Information Sciences 608 (2022) 1010–1028

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Fast distributed consensus seeking in large-scale and high-density multi-agent systems with connectivity maintenance

Guangqiang Xie [a], Haoran Xu [a], Yang Li [a,*], Xianbiao Hu [b], Chang-Dong Wang [c]

[a] School of Computer Science and Technology, Guangdong University of Technology, Guangzhou 510006, China
[b] Department of Civil and Environmental Engineering, The Pennsylvania State University, 221B Sackett Building, University Park, PA 16802-1408, USA
[c] School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China

ARTICLE INFO

Article history:
Received 15 December 2021
Received in revised form 21 June 2022
Accepted 22 June 2022
Available online 27 June 2022

Keywords:
Multi-agent system
Consensus seeking
Distributed control
Constraint set
Connectivity maintenance

ABSTRACT

With the rapid development of wireless communication and localization technologies, multi-agent systems (MASs) have emerged as a powerful distributed artificial intelligence for consensus control. However, the connectivity of an MAS with a limited sensing range is vulnerable to the evolution of agents with high mobility, so converging to one common equilibrium rapidly while maintaining connectivity under a large-scale and high-density topology has been a research challenge. To address this problem, we develop a heuristic combinatorial algorithm that combines a distributed sector-division-based (SDB) consensus algorithm and a *d*-subgraph (DSG) connectivity maintenance algorithm. First, the communication region is heuristically divided into multiple sectors, and agents select representative neighbors to calculate control inputs by simultaneously considering the number and the distribution of perceived neighbors. Second, the *d*-subgraph is designed to update agents in accordance with constraint set constructed from the nearest neighbors in their locally perceived connected components. Thus, sequentially combining the SDB and the DSG algorithms affords a heuristic combinatorial (SDB&DSG) algorithm that can effectively accelerate convergence to a common equilibrium and retain global connectivity. The theoretical proofs of connectivity and convergence are given geometrically. Extensive simulations demonstrate the superiority of our algorithm, especially in large-scale and high-density topologies.

© 2022 Elsevier Inc. All rights reserved.

1. Introduction

Recently, with the rapid development of wireless communication and localization technologies, distributed consensus algorithms have become the basis for cooperation and coordination between agents. These algorithms have received extensive attention and application in fields such as cooperative control [12,14,36], tracking [42,4,26], control engineering [30,13,37], event-triggered control [2,16,15] and task allocation [27]. In a multi-agent system (MAS), "consensus" means that the agents tend to be the same for certain state values, such as position, speed, and decision value. The consensus algorithm indicates the rule specifying the information exchange between agents and their neighbors. One of its main goals is to allow

* Corresponding author.
E-mail address: liyang@gdut.edu.cn (Y. Li).

https://doi.org/10.1016/j.ins.2022.06.079
0020-0255/© 2022 Elsevier Inc. All rights reserved.

each agent to use only locally perceived information to achieve global consensus of a certain state value (i.e., a stable equilibrium) [39,1].

Multi-agent consensus problems have received considerable attention in dynamic networks. Jadbabaie et al. [11] conducted a theoretical analysis of a model using the matrix method, and discovered that the system reaches a consensus if the network remained connected. Olfati-Saber et al. [23] proposed a theoretical framework for consensus problems and presented the design of the most common consensus algorithm, classified into continuous-time (CT) and discrete-time (DT) consensus algorithms. Wen et al. [38] constructed several new classes of multiple Lyapunov functions for consensus tracking in leader-following MASs with directed switching topologies. Xie et al. [39] proposed a motif-aware weighted MAS for enhancing the consensus of a small set of clusters. Connectivity maintenance is fundamental in multi-agent consensus problems, which has attracted research attention from many scholars. Various studies [40,44,31,10,45] have shown that the CT consensus algorithm can eventually reach a consensus if the connectivity of the entire network is maintained during the evolution process of agnets, regardless of the network topology, i.e., fixed or switching. The convergence condition of the system for the DT consensus algorithm is similar to that for the CT consensus algorithm when the step size is less than the inverse of network maximum degree [22]. The importance of the connectivity of network topology has also been demonstrated in [17,20,29].

For connectivity maintenance under the DT consensus, Parasuraman et al. [24] developed a hierarchical tracking-based rendezvous control strategy with theoretically guaranteed convergence and connectivity. Cortes et al. [3] proposed a circumcenter algorithm (CA), and demonstrated that it constrains each agent in a set known as a constraint set. This set maintains the existing topology while maximizing the motion range of each agent toward the target location of the convergence evolution. Many researchers have also developed connectivity maintenance methods based on various frameworks, including the constraint function [28,6,41], leaders and followers [25,8,5,33], topology features [32,34], and heuristic reasoning [43,17].

Despite the aforementioned advances in consensus control with connectivity maintenance, there remains no efficient approach for analyzing consensus in large-scale and high-density MASs with switching topologies. The large scale and high density of these MASs subjects each agent to a significant number of its neighbors' connectivity constraints, which means that such MASs tend to split into multiple clusters with a slower convergence speed. The solution to this problem is to equip each agent with an intelligent selection of its neighbors' connectivity constraints to eliminate unnecessary connectivity constraints. Moreau [18] noted that more communication between agents may not accelerate system convergence and may result in the loss of convergence performance. Motsch et al. [19] indicated that heterophily plays a decisive role in the process of clustering, i.e., a faster convergence speed and a better convergence effect with fewer clusters can be achieved by communicating more with those agents who are different than with those agents who are similar.

Inspired by the above-described ideas [18,3,19], this paper develops a sector-division-based consensus algorithm (SDB) combined with a connectivity maintenance algorithm with a d-subgraph (DSG) under an undirected and switching topology. We denote this the SDB&DSG algorithm, and show that it has excellent convergence performance in large-scale and high-density topologies with provable connectivity and convergence guarantees. First, agents perform a simple evaluation of the number and distribution of perceived neighbors, so they can achieve effective aggregation before calculating the control input. Moreover, they do not compute the control input via a baseline algorithm. The developed DSG algorithm for network connectivity maintenance effectively reduces the influence of connectivity constraints, thus accelerating the system's convergence. Specifically, each agent only maintains the connectivity of its local subgraph during the convergence evolution. This DSG algorithm is a general form of the traditional r-disk (RD) algorithm [3], and thus is more versatile and flexible than the RD algorithm. We introduce the convex hull and driving points to analyze the convergence and connectivity of our algorithms. Finally, the simulation results show our algorithms are superior to previously reported algorithms, especially when applied to large-scale and high-density topologies.

The remainder of this paper is organized as follows. Section 2 presents some background information on our developed algorithms and the problem statement. Section 3 describes the design of the SDB and the DSG algorithms, and provides the theoretical proofs of their connectivity and convergence. In Section 4, extensive simulation results are provided for different topologies. Section 5 summarizes the contributions made by this paper and discusses future work.

2. Preliminaries

2.1. Graph theory and matrix theory

MAS is represented by an undirected and switching graph $G(t) = (V, E(t))$, where $V = \{1, \ldots, n\}$ denotes the agents, and $E(t)$ denotes the neighbor relationship between agents. Suppose that there are $n \in \mathbb{N}^+$ agents with states $X(t) = \{x_1(t), \ldots, x_n(t)\} \in \mathbb{R}^{n\times 2}$ on a two-dimensional plane, where $x_i(t)$ represents the position of agent i and each agent has the same sensing range r_c, if any two agents i and j satisfy $\|x_i(t) - x_j(t)\| \leqslant r_c$, they are considered neighbors and thus can exchange information with each other. $N_i(t)$ represents the neighbors of the agent i, which is defined as

$$\begin{aligned} N_i(t) &= \{j \in V : (i,j) \in E(t)\} \\ &= \{j : \|x_j(t) - x_i(t)\| \leqslant r_c\}. \end{aligned} \tag{1}$$

For ease of notation, let $d_{i,j}(t) = x_j(t) - x_i(t)$, representing the vector from agents i to j. We also denote by $\langle d_{i,m_1}(t), d_{i,m_2}(t)\rangle$ the angle between $d_{i,m_1}(t)$ and $d_{i,m_2}(t)$. At time t, if agent j is the neighbor of agent i, namely $j \in N_i(t)$, then the $n \times n$ adjacency matrix $A_G(t)$ satisfying the symmetry property is used to represent graph $G(t)$. The elements in the adjacency matrix are defined as

$$A_{G:ij}(t) = \begin{cases} 1, & (i,j) \in E(t) \\ 0, & otherwise \end{cases}. \tag{2}$$

2.2. DT distributed consensus algorithm

We focus on the consensus problem in a distributed DT environment. The baseline consensus algorithm is:

$$x_i(t+1) = x_i(t) + u_i(t) \tag{3}$$

where $u_i(t)$ is the control input at time t. The construction of $u_i(t)$ has been the focus of many research efforts. We describe the baseline control input $u\prime_i(t)$ as follows, such that it does not conflict with the definition of $u_i(t)$ that we use:

$$u\prime_i(t) = \delta \sum_{j \in N_i(t)} d_{i,j}(t) \tag{4}$$

where $\delta > 0$ is the adjustment factor. Assuming that connectivity can be guaranteed for each time t, if δ is set to a suitable value, all of the agents will converge to a common equilibrium. Specifically, the MAS is said to reach a consensus if Eq. (5) is satisfied for all agents $i, j = 1, \ldots, n$ [22].

$$\lim_{t \to \infty} \|x_i(t) - x_j(t)\| = 0 \tag{5}$$

However, the hypothesis of connectivity maintenance is an ideal situation. In many cases, it is difficult to preserve the connectivity of network topology without any constraint. Thus, if the global connectivity breaks, convergence is not guaranteed.

2.3. Description of constraint set

The concept of a constraint set is used to maintain connectivity in a DT MAS [3,7]. Let $CS_i(t)$ be the constraint set of agent i at time t, which requires that the motion of agent i at time $t+1$ must lie in $CS_i(t)$, i.e. $x_i(t+1) \in CS_i(t)$. The definition of $CS_i(t)$ is related to the information of its neighbors $N_i(t)$, as shown in Eq. (6) where $B(p, r)$ denotes a closed disc with a center p and a radius r.

$$CS_i(t) = \bigcap_{j \in N_i(t)} B\left(\frac{x_i(t) + x_j(t)}{2}, \frac{r_c}{2}\right) \tag{6}$$

If the number of neighbors in the sensing range of agent i is significant, the range of $CS_i(t)$ will be small, or an empty set. Therefore, the system cannot converge effectively. To facilitate the description of the use of $CS_i(t)$, we present the following new dynamic equation:

$$x_i(t+1) = (1 - \lambda_i(t)) \cdot x_i(t) + \lambda_i(t) \cdot \hat{x}_i(t) \tag{7}$$

where $\hat{x}_i(t)$ is the target point of agent t, i.e., the position to which it wants to move. $x_i(t+1)$ belongs to the points on the line from $x_i(t)$ to $\hat{x}_i(t)$, determined by the value of $\lambda_i(t)$, i.e., $\lambda_i(t)$ is the convex combination of $x_i(t)$ and $\hat{x}_i(t)$. Because our goal is to induce a MAS to converge as soon as possible, $\lambda_i(t)$ is defined as the solution of the following convex problem:

$$\begin{aligned} &\max && \lambda_i(t) \\ &s.t. && \lambda_i(t) \leqslant 1, x_i(t+1) \in CS_i(t) \end{aligned} \tag{8}$$

Eq. (7) and (8) are also designed to solve the size of the control input $u\prime_i(t)$. Therefore, they are essentially the same as Eq. (3).

2.4. Problem description

Network topology plays a crucial role in consensus theory for MASs [20]. Limited information exchange between distant agents may hinder convergence, and thus a MAS may split into multiple clusters. To address this, connectivity and convergence are mutually constrained. Specifically, to ensure connectivity, agents are mutually constrained with their neighbors in the process of convergence evolution. Therefore, they converge slowly or ultimately cannot reach a consensus. Each agent's connectivity constraint is reduced to accelerate system convergence, resulting in the destruction of connectivity and fragmentation into multiple clusters.

Therefore, the problem is to design a distributed consensus-seeking algorithm under an undirected and time-varying communication network. This algorithm must preserve connectivity and ensure convergence. In a large-scale or a high-

density topology, a system will probably split into several clusters. Accordingly, in the evolution process of agents, an algorithm must effectively converge to a common equilibrium while preserving connectivity.

3. Methodology

In this section, we develop the SDB algorithm in Section 3.1 to simplify neighbors' information and calculate the control input of each agent. Then, we develop the DSG algorithm in Section 3.2 to reduce the effect of redundant connectivity constraints while ensuring connectivity in the evolution process. Sections 3.3 and 3.4 present connectivity and convergence analyses of our algorithms. For clarity, an outline of all proposed algorithms is shown in Table 1.

3.1. Establishment of SDB consensus algorithm

Eq. (4) shows the traditional method of calculating the control input. This method refers to the information of all neighbors and requires a unified and appropriate adjustment factor δ. For example, if agent i refers to information of k neighbors when calculating the control input $u_i(t)$, it uses the adjustment factor $\delta_i(t) = \frac{1}{k}$. In contrast, in our SDB algorithm, far fewer neighbors are selected by each agent, and each agent uses varying adjustment factors. Each agent also divides the selected neighborhood information into equal parts and obtains a compromise control input. $angle\left(\vec{\alpha}, \vec{\beta}\right)$ denotes the angle through which vector $\vec{\alpha}$ rotates counterclockwise to vector $\vec{\beta}$, as shown in Fig. 1a. $\vec{\alpha}, \vec{\beta}$ are two vectors that have the same starting point, and $angle\left(\vec{\alpha}, \vec{\beta}\right) \in [0, 2\pi)$.

As shown in Fig. 1b, to simplify the neighbor information, we divide the perception-based communication region of each agent into four sectors (i.e., quadrants) based on the x-axis and the y-axis. We denote the set of neighbors in each sector by $N_i^1(t), N_i^2(t), N_i^3(t)$, and $N_i^4(t)$, which are defined as follows:

$$\begin{aligned} N_i^Q(t) = \quad & \left\{ j \in N_i(t) | (Q-1) \cdot \tfrac{\pi}{2} \leqslant angle\left(v, d_{i,j}(t)\right) < Q \cdot \tfrac{\pi}{2} \right\} \\ & , Q \in \{1,2,3,4\} \end{aligned} \tag{9}$$

where vector $v = (1, 0)$ and Q denotes the index of a sector. We refer to $N_i^1(t)$ and $N_i^2(t), N_i^2(t)$ and $N_i^3(t), N_i^3(t)$ and $N_i^4(t)$, and $N_i^4(t)$ and $N_i^1(t)$ as adjacent neighbor sets.

To accelerate convergence, each agent selects one of the farthest neighbors from its four sectors as the reference of control input. However, because of calculation errors, there may be several coexisting farthest neighbors in each sector; this need to be further screened from many farthest neighbors to facilitate system convergence. First, the definition of the farthest neighbor set in each sector is given by

$$\widehat{N}_i^Q(t) = \left\{ k | k \in \arg\max_{j \in N_i^Q(t)} d_{i,j}(t) \right\}, \quad Q \in \{1,2,3,4\} \tag{10}$$

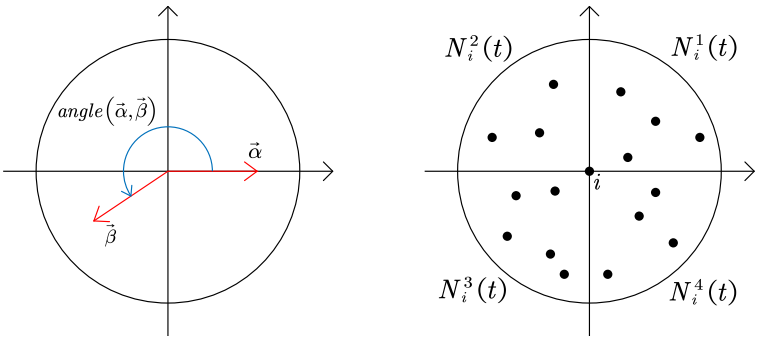
We denote by $\widehat{N}_i^{(1)}(t)$ and $\widehat{N}_i^{(2)}(t)$ two farthest adjacent neighbor sets, as shown in Fig. 2. For the convenience and simplicity of description, we use $T^i(t) \in \{0,1\}^{1\times 4}$ to represent a four-dimensional row vector whose elements are defined as

$$T_Q^i(t) = \begin{cases} 1, & |N_i^Q(t)| \neq 0 \\ 0, & otherwise \end{cases}, \quad Q \in \{1,2,3,4\}. \tag{11}$$

$T^i(t)$ is a token that records whether an agent exists in each sector of agent i. If an agent exists in the sector Q of agent i, the corresponding element $T_Q^i(t)$ is set to 1; otherwise, it is set to 0. Thus, there are four classifications of $T^i(t)$:

Table 1
Outline of our heuristic combinatorial (SDB&DSG) algorithm.

	At each time t, for each agent i with state $x_i(t)$, it	acc. to
SDB	**Step 1.** initializes the perception-based neighbor set $N_i^Q(t)$	Eq. (9)
	Step 2. selects the farthest neighbor set $\widehat{N}_i^Q(t)$ from $N_i^Q(t)$	Eq. (10)
	Step 3. evaluates and classifies the distribution of neighbors	Eq. (11) & C1-C4
	Step 4. selects the final neighbor set $\widehat{N}_i(t)$ from $\widehat{N}_i^Q(t)$	Eq. (12)
	Step 5. calculates the original control input $u_i(t)$ based on $\widehat{N}_i(t)$	Eq. (13)
DSG	**Step 6.** refines the original control input $u_i(t)$ to $\dot{u}_i(t)$	Eq. (14)
	Step 7. calculates its target point $\hat{x}_i(t)$	Eq. (15)
	Step 8. constructs the adjacency matrix of d-subgraph $A_i^d(t)$	Alg. 1
	Step 9. calculates the constraint set $\widehat{CS}_i(t)$ based on $DP_i(t)$	Eqs. (16) and (17)
	Step 10. obtains the best solution $\lambda_i^*(t)$ and evolves to $x_i(t+1)$	Eqs. (18) and (19)



(a) $angle(\vec{\alpha},\vec{\beta})$ (b) Four sectors.

Fig. 1. Definition of $angle(\vec{\alpha},\vec{\beta})$ and the sector division method.

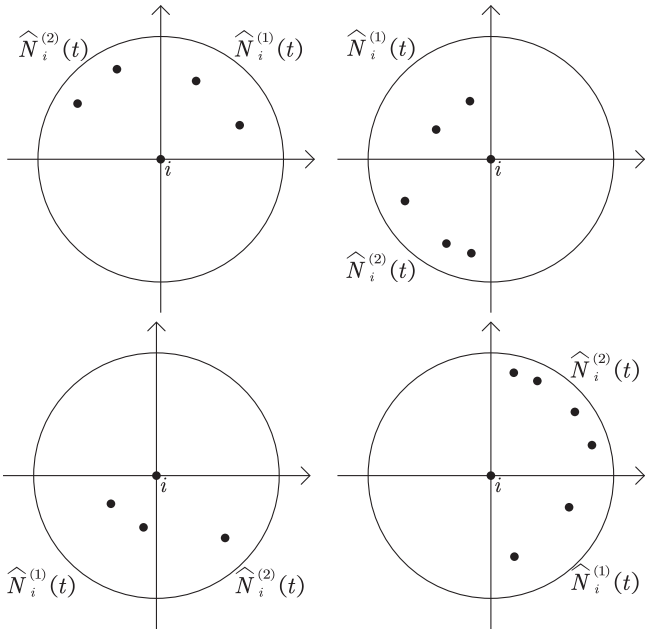


Fig. 2. Two adjacent farthest neighbor sets, i.e., $\widehat{N}_i^{(1)}(t)$ and $\widehat{N}_i^{(2)}(t)$.

C1 $\|T^i(t)\|^2 = 1$ indicates that the neighbors of agent i exist within a small range. Thus, agent i randomly selects an agent from the farthest neighbor set.

C2 $\|T^i(t)\|^2 \geqslant 3$ indicates that agent i is in a relatively central position in a local area. Thus, agent i randomly selects one of the farthest neighbors from each sector.

C3 $\|T^i(t)\|^2 = 2$ and $T^i(t)\cdot[1,-1,1,-1]^T \neq 0$ indicate that the neighbors of agent i are only in two diagonal sectors. For example, $|N_i^1(t)| = 1, |N_i^2(t)| = 0, |N_i^3(t)| = 1, |N_i^4(t)| = 0$. Thus, agent i randomly selects a neighbor from each of the two diagonal sectors.

C4 $\|T^i(t)\|^2 = 2$ and $T^i(t)\cdot[1,-1,1,-1]^T = 0$ indicate that the farthest neighbors of agent i are distributed in two adjacent farthest neighbor sets, as shown in Fig. 2. Here, we adapt and generalize the idea of heterophily in [19]. Agent i selects $m_1 \in \widehat{N}_i^{(1)}(t)$ and $m_2 \in \widehat{N}_i^{(2)}(t)$, where m_1 and m_2 need to maximize $\langle d_{i,m_1}(t), d_{i,m_2}(t)\rangle$, which is the angle between $d_{i,m_1}(t)$ and $d_{i,m_2}(t)$, as shown in Fig. 3.

The four classifications above can be summarized into the following two methods, as shown in Eq. (12): The random selection method is used to reduce the computational complexity of most agents, and the selection method is used to optimize the constraint set so that the system can converge effectively. The set is defined as

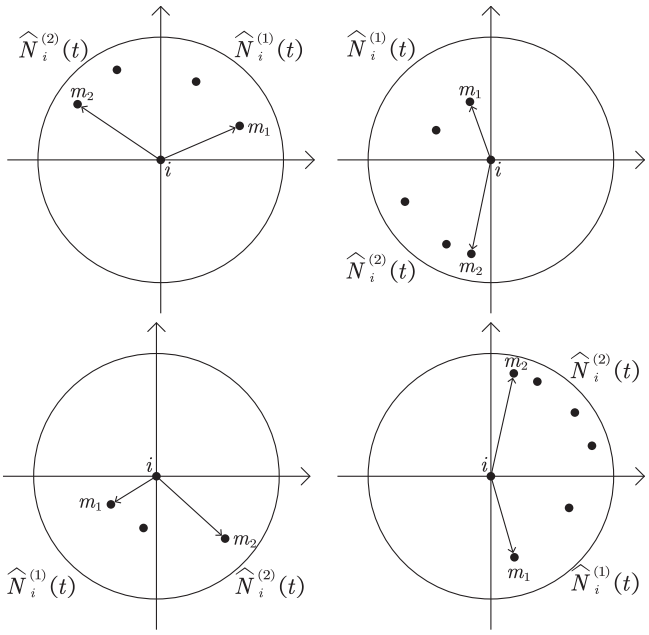


Fig. 3. Selected neighbors m_1 and m_2 from two adjacent farthest neighbor sets.

$$\widehat{N}_i(t) = \begin{cases} \bigcup_{Q=1}^{4} \left\{ rand\left(\widehat{N}_i^Q(t)\right)\right\}, & \|T^i(t)\|^2 \neq 2 or T^i(t)\cdot e \neq 0 \\ \left\{ m_1, m_2 | \arg\max_{m_1 \in \widehat{N}_i^{(1)}(t), m_2 \in \widehat{N}_i^{(2)}(t)} \left\langle d_{i,m_1}(t), d_{i,m_2}(t) \right\rangle \right\}, & otherwise \end{cases} \tag{12}$$

where $e = [1, -1, 1, -1]^T$ and the function $rand(s)$ randomly selects an element in set s.

If each agent selects the farthest neighbor to construct the control input, before the system topology reaches full connectivity, system convergence can be accelerated. However, it is difficult for the system to converge to a common equilibrium after achieving full connectivity. As shown in Fig. 4, two groups of agents select one of the farthest agents in the other group as their control input reference. Thus, the two groups of agents constantly exchange positions during the evolution process and are unable to converge to one point. Therefore, agents must switch their dynamics to ensure system convergence. We equip agents with the ability to know the switching time of the control input calculation strategy with minimal computational cost. Therefore, the control input of each agent is designed as follows:

$$u_i(t) = \begin{cases} \frac{1}{|\widehat{N}_i(t)|} \cdot \sum_{j \in \widehat{N}_i(t)} d_{ij}(t), & |N_i(t)| < n-1 \\ \frac{1}{|N_i(t)|} \cdot \sum_{j \in N_i(t)} d_{ij}(t), & otherwise. \end{cases} \tag{13}$$

Specifically, each agent selects a maximum of four neighbors to calculate the control input before the number of neighbors reaches $n-1$. Therefore, the control input of Eq. (13) can greatly reduce the number of neighbors compared with the baseline control input of Eq. (4).

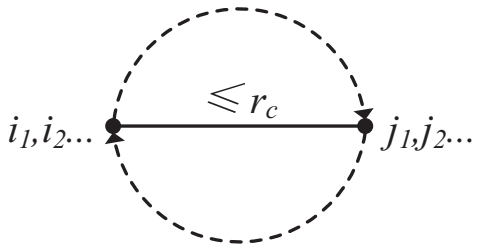


Fig. 4. Network topology that cannot converge to one point.

Remark 1. Each agent must perform a simple evaluation of locally perceived neighbors before calculating the control input. Once the number of agent i's neighbors reaches $n-1$, agent i is fully connected with all other local agents in its perception range and thus its control input calculation strategy is based on $\widehat{N}_i(t)$; otherwise, it is based on $N_i(t)$. This implies that the switching time point of agent i to calculate the control input in Eq. (13) is determined by its locally perceived neighbors. Therefore, the switching time point is not synchronized for each agent.

3.2. Establishment of DSG connectivity maintenance algorithm

This section focuses on the design of the DSG connectivity maintenance algorithm. First, we refine the original control input derived from Eq. (13) to

$$\hat{u}_i(t) = \min\left\{\frac{r_c - d}{2\|u_i(t)\|}, 1\right\} \cdot u_i(t). \tag{14}$$

where the parameter $d \in [0, r_c)$ is static during the process of convergence evolution. $\hat{u}_i(t)$ is in the same direction as $u_i(t)$ of Eq. (13), except that the length of $\hat{u}_i(t)$ is limited to $(0, \frac{r_c-d}{2}]$. Then, the definition of the target point $\hat{x}_i(t)$ of agent i is

$$\hat{x}_i(t) = x_i(t) + \hat{u}_i(t) \tag{15}$$

We use matrix $A_i^d(t) \in \{0,1\}^{n\times n}$ to represent the adjacency matrix of the d-subgraph of agent i. For convenience, we define this adjacency matrix as an $n \times n$ matrix. Thus, agent i only needs to establish an $|N_i(t)| \times |N_i(t)|$ matrix of corresponding dimensions, according to the number of its perceived neighbors $|N_i(t)|$. Each agent's control input has the upper bound $\frac{r_c-d}{2}$, and thus if the distance between any two neighbors of agent i is not greater than d at time t, these neighbors will not disconnect at time $t+1$. We present the construction of $A_i^d(t)$ in Algorithm 1, where $A_{i:ab}^d(t)$ denotes the element at the a-th row and the b-th column in $A_i^d(t)$.

Algorithm 1: Construction of the adjacency matrix of the d-subgraph $A_i^d(t)$.

Input: $N_i(t)$
Output: $A_i^d(t)$
for $a \in N_i(t)$ **do**
  for $b \in N_i(t)$ and $b \neq a$ **do**
    if $\|x_a(t) - x_b(t)\| \leqslant d$ **do**
      Update $A_{i:ab}^d(t) = A_{i:ba}^d(t) = 1$.
    end if
  end for
end for
for $j \in N_i(t)$ **do**
  if $\|d_{ij}(t)\| \leqslant \frac{r_c+d}{2} - \|\hat{u}_i(t)\|$
    Update $A_{i:ij}^d(t) = A_{i:ji}^d(t) = 1$.
  end if
end for

We need $A_i^d(t)$ to be connected to reduce the connectivity constraints imposed on agent i. However, in most cases, $A_i^d(t)$ is not connected and has multiple connected components. Suppose that $A_i^d(t)$ has $q \in \mathbb{N}^+$ connected components, where $q \in [1, \ldots, n]$. If $A_i^d(t)$ is connected, then $q = 1$. Let the connected component including agent i be $cp_i^0(t)$ and the remainder of the connected components be $cp_i^1(t)$, …, $cp_i^{q-1}(t)$. In particular, $cp_i^1(t)$, …, $cp_i^{q-1}(t)$ are connected to each other in $A_G(t)$, and $\vee_{p=1}^q cp_i^p(t)$ is a disconnected subgraph of $G(t)$. As mentioned in Section 3.2, if agent i can ensure a connection with an agent in each connected component, the local connectivity of agent i is guaranteed. Thus, the global connectivity is also guaranteed. Let the point requiring agent i to maintain connectivity be defined as

$$DP_i(t) = \bigcup_{l=1}^{q-1}\left\{\arg\min_{j\in cp_i^l(t)} \|d_{ij}(t)\|\right\}. \tag{16}$$

As shown in Fig. 5, agent i adds to $DP_i(t)$ the neighbor closest to it in each connected component. Calculation errors may lead to multiple nearest neighbors in a connected component, and all of these agents must be added to $DP_i(t)$. Subsequently, the construction of the constraint set is based on the information of neighbors in $DP_i(t)$. An additional condition is required to guarantee connections between agents and their neighbors. For example, if agent i wants to maintain a direct connection

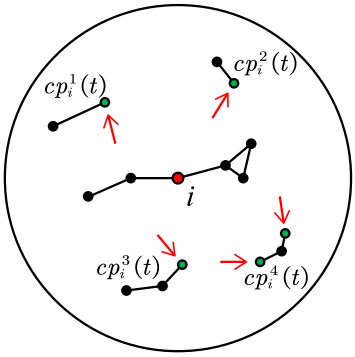


Fig. 5. Illustration of $DP_i(t)$: The agent indicated by the red arrow in each connected component closest to agent i is added to $DP_i(t)$.

with its neighbor j, it also needs agent j to maintain a direct connection with itself, i.e., bilateral connectivity is required, as unilateral connection requests do not guarantee the connectivity of the system. We define the constraint set $\widehat{CS}_i(t)$ based on $DP_i(t)$ as follows:

$$\widehat{CS}_i(t) = \bigcap_{j \in DP_i(t)} B\left(\frac{x_i(t) + x_j(t)}{2}, \frac{r_c}{2}\right) \tag{17}$$

$B(C_{ij}, \frac{r_c}{2})$ denotes a closed disc with a radius of $\frac{r_c}{2}$ and a center C_{ij} that is the midpoint of the distance between each neighbor in $DP_i(t)$ and agent i. Thus, $x_i(t) \in \widehat{CS}_i(t)$ can be obtained. The target point of agent i at the next time should be constrained in $\widehat{CS}_i(t)$ to preserve connections between agent i and its selected neighbors in $DP_i(t)$. Subsequently, each agent solves the following convex problem to obtain the best solution $\lambda_i^*(t)$:

$$\begin{aligned} &\max && \lambda_i(t) \\ &s.t. && \lambda_i(t) \leqslant 1, x_i(t+1) \in \widehat{CS}_i(t) \end{aligned} \tag{18}$$

where $x_i(t+1)$ is defined in Eq. (7).

Afterward, each agent can evolve to the convex combination between current state $x_i(t)$ and the target state $\hat{x}_i(t)$ with the coefficient $\lambda_i^*(t)$ according to Eq. (19).

$$x_i(t+1) = \left(1 - \lambda_i^*(t)\right) \cdot x_i(t) + \lambda_i^*(t) \cdot \hat{x}_i(t) \tag{19}$$

3.3. Connectivity analysis

Theorem 1. *If each agent calculates the control input according to Eq. (14), then* $\vee_{i=1}^n A_i^d(t) \wedge A(t+1) = \vee_{i=1}^n A_i^d(t)$.

Proof of Theorem 1

1. Suppose that $A_{i:ij}^d(t) = 1$. The value of the control input of agent i is $\|\hat{u}_i(t)\|$and the control input of agent j satisfies $\|\hat{u}_j(t)\| \leqslant \frac{r_c - d}{2}$, so $x_i(t+1) \in B(x_i(t), \frac{r_c-d}{2})$ and $x_j(t+1) \in B(x_j(t), \frac{r_c-d}{2})$. Furthermore, the distance between agent i and agent j is not greater than $\frac{r_c+d}{2} - \|\hat{u}_i(t)\|$. Geometrically, the distance between point a and point b is not greater than r_c for any $a \in B(x_i(t), \frac{r_c-d}{2})$ and $b \in B(x_j(t), \frac{r_c-d}{2})$. Therefore, agent i and agent j are not disconnected at time $t+1$, i.e., $A_{i:ij}^d(t) \wedge A_{G:ij}(t+1) = 1$.
2. Suppose that $A_{i:ab}^d(t) = 1$. The control input of agent a and b satisfy $\|u_a(t)\|, \|u_b(t)\| \leqslant \frac{r_c-d}{2}$, and the distance between agent a and agent b is not greater than d. Then, m_1 and m_2 are not disconnected at time $t+1$, i.e., $A_{i:ab}^d(t) \wedge A_{G:ab}(t+1) = 1$.

In summary, $\vee_{i=1}^n A_i^d(t) \wedge A_G(t+1) = \vee_{i=1}^n A_i^d(t)$.

Remark 2. Suppose that the adjacency matrix of a given topology can be expressed as $M \in \{0,1\}^{n \times n}$. If $M \wedge A(t) = M$, then M is a subgraph of $A(t)$.

Theorem 1 shows that the element in matrix $\vee_{i=1}^n A_i^d(t)$ with a value of 1 does not change in $A_G(t+1)$, i.e., the edge in $\vee_{i=1}^n A_i^d(t)$ is maintained at time $t+1$. For example, as described in Algorithm 1, if agent i remains connected with agent a at time $t+1$, it is guaranteed that no matter how agent b moves there exists at least one indirect connection between agents

i and b. In addition, we want $A_i^d(t)$ to be highly connected; in this case, agent i does not need to consider the constraints of its neighbors, because the connectivity of $A_i^d(t)$ can be guaranteed at $t+1$. Therefore, we relax the condition so that more edges can be added to $A_i^d(t)$, as shown in steps 1–4 of Algorithm 1.

Remark 3. Suppose that agents i, j, and k are neighbors. Because the value of the control input of each agent is different, the judging condition in step 6 of Algorithm 1 will be different. Therefore, the following situation occurs: $A_{i:ik}^d(t)=1$ but $A_{j:ik}^d(t)=0$, i.e., the information is asymmetric. However, this does not affect the final status. If $\vee_{j=1}^n A_{j:ik}^d(t)=1$, agents i and k remain neighbors at time $t+1$. It is sufficient that each agent focuses on its own information and does not require additional communication to reach an agreement with the information of its neighbors.

Theorem 2. *Suppose that $A_{ij}(t)=1$. If $x_i(t)\in DP_j(t)$ and $x_j(t)\in DP_i(t)$, then $A_{ij}(t+1)=1$.*

Proof of Theorem 2

Because $i\in DP_j(t)$ and $j\in DP_i(t)$, $\widehat{CS}_i(t)\subseteq B\left(\frac{x_i(t)+x_j(t)}{2},\frac{r_c}{2}\right)$, $\widehat{CS}_j(t)\subseteq B\left(\frac{x_j(t)+x_i(t)}{2},\frac{r_c}{2}\right)$. According to the definition of the constraint set, $x_i(t+1)\in CS_i(t)$ and $x_j(t+1)\in CS_j(t)$, so $x_i(t+1)\in B\left(\frac{x_i(t)+x_j(t)}{2},\frac{r_c}{2}\right)$ and $x_j(t+1)\in B\left(\frac{x_j(t)+x_i(t)}{2},\frac{r_c}{2}\right)$. The distance between any two agents in $B\left(\frac{x_i(t)+x_j(t)}{2},\frac{r_c}{2}\right)$ is not greater than r_c, so agents i and j remain connected at time $t+1$, i.e., $A_{ij}(t+1)=1$.

Theorem 2 explains the previous statement that unilateral connection requests do not guarantee global connectivity. As shown in Fig. 6, the solid line indicates the edge in $\vee_{i=1}^n A_i^d(t)$, and the dotted line indicates the edge that exists in $A(t)$ but does not exist in $\vee_{i=1}^n A_i^d(t)$. The connection of solid lines can be maintained at the next time $t+1$ based on the developed algorithm. In addition, the system in Fig. 6 has several connected components, between each of which there must be at least one shortest edge (dotted line). Two agents connected by the shortest edge must select each other as elements in $DP_i(t)$. We summarize the connectivity analysis in the following theorem:

Theorem 3. *If $A_G(0)$ is connected, $A_G(t)$ is always connected based on DSG.*

Proof of Theorem 3

Suppose that $A(t-1)$ remains connected. Let the connected component in $\vee_{i=1}^n A_i^d(t-1)$ be $cp^p(t-1)$, where the number of connected components is $q\in\{1,\dots,n\}$, so $p\in\{1,\dots,q\}$ and $\vee_{p=1}^q cp^p(t-1)=\vee_{i=1}^n A_i^d(t-1)$. Suppose that there are two different connected components, $cp^{c_1}(t-1)$ and $cp^{c_2}(t-1)$, and that there is one shortest edge (a,b) connecting these two parts. According to Eq. (16), agent a and agent b connected by edge (a,b) must satisfy $a\in DP_b(t-1)$ and $b\in DP_a(t-1)$. Thus, based on Theorem 2, $A_{G:ab}(t)=1$. Let $C=A_{G:ab}(t-1)\vee cp^{c_1}(t-1)\vee cp^{c_2}(t-1)$, thus $C\wedge A_G(t)=C$. Because there must be at least one shortest edge between each connected component, the connected components are connected in $A_G(t)$.

In summary, $A_G(t)$ remains connected for all time $t\geqslant 0$.

Remark 4. If $d=0$, then the algorithm is degenerated into $r-disk$ (RD) [3] where r means r_c in this paper. Therefore, RD is a special case of DSG.

3.4. Convergence analysis

Due to the introduction of a constraint set, agents are sometimes mutually constrained, which means that the control input of the agent does not intersect with the constraint set, i.e., $\lambda(t)=0$ in Eq. (8). In this case, the agent is stationary and fails to reach a consensus. Under the assumption that connectivity is guaranteed, our algorithm can solve the consensus problem readily. Thus, we use knowledge of geometry to prove that our SDB&DSG algorithm can enable agents to reach a consensus. The convex hull of a point set P refers to the smallest convex polygon that contains the points in P on or within its boundary. Let $\Omega(t)$ be the convex hull of $X(t)$. According to the algorithm for constructing a convex hull, if the coordinate value of agent i is the maximum or minimum value of one dimension, agent i must be on the boundary of the convex hull. We denote these driving points $P_\Omega(t)$, as shown in Fig. 7.

The proof of convergence involves two steps. In the first step, we prove that the area of $\Omega(t)$ does not increase. In the second step, we prove that the area of $\Omega(t)$ reduces in each iteration. These steps account for the convergence of the entire system.

Lemma 1. *Suppose that the time-varying network topology of an MAS is connected at each time t. If δ in Eq. (4) satisfies $0<\delta\leqslant\frac{2}{n}$, then a consensus is asymptotically reached.*

Proof of Lemma 1

We define the Lyapunov function as

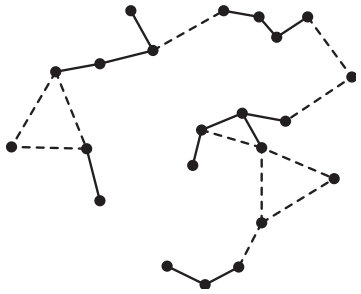


Fig. 6. The solid line length is not greater than d, while the dotted line length is greater than d but less than r_c. The dotted line represents the shortest connections between two connected components.

$$V(X(t)) = X^T(t)X(t) \tag{20}$$

which is a positive definite function. We use the Laplacian matrix $L(t)$ to denote the network topology. Thus, the matrix representation of Eqs. (3) and (4) is defined as

$$X(t+1) = X(t) + \delta L(t)X(t) = (I + \delta L(t))X(t) \tag{21}$$

Consequently, we have

$$\begin{aligned}
&\Delta V(X(t)) = V(X(t+1)) - V(X(t)) \\
&= X^T(t+1)\cdot X(t+1) - X^T(t)\cdot X(t) \\
&= X^T(t)(I+\delta L(t))^T(I+\delta L(t))X(t) - X^T(t)X(t) \\
&= X^T(t)\Big(2\delta L(t) + \delta^2 L^2(t)\Big)X(t) \\
&= 2\delta\cdot X^T(t)\Big(L(t) + \tfrac{\delta}{2}L^2(t)\Big)X(t) \\
&= 2\delta\cdot X^T(t)W(t)X(t)
\end{aligned} \tag{22}$$

where $W(t) = L(t) + \frac{\delta}{2}L^2(t)$. According to the Lyapunov stability theory, if $W(t)$ is a negative semidefinite matrix, then the MAS is asymptotically stable in the Lyapunov sense. The diagonal elements $[l_{ii}(t)]$ in $L^2(t)$ are $n_i^2(t) + n_i(t)$ and the off-diagonal elements in $[l_{ij}(t)]$ in the same row satisfy

$$\sum_{j\neq i} l_{ij} = -\big(n_i^2(t) + n_i(t)\big) \tag{23}$$

where $n_i(t)$ is the number of neighbors of agent i at time t. The elements in $W(t)$ satisfy

$$w_{ii}(t) = \frac{\delta}{2}\big(n_i^2(t) + n_i(t)\big) - n_i(t) \tag{24}$$

$$\sum_{j\neq i} w_{ij}(t) = n_i(t) - \frac{\delta}{2}\big(n_i^2(t) + n_i(t)\big) \tag{25}$$

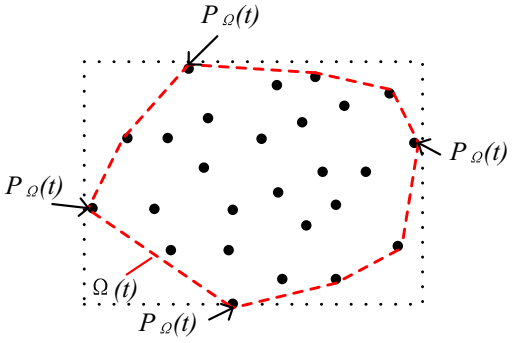


Fig. 7. Convex hull $\Omega(t)$ and driving points $P_\Omega(t)$ of a point set.

Suppose that matrix $W(t)$ is a negative semidefinite matrix, such that all eigenvalues of $W(t)$ are no greater than 0 [9]. Then, according to the Gershgorin circle theorem [9], all eigenvalues of matrix $W(t)$ are within the union of n discs, that is:

$$|\lambda - w_{ii}(t)| \leqslant \sum_{j \neq i} w_{ij} \tag{26}$$

To ensure that the centers of all discs are located left of the complex plane, the centers $w_{ii}(t)$ of all discs should satisfy $w_{ii}(t) \leqslant 0$, i.e,

$$\begin{aligned} w_{ii}(t) = \tfrac{\delta}{2}\left(n_i^2(t) + n_i(t)\right) - n_i(t) \leqslant 0 \\ \Rightarrow \delta \leqslant \tfrac{2}{n_i(t)+1} \end{aligned} \tag{27}$$

If an agent selects all of its perceived neighbors for evolution, then $max(n_i(t)) = n - 1$ and Eq. (27) can be converted to $\delta \leqslant \frac{2}{n}$.

Theorem 4. *Let Ω be a convex polygon area with the set $X = \{x_0, \ldots x_n\} \subseteq \Omega, n \in \mathbb{N}^+$. Also, let $\chi_n = \sum_{i=1}^{n} \frac{x_i - x_0}{n}$, then $x_0 + \chi_n \in \Omega$.*

Proof of Theorem 4

1. Base case: If $n = 1$, then $x_0 + \chi_1 = x_1 \in \Omega$. If $n = 2, x_0 + \chi_2 = \frac{x_1 + x_2}{2} \in \Omega$.
   Therefore, the theorem holds when $n = 1$.
2. Inductive hypothesis: Suppose the theorem holds for an arbitrary k, i.e., $x_0 + \chi_k \in \Omega$.
3. Inductive step: Let $n = k + 1$. Then the left side is

$$\begin{aligned} \chi_{k+1} &= \sum_{i=1}^{k+1} \tfrac{x_i - x_0}{k+1} \\ &= \sum_{i=1}^{k} \tfrac{x_i - x_0}{k+1} + \tfrac{x_{k+1} - x_0}{k+1} \end{aligned} \tag{28}$$

By our inductive hypothesis, we know $p1 = x_0 + \frac{k}{k+1} \cdot \chi_k \in \Omega$. Then, because $p2 = x_0 + \frac{x_{k+1} - x_0}{k+1} \in \Omega, x_0 + \chi_{k+1}$ is a point on the line segment between $p1$ and $p2, x_0 + \chi_{k+1} \in \Omega$.

The principle of mathematical induction means that the theorem holds for all $n \in \mathbb{N}$.

Theorem 4 shows that the convex hull area of the entire system does not increase, i.e., $\Omega(t+1) \subseteq \Omega(t)$. Each agent moves only within the existing convex hull range. Next, we prove that the convex hull area reduces in each iteration. Similarly, this proof is divided into two steps. In the first step, we prove that $\forall i \in P_\Omega(t)$, its control input is not zero and the constraint set is not empty. In the second step, we prove that agent i moves within the convex hull range, i.e., $\lambda_i(t) \neq 0$.

Theorem 5. *If $\forall i \in P_\Omega(t)$, then $\|u_i(t)\| \neq 0$ and $CS_i(N_i(t)) \setminus \{x_i(t)\} \neq \varnothing$.*

Proof of Theorem 5

Let $< \alpha, \beta >$ indicates the angle between the vector $\alpha$and $\beta, < \alpha, \beta > \in [0, \pi]$. From the definition of the convex hull, the inner angle of $\Omega(t)$ is less than π. Therefore, if $\forall i \in P_{CH}(t)$, then $< d_{ij}(t), d_{ik}(t) >_{j,k \in N_i(t)} \in [0, \pi)$.

1. Because agents in $N_i(t)$ are on the same side, $\|u_i(t)\| \neq 0$ according to Eq. (13).
2. Suppose that there are two agents $m, n \in DP_i(t)$ that render $B\left(\frac{x_i(t)+x_m(t)}{2}, \frac{r_c}{2}\right)$ and $B\left(\frac{x_i(t)+x_n(t)}{2}, \frac{r_c}{2}\right)$ tangential, i.e., $B\left(\frac{x_i(t)+x_m(t)}{2}, \frac{r_c}{2}\right) \bigcup B\left(\frac{x_i(t)+x_n(t)}{2}, \frac{r_c}{2}\right) \setminus \{x_i(t)\} = \varnothing$. So m and n must satisfy $\|d_{im}(t)\| = r_c, \|d_{in}(t)\| = r_c$ and $\langle d_{im}(t), d_{in}(t) \rangle = \pi$ which are contradictory to $\left\langle d_{ij}(t), d_{ik}(t) \right\rangle_{j,k \in N_i(t)} \in [0, \pi)$. Therefore, the original theorem $CS_i(t) \setminus \{x_i(t)\} \neq \varnothing$ is proved.

According to Theorem 5, there must be some agents in the system with a sufficiently large range constraint set and non-zero control input. These agents may have a relationship similar to that shown in Fig. 8, with $\lambda_i(t) = 0$, indicating that they are stationary. For agent $i \in P_\Omega(t)$, if $\forall j \in DP_i(t)$ satifies $\|d_{i,j}(t)\| < r_c$, then $x_i(t)$ must be in $CS_i(t)$ instead of being on the boundary of $CS_i(t)$ according to the definition of the constraint set in Eq. (17). If $\exists j \in DP_i(t)$ satisfies $\|d_{i,j}(t)\| = r_c$, then $x_i(t)$ must be on the boundary of $CS_i(t)$. As shown in Fig. 9, if $\exists j \in DP_i(t), \|d_{i,j}(t)\| = r_c$ satisfies $\left\langle \hat{u}_i(t), d_{i,j}(t) \right\rangle \geqslant \frac{\pi}{2}$, then $\lambda_i(t) = 0$. Thus, if $\forall j \in DP_i(t), \|d_{i,j}(t)\| = r_c$ satisfies $\left\langle \hat{u}_i(t), d_{i,j}(t) \right\rangle \geqslant \frac{\pi}{2}$, then $\lambda_i(t) \neq 0$. Consequently, because $DP_i(t) \subseteq N_i(t)$, the problem we must solve can be converted into the following form: If $\forall j \in N_i(t), \|d_{i,j}(t)\| = r_c$, then $\left\langle \hat{u}_i(t), d_{i,j}(t) \right\rangle < \frac{\pi}{2}$.

Theorem 6. *If $n > 2$, the system can converge to one common equilibrium value based on our algorithm SDB&DSG.*

Proof of Theorem 6

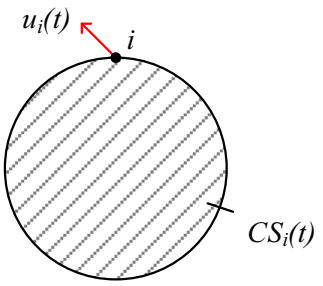


Fig. 8. Stationary case of $\lambda_i(t) = 0$.

Suppose that $i \in P_\Omega(t)$ and the state value of agent i is the maximum on the y-axis. Then, this proof is conducted by case analysis. There are three cases: $T^i(t) = [0,0,1,1], T^i(t) = [0,0,0,1]$ and $T^i(t) = [0,0,1,0]$. Because the control input switches throughout the process of convergence evolution, the proof is divided into two phases, as follows.

1. Before the system is fully connected, it must satisfy $|N_i(t)| < n-1$. According to Eq. (13), agent i selects neighbors based on Eq. (12). Therefore, two situations exist:
   (a) $\forall j \in DP_i(t)$ satisfies $\|d_{i,j}(t)\| < r_c$. According to the definition of $CS_i(t)$ in Eq. (17), agent i satisfies the following property: $x_i(t)$ is in $CS_i(t)$ and not on the boundary of $CS_i(t)$, i.e., $\lambda_i(t) \neq 0$. Thus, agent i evolves to the interior of the convex hull.
   (b) $\exists j \in DP_i(t)$ satisfies $\|d_{i,j}(t)\| = r_c$. In the case of $T^i(t) = [0,0,1,0]$ or $T^i(t) = [0,0,0,1]$, as shown in Fig. 10(a), assuming that agent i randomly selects agent m_1, the direction of the control input $\hat{u}_i(t)$ is equal to $d_{i,m_1}(t)$, and $\|d_{i,m_1}(t)\| = r_c$. Thus, we obtain $\forall k \in N_i(t), \|d_{i,k}(t)\| = r_c$ and $\langle d_{i,m_1}(t), d_{i,k}(t)\rangle$ is less than $\frac{\pi}{2}$. Thus, $\langle d_{i,m_1}(t), d_{i,j}(t)\rangle$ is less than $\frac{\pi}{2}$. In the case of $T^i(t) = [0,0,1,1]$ as shown in Fig. 10(b), if agent i selects agents m_1 and m_2, then at least one of $\|d_{i,m_1}(t)\|$and $\|d_{i,m_2}(t)\|$is equal to r_c. We observe that angle $\langle d_{i,m_1}(t), d_{i,m_2}(t)\rangle$ is less than π. Let $u_i(t) = \frac{d_{i,j1}(t)+d_{i,j2}(t)}{2}$. Then, $\forall j \in N_i(t), \|d_{i,j}(t)\| = r_c$ must satisfy $\langle u_i(t), d_{i,j}(t)\rangle < \frac{\pi}{2}$.

   Therefore, the system can be completely connected.
2. When the system reaches full connectivity, the distance between each agent is not greater than r_c, so all agents lie in an equilateral triangle of side length r_c. Thus, $\forall j \in N_i(t), \langle u_i(t), d_{i,j}(t)\rangle$ is less than $\frac{\pi}{3}$. Because $n > 2, \frac{1}{n-1} < \frac{2}{n}$, then a consensus is asymptotically reached, according to Lemma 1.

When the system is not fully connected, $i \in P_\Omega(t)$ and $|N_i(t)| = n-1$ are allowed in the proof of Theorem 6 but agent $j \in P_\Omega(t)$ must satisfy $|N_j(t)| < n-1$. This finding does not affect the soundness of our proof because agent j compresses the area of the convex hull in a limited iteration until the entire system reaches full connectivity.

4. Simulation and analysis

In this section, we conduct a series of simulations using the four algorithms, to.

1. validate the effectiveness of their connectivity and convergence;
2. optimize their adjustable parameters of proposed algorithms; and

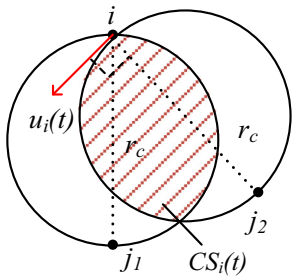
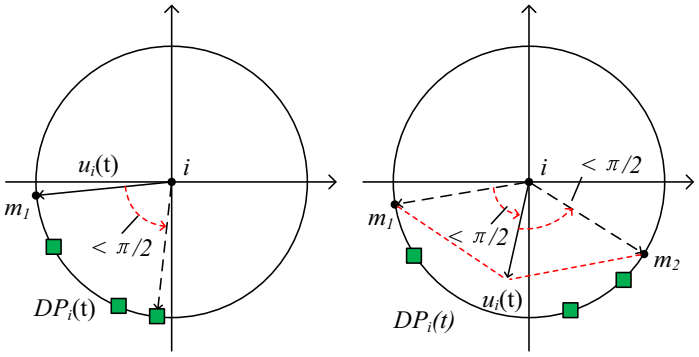


Fig. 9. If agent j_2 satisfies $< \hat{u}_i(t), d_{i,j_2}(t) >$ is equal $\frac{\pi}{2}$, then $\lambda_i(t) = 0$.



(a) $T^i(t) = \{0, 0, 1, 0\}$ (b) $T^i(t) = \{0, 0, 1, 1\}$

Fig. 10. For $\forall j \in DP_i(t)$ and $\|d_{i,j}(t)\| = r_c$ (e.g. green squares) satisfies that $\langle \hat{u}_i(t), d_{i,j}(t) \rangle$ is less than $\frac{\pi}{2}$.

3. compare the efficiency of their convergence.

4.1. Validation of connectivity and convergence

In this section, a random topology network comprising 1,500 agents is designed to illustrate the effect of the baseline algorithm of Eq. (4) and our heuristic combinatorial SDB&DSG algorithm. We let $\delta = \frac{1}{n}$ and $d = 0.1$.

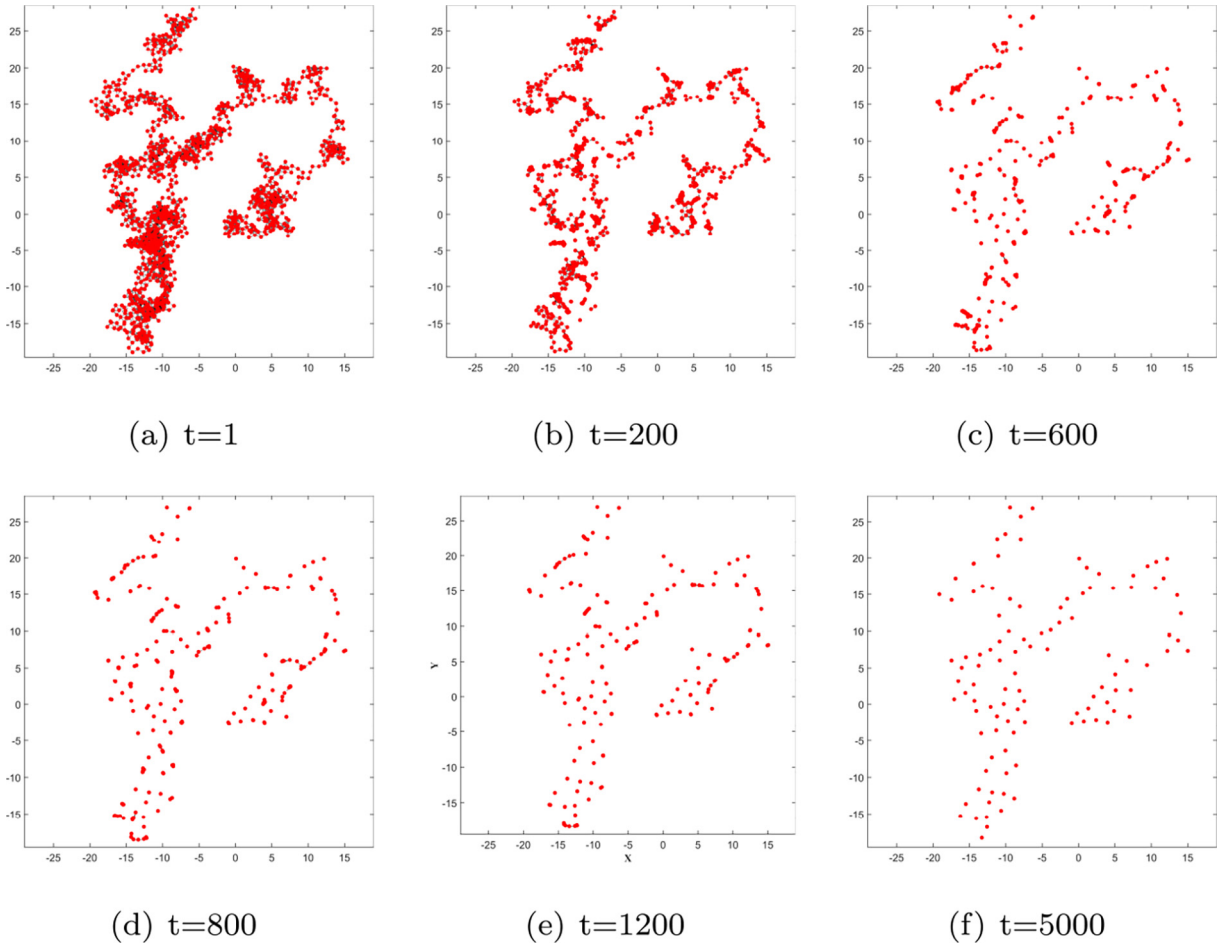


Fig. 11. Multi-agent convergence process of the baseline algorithm (Eq. (4).

As shown in Fig. 11, the system iterates more than 5,000 times from the initial state to the stable state before splitting into 113 clusters. However, under the same topology but based on our SDB&DSG algorithm, the system eventually converges to the same point at $t = 150$, as shown in Fig. 12. Interestingly, agents first converge to the local geometric center and then to the global geometric center. Therefore, DSG can effectively maintain connectivity, and SDB can ensure stable convergence of a MAS.

4.2. Optimization and analysis of adjustable parameters

In this section, we examine how the adjustable parameters affect the results. We can modify two adjustable parameters in our developed algorithms, i.e., d and the number of sectors s. In Section 3, we provide the proof for the four sectors. However, the empirical results show that these four sectors are unsuitable. For a MAS with a smaller d, the larger $\|\hat{u}_i(t)\|$ is, the greater $|DP_i(t)|$ will be, thus decreasing the range of $\widehat{CS}_i(t)$ and causing slow convergence. The converse is also true. Therefore, it is critical to determine appropriate values for adjustable parameters.

As shown in Fig. 13, we design six initially connected topologies with varying densities ρ. The topologies have 400 (D1-1), 800 (D1-2), 1,200 (D1-3), 1,600 (D1–4), 2,000 (D1-5) and 2,400 (D1-6) agents, randomly distributed in a $10r_c \times 10r_c$ square area under a uniform distribution [35,39]. Table 2 displays statistics for the six topologies based on topology type (i.e., #Top.), number of agents (i.e., #Agents), range of topological network (i.e., #Range), and the density of MAS (i.e., #Density). Fig. 14 and Fig. 15 show the simulation results, which illustrate the number of iterations (i.e., #Times) for various values of d and s.

First, we conduct experiments with varying $d \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ and $s = 4$, as shown in Fig. 14. The SDB&DSG algorithm degrades into the classic RD algorithm when $d = 0r_c$. For high-density topologies, the range of the constraint set of each agent is large, resulting in slow system convergence. When $d > 0.5r_c$, the range of the constraint set and the corresponding control input $u(t)$ decrease; therefore, the convergence performance is unsatisfactory. However,

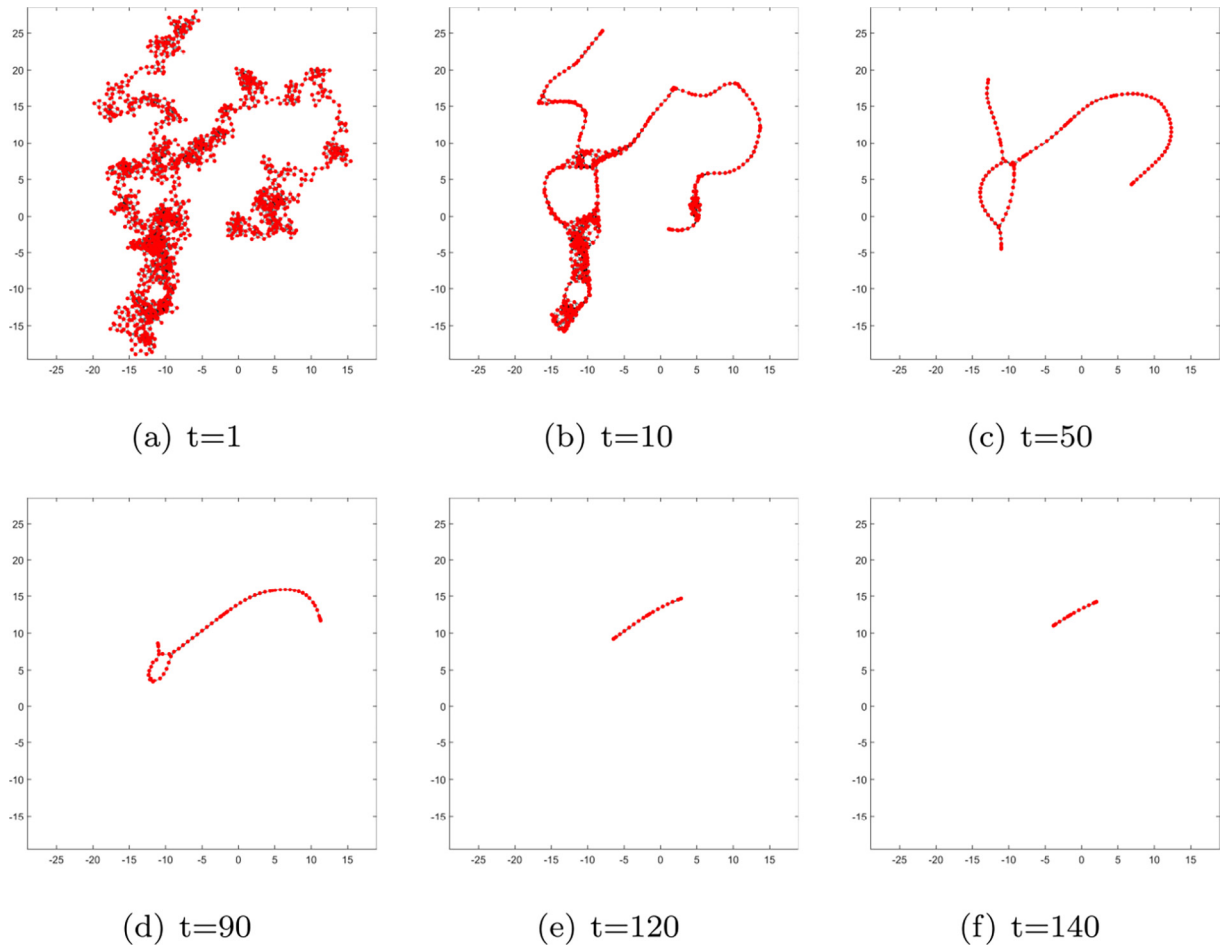


Fig. 12. Multi-agent convergence process of our SDB&DSG algorithm.

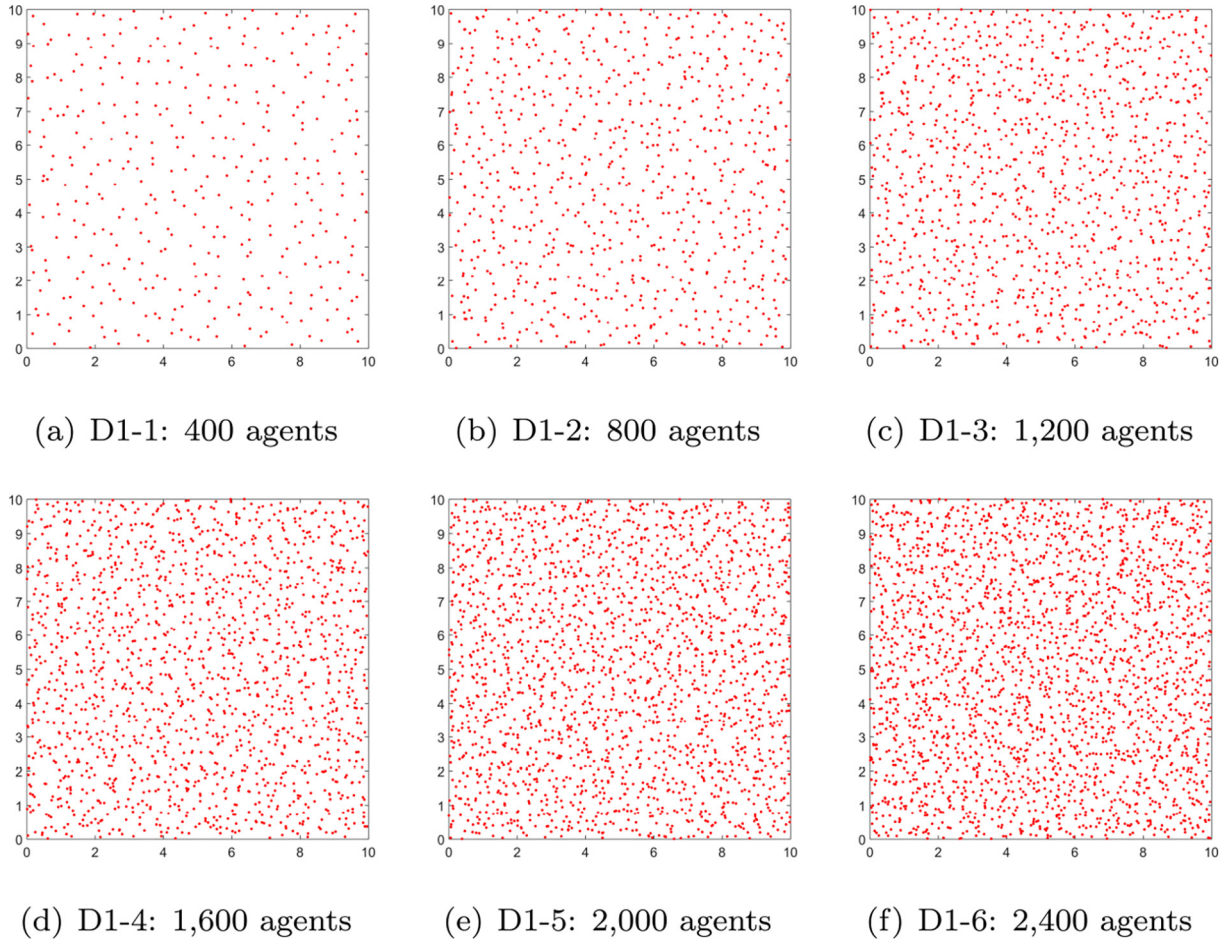


Fig. 13. Initial topological networks of various densities.

when d is concentrated in the range from $0.2r_c$ to $0.4r_c$, the convergence performance improves and is more stable. In the case of varying densities, when $d = 0$, the higher the density, the worse is the convergence performance. When $d > 0$, under the same range and varying densities, the convergence speed of the system gradually stabilizes, which also shows that the DSG algorithm can reduce the influence of density and improve the convergence performance of the system.

Second, we conduct experiments with varying $s \in \{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16\}$ and $d = 0.3$, as shown in Fig. 15. Under the low-density topology (i.e., D1-1 and D1-2) with varying s, the convergence time of the system is relatively stable, and $s \in [5, 9]$ performs best. However, the convergence time significantly increases as the density increases, and $s \in [4, 6]$ performs best. This finding indicates that calculating many neighbors may impair the convergence performance of the system if the density ρ and s increase. Therefore, selecting the most representative neighbors with a suitable s in the communication region both reduces the computational load of the agent and accelerates the system convergence, especially for high-density topologies.

Table 2
Statistics of six topologies with varying densities.

#Top.	#Agents	#Range(r_c^2)	#Density
D1-1	400	10^2	$4/r_c^2$
D1-2	800	10^2	$8/r_c^2$
D1-3	1,200	10^2	$12/r_c^2$
D1–4	1,600	10^2	$16/r_c^2$
D1-5	2,000	10^2	$20/r_c^2$
D1-6	2,400	10^2	$24/r_c^2$

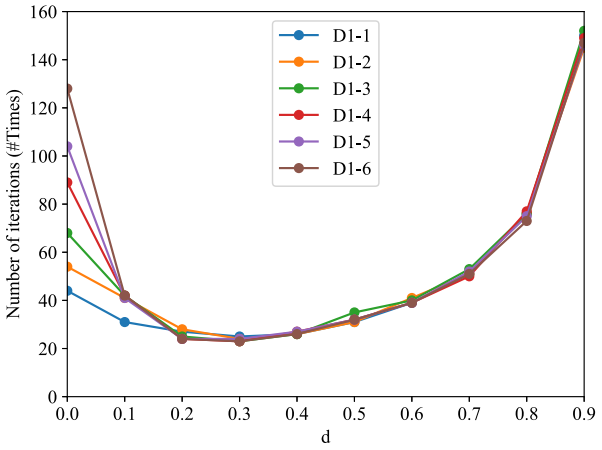


Fig. 14. Number of iterations (#Times) of DSG with SDB ($s = 4$) for six topologies.

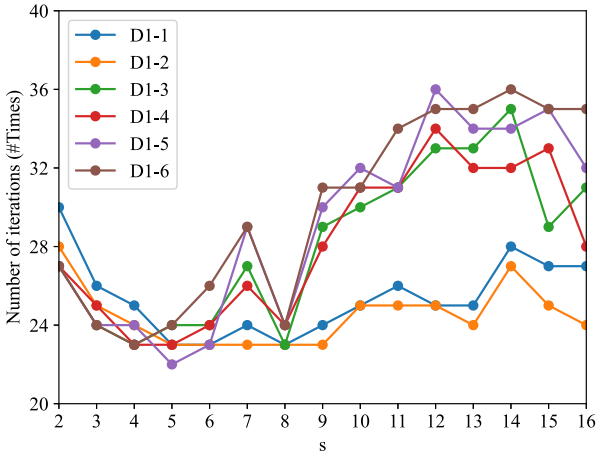


Fig. 15. Number of iterations (#Times) of SDB with DSG ($d = 0.3$) for six topologies.

4.3. Performance comparison and analysis

In this section, we combine the traditional CA and the RD algorithm in [3] with our SDB and DSG algorithms to obtain illustrative comparisons. In particular, the CA is used in each iteration to calculate the circumcircle of the convex hull of the current neighbor $N_i(t)$, with the center of the circumcircle serving as the target point. RD is a special case of DSG when $d = 0$. Thus, the SDB algorithm and the CA are consensus algorithms used to compute the control inputs of agents, whereas the DSG and RD algorithms are connectivity maintenance algorithms. We use the combined algorithms SDB(4)&DSG(0.3), SDB(8)&DSG(0.2), CA&RD, and CA&DSG(0.2) to compare the efficiency of convergence, where SDB(n) denotes the SDB algorithm with $s = n$ and DSG(n) denotes the DSG algorithm with $d = n$. As shown in Table 3, more topologies divided into different groups of experimental datasets (i.e., #DS) are designed. The construction of these topologies is similar to that used in the previous section. The simulation results based on datasets D1, D2, and D3 are shown in Fig. 16.

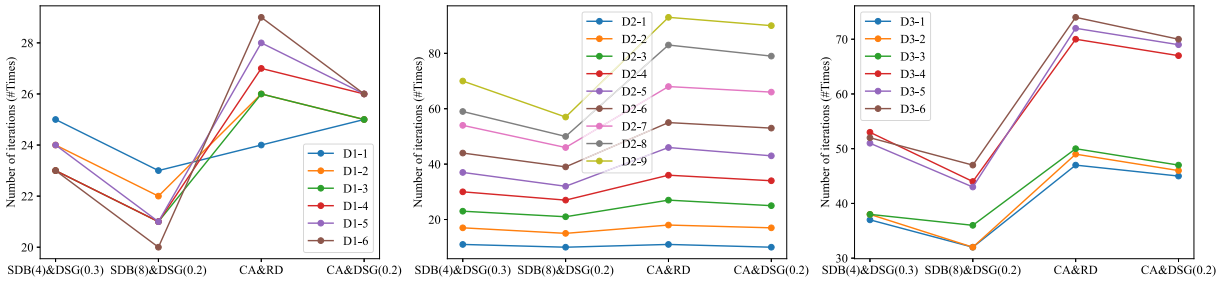
First, the SDB&DSG algorithm has a better convergence performance than the CA&RD algorithm, especially when the range of topology or the number of agents increases. As shown in Fig. 16, the SDB(8)&DSG(0.2) algorithm is better suited to large-scale and high-density networks.

Second, the convergence of the CA&DSG(0.2) algorithm is slightly faster than that of the CA&RD algorithm, indicating that combining DSG with other consensus algorithms is also efficient.

Third, from the results using datasets D1 and D3, there is no significant difference between the SDB(8)&DSG(0.2) and the SDB(4)&DSG(0.3) algorithms for a fixed range and a varying density. However, the results using D2 indicate that the performance of the SDB(8)&DSG(0.2) algorithm gradually increases compared with the SDB(4)&DSG(0.3) algorithm for a fixed density and a varying range. This finding shows that when the network topology range is large or agents are densely distributed, a larger value of s (e.g., $s = 8$) improves system convergence. This is because an agent may have many neighbors in large-

Table 3
Statistics of initial topologies for comparing four combination algorithms.

#DS.	#Top.	#Agents	#Range(r_c^2)	#Density
D1	D1-1	400	10^2	$4/r_c^2$
	D1-2	800	10^2	$8/r_c^2$
	D1-3	1,200	10^2	$12/r_c^2$
	D1–4	1,600	10^2	$16/r_c^2$
	D1-5	2,000	10^2	$20/r_c^2$
	D1-6	2,400	10^2	$24/r_c^2$
D2	D2-1	300	5^2	$12/r_c^2$
	D2-2	675	7.5^2	$12/r_c^2$
	D2-3	1,200	10^2	$12/r_c^2$
	D2-4	1,875	12.5^2	$12/r_c^2$
	D2-5	2,700	15^2	$12/r_c^2$
	D2-6	3,675	17.5^2	$12/r_c^2$
	D2-7	4,800	20^2	$12/r_c^2$
	D2-8	6,075	22.5^2	$12/r_c^2$
	D2-9	7,500	25^2	$12/r_c^2$
D3	D3-1	3600	15^2	$16/r_c^2$
	D3-2	4,500	15^2	$20/r_c^2$
	D3-3	5,400	15^2	$24/r_c^2$
	D3-4	6,400	20^2	$16/r_c^2$
	D3-5	8,000	20^2	$20/r_c^2$
	D3-6	9,600	20^2	$24/r_c^2$



(a) D1 (b) D2 (c) D3

Fig. 16. Number of iterations (#Times) of four combination algorithms using datasets D1, D2 and D3.

scale and high-density networks. Therefore, when $s = 4$, the area of each sector is too large, resulting in selected neighbors having high randomness, which may impair the convergence performance of the system.

5. Conclusion and future work

In this paper, we have investigated the consensus problem under switching and undirected network topologies. Both large-scale and high-density topologies have been considered in our scenario. In contrast to previous studies, we have developed a heuristic combinatorial algorithm that combines a distributed SDB consensus algorithm with a DSG connectivity maintenance algorithm, which has the following three advantages. (1) It can be divided into a consensus algorithm and a connectivity maintenance algorithm with respective optimizable parameters (i.e., s and d) and can be arbitrarily and seamlessly combined with other baseline algorithms. (2) It can be extended and applied with high versatility to high dimensional state space, or applied for formation control, flocking or other fundamental problems in computer science. (3) It is superior to other algorithms especially for application to both large-scale and high-density network topologies, as confirmed by extensive simulations.

In future work, we will focus on the design of consensus-seeking algorithms that preserve privacy [21] under switching directed topologies, the analysis of the performance and complexity of the algorithms, and the generalization of the algorithms from two-dimensional space to higher dimensional space.

CRediT authorship contribution statement

Guangqiang Xie: Conceptualization, Methodology, Writing – original draft. **Haoran Xu:** Software, Validation, Visualization, Formal analysis. **Yang Li:** Supervision. **Xianbiao Hu:** Writing – review & editing. **Chang-Dong Wang:** Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is partially supported by National Natural Science Foundation of China, Grant Nos. 62006047 and 618760439, Guangdong Natural Science Foundation, Grant No. 2021B0101220004. The authors would like to thank the anonymous reviewers and the editor for their insightful and helpful suggestions.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, athttps://doi.org/10.1016/j.ins.2022.06.079.

References

[1] A. Amirkhani, A.H. Barshooi, Consensus in multi-agent systems: a review, Artif. Intell. Rev. 55 (2022) 3897–3935.
[2] B. Chang, X. Mu, Z. Yang, J. Fang, Event-based secure consensus of muti-agent systems under asynchronous dos attacks, Appl. Math. Comput. 401 (2021) 126120.
[3] J. Cortés, S. Martínez, F. Bullo, Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions, IEEE Trans. Autom. Control 51 (2006) 1289–1298.
[4] G. Dong, H. Li, H. Ma, R. Lu, Finite-time consensus tracking neural network FTC of multi-agent systems, IEEE Trans. Neural Networks Learn. Syst. 32 (2021) 653–662.
[5] Y. Dong, J. Huang, Leader-following connectivity preservation rendezvous of multiple double integrator systems based on position measurement only, IEEE Trans. Autom. Control 59 (2014) 2598–2603.
[6] Y. Dong, S. Xu, A novel connectivity-preserving control design for rendezvous problem of networked uncertain nonlinear systems, IEEE Trans. Neural Networks Learn. Syst. 31 (2020) 5127–5137.
[7] Y. Fan, G. Feng, Y. Wang, Combination framework of rendezvous algorithm for multi-agent systems with limited sensing ranges, Asian J. Control 13 (2011) 283–294.
[8] Z. Feng, C. Sun, G. Hu, Robust connectivity preserving rendezvous of multirobot systems under unknown dynamics and disturbances, IEEE Trans. Control Network Syst. 4 (2016) 725–735.
[9] R.A. Horn, C.R. Johnson, Matrix analysis, Cambridge University Press, 2012.
[10] W. Hu, L. Liu, G. Feng, Consensus of linear multi-agent systems by distributed event-triggered strategy, IEEE Trans. Cybern. 46 (2015) 148–157.
[11] A. Jadbabaie, J. Lin, A.S. Morse, Coordination of groups of mobile autonomous agents using nearest neighbor rules, IEEE Trans. Autom. Control. 48 (2003) 988–1001.
[12] L. Ji, S. Tong, H. Li, Dynamic group consensus for delayed heterogeneous multi-agent systems in cooperative-competitive networks via pinning control, Neurocomputing 443 (2021) 1–11.
[13] L. Ji, C. Wang, C. Zhang, H. Wang, H. Li, Optimal consensus model-free control for multi-agent systems subject to input delays and switching topologies, Inf. Sci. 589 (2022) 497–515.
[14] D. Le, E. Plaku, Cooperative, dynamics-based, and abstraction-guided multi-robot motion planning, J. Artif. Intell. Res. 63 (2018) 361–390.
[15] J. Liu, Y. Zhang, C. Sun, Y. Yu, Fixed-time consensus of multi-agent systems with input delay and uncertain disturbances via event-triggered control, Inf. Sci. 480 (2019) 261–272.
[16] J. Liu, Y. Zhang, Y. Yu, C. Sun, Fixed-time leader-follower consensus of networked nonlinear systems via event/self-triggered control, IEEE Trans. Neural Networks Learn. Syst. 31 (2020) 5029–5037.
[17] Y. Liu, F. Zhang, P. Huang, Y. Lu, Fixed-time consensus tracking control with connectivity preservation for strict-feedback nonlinear multi-agent systems, ISA Trans. 123 (2022) 14–24.
[18] L. Moreau, Stability of multiagent systems with time-dependent communication links, IEEE Trans. Autom. Control 50 (2005) 169–182.
[19] S. Motsch, E. Tadmor, Heterophilious dynamics enhances consensus, SIAM Rev. 56 (2014) 577–621.
[20] A. Nedic, A. Olshevsky, M.G. Rabbat, Network topology and communication-computation tradeoffs in decentralized optimization, Proc. IEEE 106 (2018) 953–976.
[21] E. Nozari, P. Tallapragada, J. Cortés, Differentially private average consensus: Obstructions, trade-offs, and optimal algorithm design, Autom. 81 (2017) 221–231.
[22] R. Olfati-Saber, J.A. Fax, R.M. Murray, Consensus and cooperation in networked multi-agent systems, Proc. IEEE 95 (2007) 215–233.
[23] R. Olfati-Saber, R.M. Murray, Consensus problems in networks of agents with switching topology and time-delays, IEEE Trans. Autom. Control 49 (2004) 1520–1533.
[24] R. Parasuraman, J. Kim, S. Luo, B. Min, Multipoint rendezvous in multirobot systems, IEEE Trans. Cybern. 50 (2020) 310–323.
[25] B.S. Park, S.J. Yoo, Connectivity-maintaining and collision-avoiding performance function approach for robust leader-follower formation control of multiple uncertain underactuated surface vessels, Autom. 127 (2021) 109501.
[26] Z. Peng, Y. Zhao, J. Hu, B.K. Ghosh, Data-driven optimal tracking control of discrete-time multi-agent systems with two-stage policy iteration algorithm, Inf. Sci. 481 (2019) 189–202.
[27] S. Raja, G. Habibi, J.P. How, Communication-aware consensus-based decentralized task allocation in communication constrained environments, IEEE Access 10 (2022) 19753–19767.
[28] E. Restrepo, A. Loría, I. Sarras, J. Marzat, Edge-based strict lyapunov functions for consensus with connectivity preservation over directed graphs, Autom. 132 (2021) 109812.

[29] L. Sabattini, C. Secchi, N. Chopra, Decentralized estimation and control for preserving the strong connectivity of directed graphs, IEEE Trans. Cybern. 45 (2014) 2273–2286.
[30] M. Sader, Z. Chen, Z. Liu, C. Deng, Distributed robust fault-tolerant consensus control for a class of nonlinear multi-agent systems with intermittent communications, Appl. Math. Comput. 403 (2021) 126166.
[31] S. Su, Z. Lin, Distributed consensus control of multi-agent systems with higher order agent dynamics and dynamically changing directed interaction topologies, IEEE Trans. Autom. Control 61 (2015) 515–519.
[32] S. Su, Z. Lin, Connectivity enhancing coordinated tracking control of multi-agent systems with a state-dependent jointly-connected dynamic interaction topology, Autom. 101 (2019) 431–438.
[33] Y. Su, Leader-following rendezvous with connectivity preservation and disturbance rejection via internal model approach, Automatica 57 (2015) 203–212.
[34] Z. Sun, J. Huang, A note on connectivity of multi-agent systems with proximity graphs and linear feedback protocol, Automatica 45 (2009) 1953–1956.
[35] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, O. Shochet, Novel type of phase transition in a system of self-driven particles, Phys. Rev. Lett. 75 (1995) 1226–1229.
[36] D. Wang, N. Zhang, J. Wang, W. Wang, Cooperative containment control of multiagent systems based on follower observers with time delay, IEEE Trans. Syst. Man Cybern.: Syst. 47 (2016) 13–23.
[37] Q. Wang, Z. Duan, Y. Lv, Q. Wang, G. Chen, Linear quadratic optimal consensus of discrete-time multi-agent systems with optimal steady state: A distributed model predictive control approach, Autom. 127 (2021) 109505.
[38] G. Wen, W.X. Zheng, On constructing multiple lyapunov functions for tracking control of multiple agents with switching topologies, IEEE Trans. Autom. Control. 64 (2019) 3796–3803.
[39] G. Xie, J. Chen, Y. Li, Hybrid-order network consensus for distributed multi-agent systems, J. Artif. Intell. Res. 70 (2021) 389–407.
[40] S.J. Yoo, Connectivity-preserving design strategy for distributed cooperative tracking of uncertain nonaffine nonlinear time-delay multi-agent systems, Inf. Sci. 514 (2020) 541–556.
[41] J. Yu, X. Dong, Q. Li, Z. Ren, Practical time-varying output formation tracking for high-order multi-agent systems with collision avoidance, obstacle dodging and connectivity maintenance, J. Frankl. Inst. 356 (2019) 5898–5926.
[42] P. Zhang, H. Xue, S. Gao, J. Zhang, Distributed adaptive consensus tracking control for multi-agent system with communication constraints, IEEE Trans. Parallel Distributed Syst. 32 (2021) 1293–1306.
[43] X. Zhang, B. Li, S. Cai, Y. Wang, Efficient local search based on dynamic connectivity maintenance for minimum connected dominating set, J. Artif. Intell. Res. 71 (2021) 89–119.
[44] Y. Zheng, J. Ma, L. Wang, Consensus of hybrid multi-agent systems, IEEE Trans. Neural Networks Learn. Syst. 29 (2017) 1359–1365.
[45] Y. Zheng, L. Wang, Consensus of switched multiagent systems, IEEE Trans. Circuits Syst. II Express Briefs 63 (2015) 314–318.