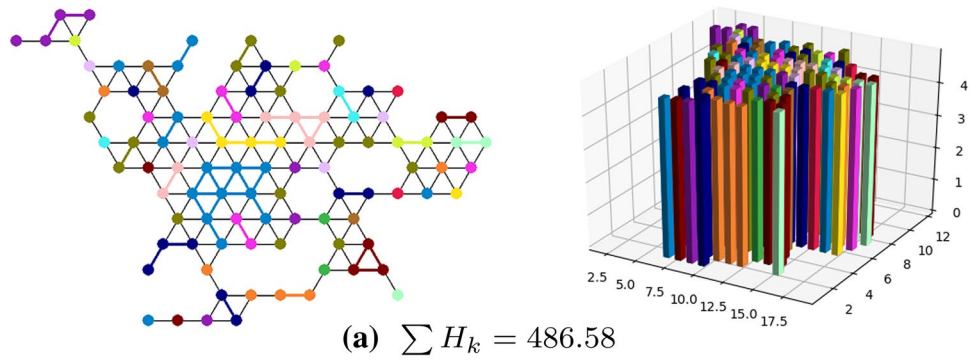
This section evaluates the presented consensus decision making algorithm in terms of (1) the effect of network topology in algorithm performance; (2) scalability regarding the network size (m) and the number of decisions (n); (3) the effect of external interference on the final decision; and (4) comparison with other consensus achievement algorithms. The performance is measured by the number of iterations required to achieve convergence.

3.1 Simulation environment and measure of uncertainty

The simulations are implemented in Python, with Pygame library for visualizing the virtual robots and the environment, and Matplotlib library for graph analysis. The hosting computer has Intel Core i5 CPU, 8GB of memory, and AMD integrated graphics card, and operates Ubuntu 16.04.

The equilateral triangle grid pattern is used to generate random networks. Nodes are placed at the joints of the edges, and adjacent nodes are automatically connected if there is an edge in between. Since the robots are assumed to have limited sensing and communication range, the 2D equilateral triangle grid is a reasonable model to represent the 2D network configuration of a robotic swarm. The triangle grid limits the number of neighbors of a robot to a maximum of 6. For a larger number of neighbors, more

Fig. 3 The consensus process with 3D bar graphs visualizing convergence improvement. At each step of iteration, summation of discrete entropy of each robot's preference distribution is calculated as $\sum H_k$. The lower the value, the faster the convergence rate



complex grid patterns may be used. The networks are created by randomly placing new nodes one by one to the surroundings of existing nodes.

To quantify the consensus process, the summation of discrete entropy values calculated for individual preference distributions is used: $\sum_{k=1}^m H_k$, where $H_k = -\sum_{j=1}^n P_k(j) \log_2 P_k(j)$. H_k reaches the maximum when the preference distribution is uniform, such that $P_k(1) = \dots = P_k(n) = 1/n$, resulting in $H_k = \log_2 n$. It becomes the minimum when R_k has 100% preference probability towards one specific decision and 0% towards the rest, resulting in $H_k = 0$. A smaller value of H_k implies that R_k is more decisive for its exhibited decision. The presented consensus algorithm aims to achieve a consensus with lower uncertainties in the preference distributions.

Figure 3 shows a random 100-node network ($m = 100$) and its convergence towards a consensus using the algorithm described in Sect. 2. Different colors indicate exhibited decisions of individual nodes among 30 possible choices ($n = 30$). An edge between two nodes turns into a specific color if they form a local consensus group, D_k . The 3D bar graph visualizes the discrete entropy value of the preference distributions of individual nodes. The bar for each node is placed in its corresponding location within the network on the x - y plane. The color represents its exhibited decision and the height indicates the calculated value of $\sum H_k$. At the beginning of the simulation, the preference distributions are randomly generated over 30 choices. As the consensus process proceeds, several local consensus groups are formed and eventually all nodes converge to the same consensus group.

3.2 Effect of network topology on convergence performance

Topology plays a significant role in the consensus process. To unveil how convergence performance is affected by the network topology, a new concept of *network dependency* is introduced. Network dependency, denoted as

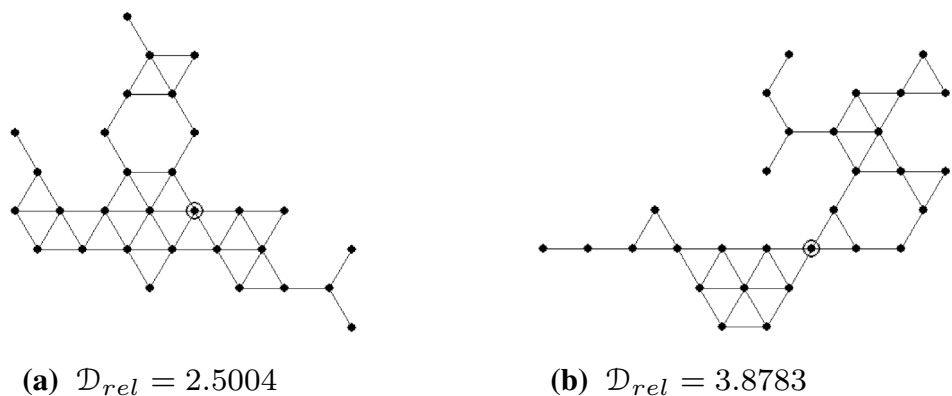
\mathcal{D}_{rel} , measures the degree of dependency of the rest of the nodes on the most reliable node in order to maintain network connectivity. This representation differs from the connectivity in the graph theory, which is defined as the minimum number of elements (i.e., nodes or edges) required for disconnecting one node from the rest of the nodes. The traditional definition puts too much weight on the weakest part of the network, without considering the network as a whole. The new definition, \mathcal{D}_{rel} , takes a holistic view of the network, which is defined as the ratio of the maximum individual dependency to the average dependency value, such that

$$\mathcal{D}_{rel} = \frac{\mathcal{D}_{max}}{\mathcal{D}_{mean}} \tag{5}$$

\mathcal{D}_{max} is the maximum value of the individual dependency of all nodes, and \mathcal{D}_{mean} is the average value (See Algorithm 1 in "Appendix 1" for details). The calculated \mathcal{D}_{rel} value is usually within the range of [1, 5] for randomly generated networks with up to 150 nodes. Figure 4 shows two different networks with 30 nodes and their \mathcal{D}_{rel} values. The node with an outer circle indicates the node with the highest dependency. A higher value of \mathcal{D}_{rel} indicates that the network has a higher dependency on a specific node (or a small number of nodes) where many other nodes rely on in order to establish connection.

To evaluate the effect of network topology, the network size and number of decisions are fixed at 30. Thirty random networks are generated and \mathcal{D}_{rel} is calculated for each. The \mathcal{D}_{rel} for these networks range between 2.2 and 4.3. Figure 5 shows the number of iterations versus \mathcal{D}_{rel} . The graph shows that \mathcal{D}_{rel} is moderately correlated with the number of iterations ($\rho = 0.65$), indicating that a network with a higher \mathcal{D}_{rel} requires more iterations for convergence. It is also noted that the network connectivity is a multi-dimensional property and any single measure may not be comprehensive enough to cover all topology-related properties.

Fig. 4 \mathcal{D}_{rel} values for two randomly generated networks with 30 nodes



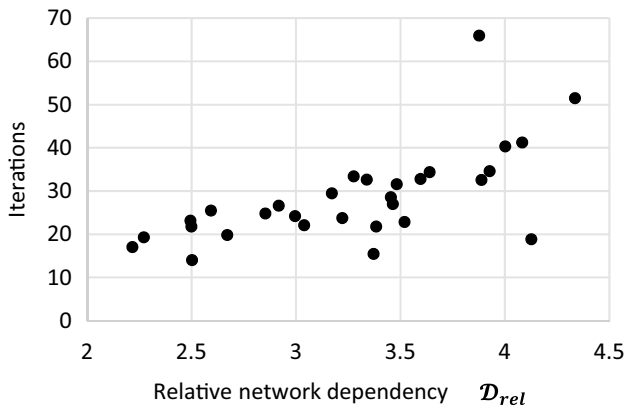


Fig. 5 Number of iterations versus \mathcal{D}_{rel}

3.3 Scalability evaluation

Scalability is evaluated for the network size (m in the range of [30, 150] with step size of 10) and the number of available decisions (n in the range of [10, 300] with step size of 10). The network size and decision numbers are chosen to be as large as possible so as to pose a challenge to presented algorithm, but at same time representative of the kind of problems it is designed to solve. For each test case, 100 simulations are performed and averaged number of iterations are calculated.

Figure 6 shows the results for iterations versus m . In this experiment, n is fixed at 30 and \mathcal{D}_{rel} is controlled at 3.0 ± 0.1 . The error bars measure the standard deviation of the 100 trials for each situation. A positive correlation between the number of iterations and m has been found with correlation $\rho = 0.87$. The noise in the linearity is mainly caused by the difference of \mathcal{D}_{rel} in the networks. To evaluate the effect of network size on the convergence

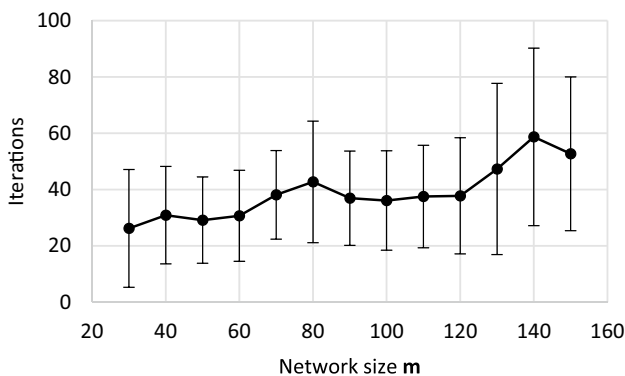


Fig. 6 Number of iterations versus network size m

rate, all the rest of the parameters must be fixed. While n can be fixed, \mathcal{D}_{rel} cannot be completely controlled for randomly generated networks. Instead, the networks were carefully chosen to have \mathcal{D}_{rel} values as close to each other as possible within 0.1 difference. Figure 7 shows the number of iterations required for convergence over the number of decisions, n . This simulation is performed in a network with $m = 30$ and $\mathcal{D}_{rel} = 2.672$. The graph shows a stable increase of iterations along with the increase of the number of decisions. The calculated correlation is $\rho = 0.91$.

3.4 External interference

This subsection demonstrates the effect of external interference on the decision making process, and thus shows how to achieve controllable collective decision making using the presented algorithm. In one case scenario, a small group of robots receives an external command or detects a significant event from the environment that triggers these robots to exhibit a specific decision. The robots that adopt such external inputs are referred to as *seed* robots. In many cases, the seed robots govern the entire decision making process, and thus can effectively control the final decision. If the number of seed robots is too small compared to the network size, or if these robots are located in the corner, the rest of the robots may converge to another decision and may override the seed robots' decision.

Figure 8a shows a simulation of 100 robots with 10 seed robots holding the same exhibited decision initially located in the middle of the network. The exhibited decision of the seed robots shown in "blue" quickly dominates the entire swarm. All 100 randomized trials result in convergence of "blue" decision, with average of 16.99 iterations. It is worth noting that this number is significantly

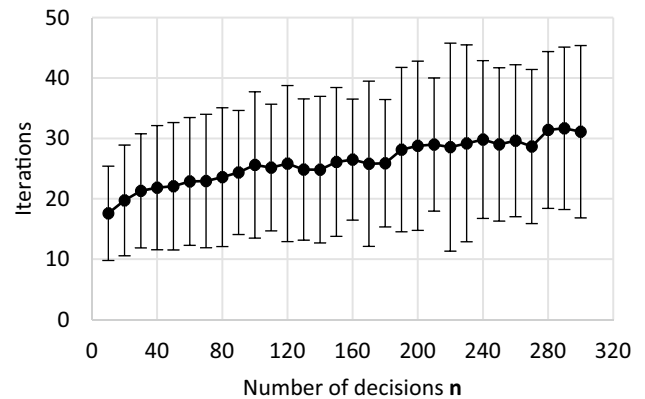
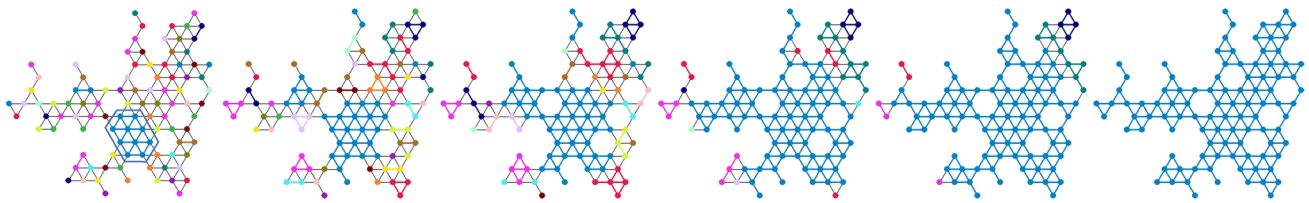
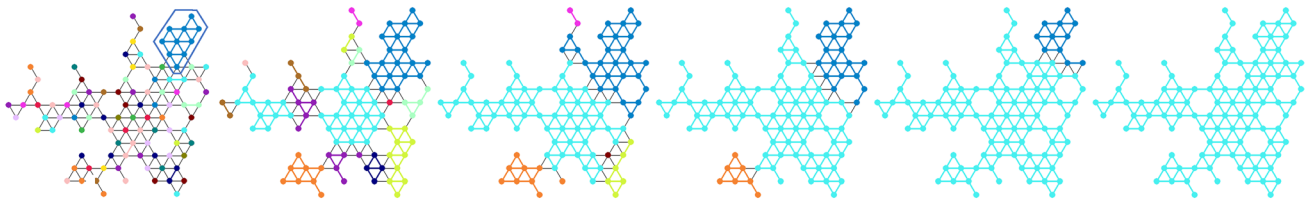


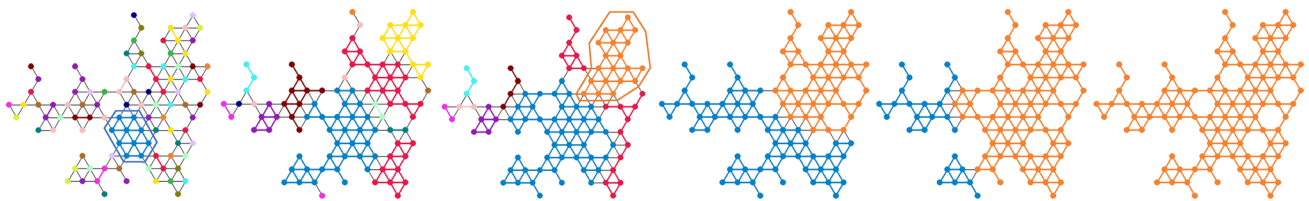
Fig. 7 Number of iterations versus number of decisions n



(a) Collective decision making with 10 seed robots (blue) in the middle of the network gradually dominating the process to converge to the “blue” decision.



(b) Collective decision making with 10 seed robots (blue) on the up-right corner of the network demonstrating the location of the seed robots in a specific topology affects the overall consensus result.



(c) Collective decision making with 10 seed robots (blue); while 20 robots detect an adverse event (orange) which can override the consensus process previously dominated by the “blue” decision.

Fig. 8 Simulations demonstrating controllable swarm behavior under three different situations. The results show that the presented algorithm can be used for achieving controllable swarm

behavior while also autonomously reacting to unexpected conditions to avoid “catastrophic” failure by overriding provided control input

lower than that without the seed robots presented (i.e., 38.88 iterations). In Fig. 8b, 10 seed robots are located at the upper-right corner of the network at the beginning, and they fail to control the swarm to converge to the “blue” decision. Figure 8c shows another scenario demonstrating that 10 seed robots start dominating the decision making process, until 20 other robots detect or receive some critical information triggering them to exhibit the “orange” decision. Since this decision would be difficult to override, the swarm gradually reaches to the orange decision collectively.

Figure 9 further examines the relationships between the number of seed robots on the convergence behavior when the seed robots are located around the middle of the

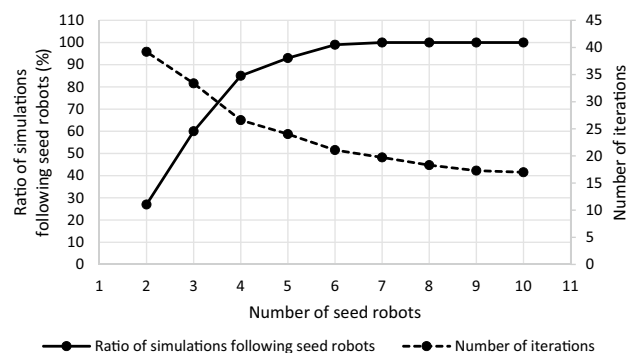


Fig. 9 Ratio of simulations following seed robots’ decision and number of iterations versus number of seed robots located in the middle of the network

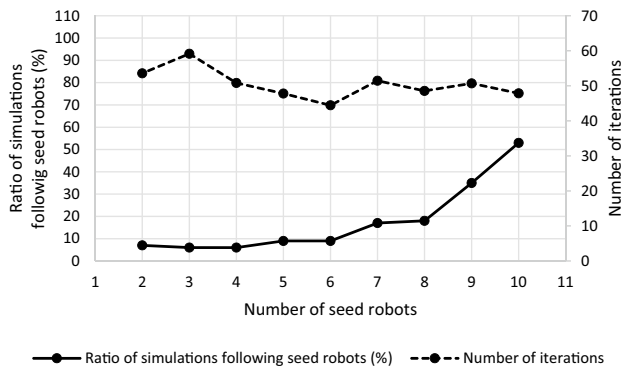


Fig. 10 Ratio of simulations following seed robots and number of iterations versus number of seed robots located on the top right corner of the network

network. Each data point is the average of 100 trials. With 2 seed robots, only 27% of the simulations converge to the “blue” decision. However, with 6 or more seed robots, the convergence rate reaches 100%. The other curve with dashed line shows the average number of iterations for those simulations ruled by the seed robots. With more seed robots it took fewer iterations to converge. Figure 10 shows the results when the seed robots are located in a corner of the network. Seed robots fewer than 6 are not enough to lead the swarm to the target decision. With 10 seed robots, about 53% convergence rate towards the seed robots’ decision is achieved.

3.5 Algorithm comparison

It is noted that our algorithm and the problem setup are significantly different from other existing methods, and therefore, direct comparison is not applicable. The statistical results in this subsection are largely based on empirical estimations with the purpose of putting different methods on the same benchmark, while the environmental settings, assumptions about the robots, and implementation methods may be significantly different from each other.

The chosen *test* algorithm is a collective comparison strategy which aims to find the best among the alternatives by individual robots estimating the targets and communicating with nearby robots [40]. The experimental scenario involves many robots moving around in a large arena with two target sites to choose from. The robot observes the quality of a target upon identifying one, then delays for T seconds before finding a teammate and sends it a

recruit-message. The other robot will evaluate the message by itself and decide its favored target. The key of the test algorithm is that the wait time after target observation is inversely proportional to the observed quality. The higher the target quality, the less time it waits, therefore the more robots it can recruit.

The presented algorithm is similar to this test algorithm in that the robots aim to settle on one target among multiple targets. Each robot can have its own observation about the targets and be influenced by nearby robots through local communication. The difference between the two algorithms is that the robots in the test algorithm observe the targets by physically moving towards them and examining them, while the robots in our algorithm are initialized with random preference towards the targets. 10–15 robots are considered in the arena with two targets to choose from. The number of iterations is used as a convergence performance measure. In one iteration, robots communicate once with nearby robots. Our algorithm requires less than 20 iterations on average among 100 trials to converge. The test algorithm requires approximately 1000–1250 seconds to converge. If the averaged moving time between two iterations is estimated to be 10 s, then about 100–125 iterations are required for convergence. If about 50 s are considered per iteration, convergence would require around 20–25 iterations. The estimated numbers of iterations for the test algorithm can be significantly different based on the general wait time and move time between the robot interactions. Nonetheless, the results imply that the presented algorithm can be comparable to, or possibly faster than, an existing method. For further validation of the algorithm, experimental evaluations using physical robots and a benchmarking scenario must be followed.

4 Conclusion

A new distributed collective decision making algorithm for swarm robotic applications has been presented in this paper. Individual robots are assumed to be primitive with limited sensing, communication, and processing capabilities. Under this assumption, while individual robots may exhibit any of n possible decisions, the swarm can only exhibit a global behavior if most of individual robots, if not all, agree on a specific target goal. The presented

algorithm achieves guaranteed consensus in a connected network regardless of the number of decisions, network sizes, and topologies, while the speed of convergence can be affected by these factors. The effect of different topologies, network size, and the number of decisions in convergence performance has been evaluated. Convergence performance under external inputs has also been simulated to demonstrate controllable consensus processes under different situations.

The proposed algorithm relies on the robots knowing the size of the local consensus group in order to resolve conflicts with dissident robots. However, when a robot quits or joins a local consensus group, there is delay for other robots in the group to be updated, and such delay is not considered in our simulation. In a small network size with less than 100 robots, the effect of the delay should be negligible, but it may become evident with a larger network size. For future work, the effect of this delay will be taken account in the simulation as well as in physical experiments. To further improve the convergence rate, simulations covering more variations of this algorithm

must be performed. And they may combine probability distribution with local information such as a local measure of group connectivity for resolving conflicts faster.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Appendix

Algorithm 1 shows how \mathcal{D}_{rel} is calculated for each network. \mathcal{D}_{ind} is first calculated for individual nodes for their dependencies, describing the degree of how much each node is being depended on for maintaining remote node connections. For all shortest paths between node k_1 and k_2 , each shares equal dependency value $1/size(P)$, and it will be added to all nodes except the starting and ending nodes. This algorithm returns the relative network dependency.

Algorithm 1: Network Dependency

```

input : A set of node positions  $S$  for the network.
output: Relative network dependency  $\mathcal{D}_{rel}$ 
 $\mathcal{D}_{ind} = [0, 0, \dots, 0]$ 
for  $\{k_1, k_2\} \subseteq S$  do
    find all shortest paths  $P = \{p_1, p_2, \dots\}$  connecting  $k_1$  and  $k_2$ 
     $\mathcal{D}_{\Delta} = 1/size(P)$ 
    for  $p \in P$  do
        for  $i \in p$  do
            if  $i \neq k_1$  and  $i \neq k_2$  then
                 $\mathcal{D}_{ind}[i] = \mathcal{D}_{ind}[i] + \mathcal{D}_{\Delta}$ 
            end
        end
    end
end
 $\mathcal{D}_{mean} = average(\mathcal{D}_{ind})$ 
 $\mathcal{D}_{max} = max(\mathcal{D}_{ind})$ 
 $\mathcal{D}_{rel} = \mathcal{D}_{max}/\mathcal{D}_{mean}$ 
return  $\mathcal{D}_{rel}$ 

```

References

- Brambilla M, Ferrante E, Birattari M, Dorigo M (2013) Swarm robotics: a review from the swarm engineering perspective. *Swarm Intell* 7(1):1–41
- Mohan Y, Ponnambalam SG (2009) An extensive review of research in swarm robotics. In: 2009 World congress on nature and biologically inspired computing (NaBIC). IEEE, pp 140–145
- Şahin E (2004) Swarm robotics: from sources of inspiration to domains of application. In: International workshop on swarm robotics, vol 17. Springer, Berlin, pp 10–20
- Soysal O, Şahin E (2005) Probabilistic aggregation strategies in swarm robotic systems. In: Swarm intelligence symposium, 2005. SIS 2005. Proceedings 2005 IEEE. IEEE, pp 325–332
- Soysal O, Şahin E (2006) A macroscopic model for self-organized aggregation in swarm robotic systems. In: International workshop on swarm robotics, vol 30. Springer, Berlin, pp 27–42
- Timmis J, Ismail AR, Bjerknes JD, Winfield AF (2016) An immune-inspired swarm aggregation algorithm for self-healing swarm robotic systems. *Biosystems* 1(146):60–76
- Amjadi AS, Raoufi M, Turgut AE, Broughton G, Krajník T, Arvin F (2019) Cooperative pollution source localization and cleanup with a bio-inspired swarm robot aggregation. *arXiv preprint arXiv:1907.09585*
- Ramroop S, Arvin F, Watson S, Carrasco-Gomez J, Lennox BA (2018) Bio-inspired aggregation with robot swarm using real and simulated mobile robots. In: Annual conference towards autonomous robotic systems, vol 25. Springer, Cham, pp 317–329
- Shucker B, Bennett JK (2007) Scalable control of distributed robotic macrosensors. In: Distributed autonomous robotic systems, vol 6. Springer, Tokyo, pp 379–388
- Nouyan S, Campo A, Dorigo M (2008) Path formation in a robot swarm. *Swarm Intell* 2(1):1–23
- Derakhshandeh Z, Gmyr R, Richa AW, Scheideler C, Strothmann T (2016) Universal shape formation for programmable matter. In: Proceedings of the 28th ACM symposium on parallelism in algorithms and architectures. ACM, pp 289–299
- Yang J, Wang X, Bauer P (2018) Line and V-shape formation based distributed processing for robotic swarms. *Sensors* 18(8):2543
- Wang Q, Mao X, Yang S, Chen Y, Liu X (2018) Grouping-based adaptive spatial formation of swarm robots in a dynamic environment. *Int J Adv Robot Syst* 15(3):1729881418782359
- Rubenstein M, Cornejo A, Nagpal R (2014) Programmable self-assembly in a thousand-robot swarm. *Science* 345(6198):795–799
- Klavins E (2007) Programmable self-assembly. *IEEE Control Syst* 27(4):43–56
- Spears WM, Heil R, Zarzhitsky D (2005) Artificial physics for mobile robot formations. In: 2005 IEEE international conference on systems, man and cybernetics, vol 3. IEEE, pp 2287–2292
- Ardiny H, Witwicki S, Mondada F (2015) Construction automation with autonomous mobile robots: a review. In: 2015 3rd RSI international conference on robotics and mechatronics (ICROM). IEEE, pp 418–424
- Werfel J, Petersen K, Nagpal R (2014) Designing collective behavior in a termite-inspired robot construction team. *Science* 343(6172):754–758
- Ferrante E, Brambilla M, Birattari M, Dorigo M (2013) Socially-mediated negotiation for obstacle avoidance in collective transport. In: Distributed autonomous robotic systems. Springer, Berlin, pp 571–583
- Shakya A, Mishra M, Maity D, Santarsiero G (2019) Structural health monitoring based on the hybrid ant colony algorithm by using Hooke-Jeeves pattern search. *SN Applied Sciences* 1(7):799
- Mishra M, Barman SK, Maity D, Maiti DK (2019) Ant lion optimization algorithm for structural damage detection using vibration data. *J Civ Struct Health Monit* 9(1):117–136
- Di Caro GA, Ducatelle F, Gambardella LM (2009) Wireless communications for distributed navigation in robot swarms. In: Workshops on applications of evolutionary computation, vol 15. Springer, Berlin, pp 21–30
- Ducatelle F, Di Caro GA, Pinciroli C, Mondada F, Gambardella L (2011) Communication assisted navigation in robotic swarms: self-organization and cooperation. In: 2011 IEEE/RSJ international conference on intelligent robots and systems. IEEE, pp 4981–4988
- Howard A, Mataric MJ, Sukhatme GS (2002) Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem. In: Distributed autonomous robotic systems, vol 5. Springer, Tokyo, pp 299–308
- Marcolino LS, dos Passos YT, de Souza AA, dos Santos Rodrigues A, Chaimowicz L (2017) Avoiding target congestion on the navigation of robotic swarms. *Auton Robots* 41(6):1297–320
- Inácio FR, Macharet DG, Chaimowicz L (2018) United we move: decentralized segregated robotic swarm navigation. In: Distributed autonomous robotic systems. Springer, Cham, pp 313–326
- Wessnitzer J, Melhuish C (2003) Collective decision-making and behaviour transitions in distributed ad hoc wireless networks of mobile robots: target-hunting. In: European conference on artificial life. Springer, Berlin, pp 893–902
- Kanakia A, Klingner J, Correll N (2016) A response threshold sigmoid function model for swarm robot collaboration. In: Distributed autonomous robotic systems. Springer, Tokyo, pp 193–206
- de Oca MAM, Ferrante E, Scheidler A, Pinciroli C, Birattari M, Dorigo M (2011) Majority-rule opinion dynamics with differential latency: a mechanism for self-organized collective decision-making. *Swarm Intell* 5(3–4):305–327
- Schmickl T, Thenius R, Moeslinger C, Radspieler G, Kernbach S, Szymanski M, Crailsheim K (2009) Get in touch: cooperative decision making based on robot-to-robot collisions. *Auton Agents Multi-Agent Syst* 18(1):133–55
- Vigelius M, Meyer B, Pascoe G (2014) Multiscale modelling and analysis of collective decision making in swarm robotics. *PLoS ONE* 9(11):e111542
- Valentini G, Hamann H, Dorigo M (2014) Self-organized collective decision making: the weighted voter model. In: Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems. International Foundation for Autonomous Agents and Multiagent Systems, pp 45–52
- Francesca G, Brambilla M, Trianni V, Dorigo M, Birattari M (2012) Analysing an evolved robotic behaviour using a biological model of collegial decision making. In: International conference on simulation of adaptive behavior. Springer, Berlin, pp 381–390
- Correll N, Martinoli A (2007) Modeling self-organized aggregation in a swarm of miniature robots. In: IEEE 2007 international conference on robotics and automation workshop on collective behaviors inspired by biological and biochemical systems, no SWIS-CONF-2007-002
- Campo A, Garnier S, Dédrich O, Zekkri M, Dorigo M (2011) Self-organized discrimination of resources. *PLoS ONE* 6(5):e19888
- Am'e JM, Halloy J, Rivault C, Detrain C, Deneubourg JL (2006) Collegial decision making based on social amplification leads to optimal group formation. *Proc Natl Acad Sci* 103(15):5835–5840
- Garnier S, Gautrais J, Asadpour M, Jost C, Theraulaz G (2009) Self-organized aggregation triggers collective decision making in a group of cockroach-like robots. *Adapt Behav* 17(2):109–133

38. Rohrseitz K, Tautz J (1999) Honey bee dance communication: waggle run direction coded in antennal contacts? *J Comp Physiol A* 184(4):463–470
39. Gutiérrez A, Campo A, Monasterio-Huelin F, Magdalena L, Dorigo M (2010) Collective decision-making based on social odometry. *Neural Comput Appl* 19(6):807–823
40. Parker CA, Zhang H (2011) Biologically inspired collective comparisons by robotic swarms. *Int J Robot Res* 30(5):524–535

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.