



# Local information-based control for probabilistic swarm distribution guidance

Inmo Jang<sup>1</sup> · Hyo-Sang Shin<sup>1</sup> · Antonios Tsourdos<sup>1</sup>

Received: 13 April 2018 / Accepted: 2 November 2018 / Published online: 16 November 2018  
© The Author(s) 2018

## Abstract

This paper proposes a closed-loop decentralised framework for *swarm distribution guidance*, which disperses homogeneous agents over bins to achieve a desired density distribution by using feedback gains from the current swarm status. The key difference from existing works is that the proposed framework utilises only *local information*, not *global information*, to generate the feedback gains for stochastic policies. Dependency on local information entails various advantages including reduced inter-agent communication, a shorter timescale for obtaining new information, asynchronous implementation, and deployability without a priori mission knowledge. Our theoretical analysis shows that, even utilising only local information, the proposed framework guarantees convergence of the agents to the desired status, while maintaining the advantages of existing closed-loop approaches. Also, the analysis explicitly provides the design requirements to achieve all the advantages of the proposed framework. We provide implementation examples and report the results of empirical tests. The test results confirm the effectiveness of the proposed framework and also validate the robustness enhancement in a scenario of partial disconnection of the communication network.

**Keywords** Swarm robotics · Multi-agent systems · Decentralised decision-making · Self-organisation · Probabilistic swarm guidance · Markov-chain-based

## 1 Introduction

This paper addresses a *swarm distribution guidance problem* (Acikmese and Bayard 2012, 2014, 2015), which concerns how to distribute a swarm of agents into given bins in order to achieve the desired population fraction (or swarm density) for each bin, as illustrated in Fig. 1. In this study, we propose a closed-loop framework that relies on *the Local Information*

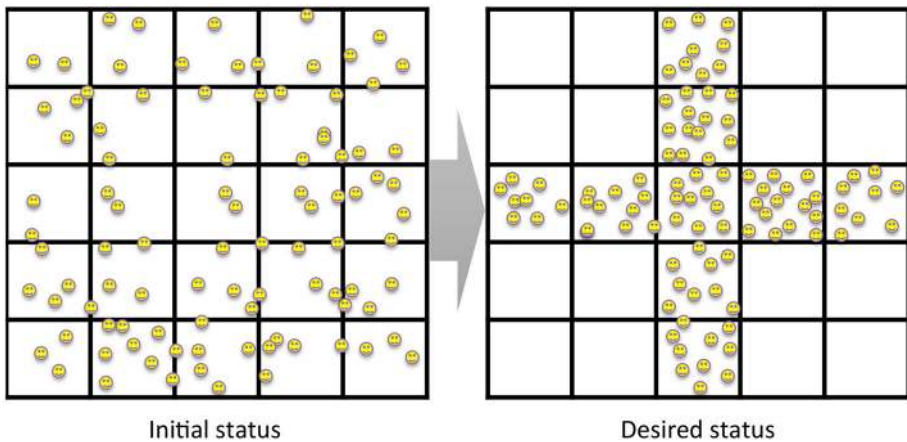
---

✉ Hyo-Sang Shin  
h.shin@cranfield.ac.uk

Inmo Jang  
inmo.jang@cranfield.ac.uk

Antonios Tsourdos  
a.tsourdos@cranfield.ac.uk

<sup>1</sup> Centre for Autonomous and Cyber-Physical Systems, Cranfield University, Bedford MK43 0AL, UK



**Fig. 1** Swarm distribution guidance problem: how to distribute homogeneous agents into bins, satisfying a desired population fraction for each bin

*Consistency Assumption* (LICA), i.e. only *local* information needs to be consistently known by the *local agent groups*. The idea behind this framework was inspired by fish swarm behaviours, where each fish adjusts its individual behaviour based on the behaviours of its neighbours (Couzin et al. 2002, 2005; Gautrais et al. 2008; Hoare et al. 2004). Likewise, each agent in the proposed framework uses its *local information*<sup>1</sup> to generate its individual stochastic policies.

Dependency on local information enables various advantages. The first obvious benefit is reduction in inter-agent communication required for decentralised decision-making, as can be seen from the experimental results in Fig. 6d. The proposed approach consequently “provides a much shorter timescale for using new information because agents are not required to ensure that this information has propagated to the entire team before using it” (Johnson et al. 2016). Exploiting LICA also enables an asynchronous implementation of the framework (as shown in Sect. 5) and provides robustness against dynamical changes in bins as well as in agents. Furthermore, the agents do not need to acquire global mission knowledge such as a desired distribution a priori (i.e. before a mission starts) as long as they can sense all the neighbour bins’ desired swarm densities in an impromptu manner.

In this study, the stability and performance of the proposed framework are extensively investigated via theoretical analysis and empirical tests. In the theoretical analysis, we prove that the agents asymptotically converge towards the desired swarm distribution even using local information-based feedback. This paper also provides the design requirements for the time-inhomogeneous Markov chain to achieve all the benefits aforementioned. Empirical tests demonstrate the performance of the proposed framework in three implementation examples: (1) travelling cost minimisation; (2) convergence rate maximisation under flux upper limits; and (3) quorum-based policies generation [similar to Halasz et al. (2007), Hsieh et al. (2008)]. Moreover, we show an asynchronous version of the proposed framework and demonstrate that it is more robust against sporadic network disconnection of partial agents, compared with the recent work in Bandyopadhyay et al. (2017).

<sup>1</sup> This paper will use the term “local information” to refer to the knowledge available to all agents in the same bin and the information obtainable via communication with other agents in all *neighbour bins* (defined in Definition 7).

## 1.1 Related work

For swarm distribution guidance problems, there have been two widely studied approaches: probabilistic approaches based on Markov chains (Chattopadhyay and Ray 2009; Acikmese and Bayard 2012, 2014, 2015; Demir and Acikmese 2015; Luo et al. 2014; Demir et al. 2015; Morgan et al. 2014; Bandyopadhyay et al. 2017) or differential equations (Halasz et al. 2007; Hsieh et al. 2008; Berman et al. 2008, 2009; Mather and Hsieh 2011; Prorok et al. 2017). These approaches generally focus not on individual agents, but on their ensemble dynamics. This is the reason why they are often called *Eulerian* (Bandyopadhyay et al. 2017; Morgan et al. 2014) or *macroscopic* frameworks (Lerman et al. 2005; Mather and Hsieh 2011). In these approaches, swarm densities over given bins are represented as system states. A state-transition matrix for the states describes *stochastic (decision) policies*, i.e. the probabilities that agents in a bin switch to another within a time unit. Accordingly, individual agents make decisions in a random, independent, and memoryless manner.

These approaches can be classified into two framework groups: *open-loop* (Berman et al. 2008, 2009; Mather and Hsieh 2011; Chattopadhyay and Ray 2009; Acikmese and Bayard 2012, 2014, 2015) and *closed-loop* frameworks (Halasz et al. 2007; Hsieh et al. 2008; Luo et al. 2014; Demir et al. 2015; Morgan et al. 2014; Bandyopadhyay et al. 2017; Prorok et al. 2017). Agents under open-loop frameworks are controlled by time-invariant stochastic policies. The policies, which make a swarm converge to a desired distribution, are predetermined by a central controller and broadcasted to each agent before the mission begins. Communication between agents is hardly required during the mission, and thus the communication complexity is minimised. However, the agents only have to follow the predetermined policies without incorporating any feedback, and thus there still remain some agents who unnecessarily and continuously move around bins even after the swarm reaches the desired status. Therefore, the trade-off between convergence rate and long-term system efficiency becomes critical in these frameworks (Berman et al. 2009).

*Closed-loop* frameworks allow agents to adaptively construct their own stochastic policies at the expense of communicating with other agents to perceive the concurrent swarm status. Based on such information, the agents can synthesise a time-inhomogeneous transition matrix to achieve certain objectives and requirements: for example, maximising convergence rates (Demir et al. 2015), minimising travelling costs (Bandyopadhyay et al. 2017), and temporarily adjusting the policies when bins are overpopulated or underpopulated (Halasz et al. 2007; Hsieh et al. 2008). In particular, Bandyopadhyay et al. (2017) recently proposed a closed-loop algorithm that exhibits faster convergence as well as less undesirable transition behaviours, compared with an open-loop algorithm. This algorithm can mitigate the issue with trade-off that is critical in open-loop frameworks.

To the best of our knowledge, most of the existing closed-loop algorithms are based on *the Global Information Consistency Assumption* (GICA) (Johnson et al. 2016). GICA implies that the information necessary to generate time-varying stochastic policies is about the global swarm status (i.e. *global information*), and it also needs to be consistently known by the *entire swarm*. Achieving GICA requires each agent to somehow interact with all the others in a multi-hop fashion and it “happens on a global communication timescale” (Johnson et al. 2016).

The main focus of this work is to relax GICA and to formally show that relying on LICA is sufficient to solve the swarm distribution guidance problem. In fact, the existing closed-loop methods in Bandyopadhyay et al. (2017) and Prorok et al. (2017) do not necessarily require every agent to perceive global knowledge exactly, while providing a graceful performance degradation even when local estimates of the global information are used. However, the key

**Table 1** Nomenclature

Symbol	Description
$\mathcal{B}_j$	The $j$ th bin amongst a set of $n_b$ bins (Definition 1)
$\mathcal{A}$	A set of $n_a$ agents (Definition 1)
$\mathbf{A}_k$	Physical motion constraint matrix (Definition 2)
$\mathbf{x}_k$	The current (global) swarm distribution (Definition 4)
$M_k$	Stochastic policy of the agents (Definition 5)
$\Theta$	The desired swarm distribution (Definition 6)
$\mathcal{N}_k(j)$	A set of neighbour bins of the $j$ th bin (Definition 7)
$\mathcal{A}_{\mathcal{N}_k(j)}$	A set of agents in $\mathcal{N}_k(j)$
$\mathbf{n}_k[j]$	The number of agents in $\mathcal{B}_j$ at time instant $k$ [Eq. (4)]
$\bar{\mathbf{x}}_k[j]$	The current <i>local</i> swarm density at the $j$ th bin [Eq. (4)]
$\bar{\Theta}[j]$	The locally desired swarm density at the $j$ th bin [Eq. (5)]
$P_k$	Primary guidance matrix [Eq. (10)]
$S_k$	Secondary guidance matrix [Eq. (10)]
$\bar{\xi}_k[j]$	Primary local-feedback gain [e.g. Eq. (6)]
$G_k[j]$	Secondary local-feedback gain [Eq. (9)]

difference from the previous works is that the proposed LICA-based framework utilises local information as its feedback gains. Consequently, the proposed framework exploits the various advantages from LICA, which could not be straightforwardly achievable in a GICA-based framework.

### 1.2 Outline of this paper

The rest of the paper is organised as follows. Section 2 introduces essential definitions and notations of a Markov-chain-based approach. In Sect. 3, we describe the desired features for swarm distribution guidance, propose a closed-loop framework with its design requirements, and perform a theoretical analysis. We provide examples of how to exploit the framework for specific problems in Sect. 4 and asynchronous implementation in Sect. 5. Numerical experimental results are provided in Sect. 6, followed by concluding remarks in Sect. 7.

## 2 Preliminaries

This section provides the basic concept of a Markov-chain-based approach and presents definitions and assumptions necessary for our proposed framework, which will be shown in Sect. 3. Note that most of them are embraced from the existing literature (Demir et al. 2015; Bandyopadhyay et al. 2017). In this paper,  $\emptyset$ ,  $\mathbf{0}$ ,  $I$ , and  $\mathbf{1}$  denote the empty set, the zero matrix of appropriate sizes, the identity matrix of appropriate sizes, and a row vector with all elements are equal to one, respectively.  $\mathbf{v} \in \mathbb{P}^n$  is a  $1 \times n$  row-stochastic vector such that  $\mathbf{v} \geq \mathbf{0}$  and  $\mathbf{v} \cdot \mathbf{1}^T = 1$ .  $\mathbf{v}[i]$  indicates the  $i$ th element of vector  $\mathbf{v}$ . Note that some of the symbols and the definitions primarily used in this paper are shown in Table 1.

**Definition 1** (*Agents and Bins*) A set of  $n_a$  homogeneous agents  $\mathcal{A} = \{a_1, a_2, \dots, a_{n_a}\}$  are supposed to be distributed over a prescribed region in a state space  $\mathcal{B}$ . The entire space

is partitioned into  $n_b$  disjoint bins (subspaces) such that  $\mathcal{B} = \cup_{j=1}^{n_b} \mathcal{B}_j$  and  $\mathcal{B}_j \cap \mathcal{B}_l = \emptyset, \forall l \neq j$ . We also regard  $\mathcal{B} = \{\mathcal{B}_1, \dots, \mathcal{B}_{n_b}\}$  as the set of all the bins. Each bin  $\mathcal{B}_j$  represents a predefined range of an agent’s state, e.g. position, task assigned, behaviour, etc. Note that binning of the state space should be done problem-specifically in a way that accommodates all the following assumptions and definitions in this section. For example, the bin sizes are not necessarily required to be uniform, but can vary depending on physical constraints of the space or communication radii of given agents.

**Definition 2 (Agent motion constraint)** The agent motion constraints over the given bins  $\mathcal{B}$  are represented by  $\mathbf{A}_k \in \{0, 1\}^{n_b \times n_b}$ , where  $\mathbf{A}_k[j, l]$  is one if any agent in  $\mathcal{B}_j$  at time instant  $k$  is able to transition to  $\mathcal{B}_l$  by the next time instant, and zero otherwise.  $\mathbf{A}_k$  is symmetric and irreducible (defined in “Appendix”);  $\mathbf{A}_k[j, j] = 1, \forall j$ . Equivalently, it can be also said that the topology of the bins is modelled as a bidirectional and strongly connected graph,  $\mathcal{G}_k = (\mathbf{A}_k, \mathcal{B})$ , where  $\mathbf{A}_k$  is edges (i.e. adjacent matrix) and  $\mathcal{B}$  is nodes (i.e. bins).

**Definition 3 (Agent’s state)** Let  $\mathbf{s}_k^i \in \{0, 1\}^{n_b}$  be the state indicator vector of agent  $a_i \in \mathcal{A}$  at time instant  $k$ . If the agent’s state belongs to bin  $\mathcal{B}_j$ , then  $\mathbf{s}_k^i[j] = 1$ , otherwise 0. Note that the definition of the time instant will be described later in Definition 8.

**Definition 4 (Current swarm distribution)** The current (global) swarm distribution  $\mathbf{x}_k \in \mathbb{P}^{n_b}$  is a row-stochastic vector such that each element  $\mathbf{x}_k[j]$  is the population fraction (or swarm density) of  $\mathcal{A}$  in bin  $\mathcal{B}_j$  at time instant  $k$ :

$$\mathbf{x}_k := \frac{1}{|\mathcal{A}|} \sum_{\forall a_i \in \mathcal{A}} \mathbf{s}_k^i. \tag{1}$$

**Definition 5 (Stochastic policy)** The probability that agent  $a_i$  in bin  $\mathcal{B}_j$  at time instant  $k$  will transition to bin  $\mathcal{B}_l$  before the next time instant is called its stochastic policy, denoted as:

$$M_k^i[j, l] := \text{Prob}(\mathbf{s}_{k+1}^i[l] = 1 | \mathbf{s}_k^i[j] = 1). \tag{2}$$

Note that  $M_k^i \in \mathbb{P}^{n_b \times n_b}$  is a row-stochastic matrix such that  $M_k^i \geq \mathbf{0}$  and  $M_k^i \cdot \mathbf{1}^\top = \mathbf{1}^\top$ , and will be referred as Markov matrix.

Assuming that all agents in bin  $\mathcal{B}_j$  at time instant  $k$  are independently governed by an identical row-stochastic vector (denoted by  $M_k[j, l], \forall l$ ), the dynamics of the swarm distribution is modelled by

$$\mathbf{x}_{k+1} = \mathbf{x}_k M_k, \tag{3}$$

as  $n_a$  increases towards infinity. We will examine in Sect. 6.4 the performance degradation of possible differences between individual  $M_k^i$  on the proposed framework. Please keep in mind that, for each agent in  $\mathcal{B}_j$ , it is not necessary to know the other bins’ stochastic policies (i.e.  $M_k[j', l], \forall j' \neq j, \forall l$ ), and that this paper only introduces such a matrix form, e.g. Eq. (3), for the sake of theoretical analysis of the ensemble.

Every agent in each bin  $\mathcal{B}_j$  executes Algorithm 1 at every time instant  $k$ . The detail regarding how to generate its stochastic policies (i.e. Line 2) will be presented in Sect. 3.

**Definition 6 (Desired swarm distribution)** The desired swarm distribution  $\Theta \in \mathbb{P}^{n_b}$  is a row-stochastic vector such that each element  $\Theta[j]$  indicates the desired swarm density for bin  $\mathcal{B}_j$ .

**Algorithm 1** Probabilistic Swarm Distribution Guidance

---

```

// For each agent at time instant k:
1: Identify the current bin  $\mathcal{B}_j$ 
2: Compute  $M_k[j, l], \forall l$ 
3: Draw a random number  $z$  from the uniform distribution on  $[0, 1]$ 
4: Select bin  $\mathcal{B}_q$  such that
   
$$\sum_{l=1}^{q-1} M_k[j, l] \leq z < \sum_{l=1}^q M_k[j, l]$$

5: Move to the selected bin

```

---

**Assumption 1** For ease of description for this paper, we assume that  $\Theta[j] > 0, \forall j \in \{1, \dots, n_b\}$ . In practice, there may exist some bins whose desired swarm densities are zero. These bins can be accommodated by adopting any subroutine ensuring that all agents eventually move to and remain in any of the positive-desired-density bins (for an example, refer to Sect. 6.6). In this case, it should be assumed that the agent motion constraints over every bin  $\mathcal{B}_j$  such that  $\Theta[j] > 0$  are at least (bidirectionally) strongly connected (i.e.  $(\Theta^\top \Theta) \odot \mathbf{A}_k$  is irreducible, where  $\odot$  denotes the Hadamard product).

**Assumption 2** (Communicational connectivity over bins) The physical motion constraint of a robotic agent is, in general, more stringent than its communicational constraint. From this, it can be assumed that if the transition of agents between bin  $\mathcal{B}_j$  and  $\mathcal{B}_l$  is allowed within a unit time interval (i.e.  $\mathbf{A}_k[j, l] = 1$ ), then one bin is within the communication range of the agents in the other bin, and vice versa.

**Definition 7** (*Neighbour bins, neighbour agents, and local information*) For each bin  $\mathcal{B}_j$ , we define the set of its *neighbour bins* as  $\mathcal{N}_k(j) = \{\forall \mathcal{B}_l \in \mathcal{B} \mid \mathbf{A}_k[j, l] = 1\}$ . From Assumption 2, each agent in  $\mathcal{B}_j$  can communicate with other agents in  $\mathcal{N}_k(j)$ . The set of these agents is called *neighbour agents*, denoted by  $\mathcal{A}_{\mathcal{N}_k(j)} = \{\forall a_i \in \mathcal{A} \mid \mathbf{s}_k^i[l] = 1, \forall l : \mathcal{B}_l \in \mathcal{N}_k(j)\}$ . This paper refers to the local knowledge available from  $\mathcal{A}_{\mathcal{N}_k(j)}$  as *the local information of the agents in bin  $\mathcal{B}_j$* .

**Assumption 3** (Known information) Each agent has reasonable sensing capabilities such that agents in bin  $\mathcal{B}_j$  can perceive neighbour bins' information such as  $\Theta[l]$  and  $\mathbf{A}_k[j, l]$  in real time under Assumption 2. Note that the global information regarding  $\Theta$  and  $\mathbf{A}_k$  is not required to be known by the agents a priori, as will be described in Remark 3 later. Other predetermined values such as variables regarding objective functions and design parameters (which will be introduced later) are known to all the agents.

**Assumption 4** (Agent's capability) Each agent can determine the bin to which it belongs, and know the locations of neighbour bins so that it can navigate towards any of the bins. The agent is capable of avoiding collisions with other agents.

**Assumption 5** (The number of agents) The number of agents  $n_a$  is large enough such that the time evolution of the swarm distribution is governed by the Markov process in Eq. (3). Although the finite cardinality of the agents may cause a residual convergence error (i.e. a sense of the difference between  $\Theta$  and  $\mathbf{x}_\infty$ ), a lower bound on  $n_a$  that probabilistically guarantees a certain level of convergence error is analysed in Bandyopadhyay et al. (2017, Theorem 6). Note that this theorem is generally applicable and thus is also valid for our work.

**Definition 8** (*Time instant*) We define *time instant  $k$*  to be the time when all the agents complete not only transitioning towards the bins selected at time instant  $k - 1$ , but also obtain

the local information necessary to construct  $M_k$ . Hence, a temporal interval (in the real-time scale) between any two sequential time instants may not always be consistent in practice. This might be because of the required inter-agent communication and/or physical congestion, which are varied at every time instant. In the worst case, due to some bins whose agents are somehow not ready in terms of transitioning or obtaining local knowledge, the temporal interval may be arbitrarily elongated. However, the proposed method can accommodate those bins by incorporating the asynchronous implementation in Sect. 5. It is worth mentioning that since our proposed approach demands relatively less communication burden on the agents, the temporal intervals would be shorter than those in GICA-based approaches.

### 3 The proposed closed-loop framework under LICA

The objective of the swarm distribution guidance problem considered in this paper is to distribute a set of agents  $\mathcal{A}$  over a set of bins  $\mathcal{B}$  by the Markov matrix  $M_k$  so as to offer the following desired features:

**Desired Feature 1** The swarm distribution  $\mathbf{x}_k$  asymptotically converges to the desired swarm distribution  $\Theta$  as time instant  $k$  goes to infinity.

**Desired Feature 2** The transition of the agents between the bins is encoded to enforce the property that  $M_k$  becomes close to  $I$  as  $\mathbf{x}_k$  converges to  $\Theta$ . This implies that the agents are settling down as being close to  $\Theta$ , and thus unnecessary transitions, which would have occurred in an open-loop framework, can be reduced. Moreover, the agents identify and compensate any partial loss or failure of the swarm distribution.

**Desired Feature 3** For each agent in bin  $\mathcal{B}_j$ , the information required for generating time-varying stochastic policies is not global information about the entire agents  $\mathcal{A}$  but only local information available from local agent group  $\mathcal{A}_{\mathcal{N}_k(j)}$ . Thereby, the resultant time-inhomogeneous Markov process is based on LICA and has benefits such as reduced inter-agent communication, a shorter timescale for obtaining new information (than GICA), and the possibility to be implemented asynchronously.

This section proposes a LICA-based framework for the swarm distribution guidance problem. The framework is different from the recent closed-loop algorithms in Bandyopadhyay et al. (2017) and Demir et al. (2015) in the sense that they utilise global information [e.g. the current swarm distribution in Eq. (1)] to construct a time-inhomogeneous Markov matrix, whereas ours uses the local information in Eq. (7), which will be shown later. We study how, in spite of using such relatively insufficient information, the desired features aforementioned can be achieved in the proposed framework. Before that, we introduce biological findings that concern decision-making mechanisms of a fish school, which motivate our framework, and particularly Desired Feature 3.

#### 3.1 Motivation

For a fish school, it has commonly been assumed that crowdedness limits an individual's perception range over other individuals, and that the school cardinality restricts the ability to recognise other individuals (Couzin et al. 2005). How fish end up with collective behaviours is different from the ways of other social species such as bees and ants, which are known



to use recruitment signals for the guidance of the entire swarm (Seeley 1995; Keller et al. 2000). Thus, in the biology domain, a question naturally has arisen about the decision-making mechanism of fish in an environment where only local information is available and information transfer between members does not explicitly happen (Partridge 1982; Couzin et al. 2005; Becco et al. 2006; Couzin et al. 2002; Gautrais et al. 2008; Hoare et al. 2004).

It has been experimentally shown that fish’s swimming activities vary depending on their perceivable neighbours. According to Partridge (1982), fish have the tendency to maintain their statuses (e.g. position, speed, and heading angle) relative to those of other nearby fish, which results in their organised formation structures. In addition, Becco et al. (2006) shows that spatial density of fish has influences on both the minimum distances between them and the primary orientation of the fish school. Based on this knowledge, the works in (Couzin et al. 2002, 2005; Gautrais et al. 2008; Hoare et al. 2004) suggest individual-based models to further understand the collective behavioural mechanisms of fish: for example, their repelling, attracting, and orientating behaviours (Couzin et al. 2002; Gautrais et al. 2008); how the density of informed fish affects the elongation of the formation structure (Couzin et al. 2005); and group-size choices (Hoare et al. 2004). The common and fundamental characteristic of these models is that every agent maintains or adjusts its personal status with consideration of those of other individuals within its limited perception range.

As inspired by the understanding of fish, we believe that there must be an enhanced swarm distribution guidance approach in which each agent only needs to keep its relative status by relying on local information available from its nearby neighbours. In our approach, the agents are not required to possess global knowledge, and thereby the corresponding requirement of extensive information sharing over all the agents can be alleviated.

### 3.2 The local information required in the proposed approach

We will show that global information is not required to generate feedback gains to operate closed-loop frameworks for robotic swarms. Instead, the main underlying idea is to use the deviation of current and desired swarm density at each local bin as local-feedback gain. Specifically, in most of GICA-based frameworks, the feedback gains are generated from the difference between  $\mathbf{x}_k$  and  $\Theta$ , which requires global information. In contrast, in the proposed LICA-based framework, agents in bin  $\mathcal{B}_j$  use the difference between *the current local swarm density*  $\bar{\mathbf{x}}_k[j]$  and the *locally desired swarm density*  $\bar{\Theta}[j]$ , which are, respectively, defined as follows:

$$\bar{\mathbf{x}}_k[j] := \frac{\mathbf{n}_k[j]}{\sum_{\forall \mathcal{B}_l \in \mathcal{N}_k(j)} \mathbf{n}_k[l]}, \tag{4}$$

where  $\mathbf{n}_k[j]$  is the number of agents in  $\mathcal{B}_j$  at time instant  $k$  (see Fig. 2 for an example); and

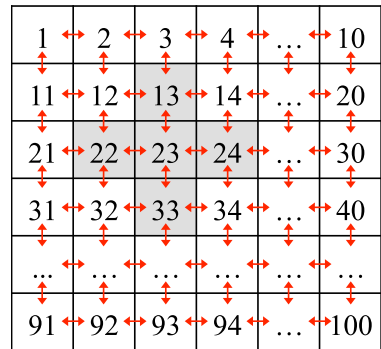
$$\bar{\Theta}[j] := \frac{\Theta[j]}{\sum_{\forall \mathcal{B}_l \in \mathcal{N}_k(j)} \Theta[l]}. \tag{5}$$

The two quantities are both locally available information within  $\mathcal{A}_{\mathcal{N}_k(j)}$ . The difference between  $\bar{\mathbf{x}}_k[j]$  and  $\bar{\Theta}[j]$  is utilised for a local feedback gain, denoted by  $\bar{\xi}_k[j]$ , which is a scalar in  $(0, 1]$  that monotonically decreases as  $\bar{\mathbf{x}}_k[j]$  converges to  $\bar{\Theta}[j]$ . For instance, this paper uses

$$\bar{\xi}_k[j] := \left( \frac{|\bar{\Theta}[j] - \bar{\mathbf{x}}_k[j]|}{\bar{\Theta}[j]} \right)^\alpha, \tag{6}$$



**Fig. 2** An example showing how to calculate  $\bar{x}_k[j]$ : for bin  $\mathcal{B}_{23}$ ,  $\bar{x}_k[23] = \mathbf{n}_k[23]/(\mathbf{n}_k[13] + \mathbf{n}_k[22] + \mathbf{n}_k[23] + \mathbf{n}_k[24] + \mathbf{n}_k[33])$ . In the proposed framework, agents in the bin only need to obtain the local information from other agents in its neighbour bins (shaded). Note that each square indicates each bin, and the red arrow between two bins  $\mathcal{B}_j$  and  $\mathcal{B}_l$  means that  $\mathbf{A}_k[j, l] = 1$



being saturated to  $[\epsilon_\xi, 1]$  if the value lies outside this range. This is called *primary local-feedback gain*, controlling the primary guidance matrix  $P_k$  (shown in the next subsection). Here,  $\alpha > 0$  is the sensitivity parameter affecting  $\bar{\xi}_k[j]$  with regard to the difference between  $\bar{x}_k[j]$  and  $\bar{\Theta}[j]$  (as shown in Fig. 5a);  $\epsilon_\xi > 0$  is a reasonably small positive value ensuring that  $\bar{\xi}_k[j] > 0$  in order to mathematically guarantee (R4), which will be described in the next subsection. How to use  $\bar{\xi}_k[j]$  explicitly may be different depending on different applications, and hence it will be given along with some implementation examples in Sect. 4.

**Remark 1** Equation (4) is equivalent to the  $j$ th element of the following vector:

$$\bar{x}_k^j := \frac{1}{|\mathcal{A}_{\mathcal{N}_k(j)}|} \sum_{\forall a_i \in \mathcal{A}_{\mathcal{N}_k(j)}} s_k^i. \tag{7}$$

Here, we intentionally introduce Eq. (7) for ease of comparison with the information required in the existing literature [i.e. Eq. (1)]. Equation (7) implies that, in order for each agent in bin  $\mathcal{B}_j$  to estimate  $\bar{x}_k^j[j]$  (i.e. the current local swarm density  $\bar{x}_k[j]$ ), the set of other agents whose information is necessary is just  $\mathcal{A}_{\mathcal{N}_k(j)}$ . That is, each agent needs to have neither a large perception radius nor an extensive information consensus process over the entire agents.

**Remark 2** In the rest of this paper, it is assumed that the current local swarm density  $\bar{x}_k[j]$  in Eq. (4) is accurately accessible by each agent in bin  $\mathcal{B}_j$ , for the ease of description. This can actually happen via a simple multi-hop communication over all the agents in  $\mathcal{A}_{\mathcal{N}_k(j)}$ . In order to reduce the required communication burden, we could utilise distributed density estimation methods in Bandyopadhyay and Chung (2014) and Demir et al. (2015) at the expense of a certain level of estimation error. Hence, we will numerically examine the effect of the uncertainty on the proposed framework, as will be shown in Sect. 6.4.

**Remark 3** It is worth repeating that, in the proposed approach, each agent in bin  $\mathcal{B}_j$  only relies on its local information about its neighbour bins  $\mathcal{N}_k(j)$ . This also applies to mission information such as the desired distribution  $\Theta$ . That is, the agent does not necessarily need to know the entire desired distribution a priori (which is the case in most of the existing works), but can obtain  $\bar{\Theta}[j]$  in a real-time manner during a mission as long as Assumption 2 holds. This is also the case for the motion constraint  $\mathbf{A}_k$  as long as motion constraints regarding neighbour bins are perceivable under Assumption 2 along with reasonably capable sensors.

### 3.3 The LICA-based Markov matrix

We present our methodology to generate a time-inhomogeneous Markov matrix  $M_k$  that achieves Desired Features 1–3 by using the local information feedback. The basic form of the stochastic policy for every agent in bin  $\mathcal{B}_j$  is such that

$$M_k[j, l] := (1 - \omega_k[j])P_k[j, l] + \omega_k[j]S_k[j, l], \quad \forall \mathcal{B}_l \in \mathcal{B}. \tag{8}$$

Here,  $\omega_k[j] \in [0, 1)$  is the weighing factor to have different weights on the *primary policy*  $P_k[j, l] \in \mathbb{P}$  and the *secondary policy*  $S_k[j, l] \in \mathbb{P}$ . The weighing factor is defined as

$$\omega_k[j] := \exp(-\lambda k) \cdot G_k[j] \tag{9}$$

where  $\lambda > 0$  is a design parameter that controls decay of  $\omega_k[j]$ ; and  $G_k[j] \in [0, 1]$  is the *secondary local-feedback gain*, which activates  $S_k[j, l]$  depending on the difference between  $\bar{x}_k[j]$  and  $\bar{\theta}[j]$ .

We introduced two policies  $P_k$  and  $S_k$  in order to help prospective users to have more design flexibility when implementing the framework into their own specific problems. As you will see, the asymptotic stability of agents towards  $\Theta$  is in fact guaranteed by  $P_k$  under the condition that the following Requirements 1–4 are satisfied. Since  $\omega_k[j]$  is diminishing as time instant  $k$  goes to infinity, users may adopt any temporary policies as  $S_k$  in addition to  $P_k$ , if necessary. For instance, when it is desired to disperse agents in  $\mathcal{B}_j$  into its neighbour bins more quickly if the bin is too overpopulated, it can happen by setting  $S_k[j, l] = 1/|\mathcal{N}_k(j)|$ ,  $\forall \mathcal{B}_l \in \mathcal{N}_k(j)$  and designing  $G_k[j]$  to be close to one when  $\bar{x}_k[j] \gg \bar{\theta}[j]$ , zero otherwise. Note that, in Sect. 4, we will provide explicit descriptions of  $P_k$ ,  $S_k$ , and  $G_k$ , which are varied depending on specific implementations.

Equation (8) can be represented in matrix form as

$$M_k = (I - W_k)P_k + W_kS_k, \tag{10}$$

where  $P_k \in \mathbb{P}^{n_b \times n_b}$  and  $S_k \in \mathbb{P}^{n_b \times n_b}$  are row-stochastic matrices, called *primary guidance matrix* and *secondary guidance matrix*, respectively.  $W_k \in \mathbb{R}^{n_b \times n_b}$  is a diagonal matrix such that  $W_k = \text{diag}(\omega_k[1], \dots, \omega_k[n_b])$ .

We claim that, in order for the Markov system to achieve all the desired Features,  $P_k$  must satisfy the following requirements.

**Requirement 1**  $P_k$  is a matrix with row sums equal to one, i.e.

$$\sum_{l=1}^{n_b} P_k[j, l] = 1, \quad \forall j. \tag{R1}$$

In fact,  $P_k$  needs to be row stochastic, for which it should further hold that  $P_k[j, l] \geq 0$ ,  $\forall j, l$ . Note that this constraint is implied by (R4), which will be introduced later.

**Requirement 2** All diagonal elements are positive, i.e.

$$P_k[j, j] > 0, \quad \forall j. \tag{R2}$$

**Requirement 3** The stationary distribution (i.e. equilibrium) of  $P_k$  is the desired swarm distribution  $\Theta$ , i.e.  $\Theta P_k = \Theta$  (or  $\sum_{j=1}^{n_b} \Theta[j]P_k[j, l] = \Theta[l], \forall l$ ). Along with (R1), this can be fulfilled by setting

$$\Theta[j]P_k[j, l] = \Theta[l]P_k[l, j], \quad \forall j, \forall l. \tag{R3}$$

A Markov process satisfying this property is said to be *reversible*.

**Requirement 4**  $P_k$  is irreducible such that

$$\begin{aligned} P_k[j, l] &> 0 \quad \text{if } \mathbf{A}_k[j, l] = 1, \\ P_k[j, l] &= 0 \quad \text{otherwise.} \end{aligned} \tag{R4}$$

Note that  $\mathbf{A}_k$  is also irreducible from Definition 2.

**Requirement 5**  $P_k$  becomes close to  $I$  as  $\bar{\mathbf{x}}_k$  converges to  $\bar{\Theta}$ , i.e.

$$P_k[j, j] \rightarrow 1 \text{ as } \bar{\mathbf{x}}_k[j] \rightarrow \bar{\Theta}[j] \text{ (or } \bar{\xi}_k[j] \rightarrow 0), \quad \forall j. \tag{R5}$$

Every agent in each bin  $\mathcal{B}_j$  executes the following subroutine to generate its stochastic policies at every time instant. Depending on missions,  $\bar{\xi}_k[j]$ ,  $P_k$ ,  $S_k$ , and  $G_k[j]$  can be designed differently under given specific constraints. (The detail regarding Lines 4–6 will be presented in Sect. 4, which shows examples of how to implement this framework.) As long as  $P_k$  holds (R1)–(R5) for every time instant  $k$ , the aforementioned desired features are achieved. Note that (R1)–(R4) are associated with Desired Feature 1, whereas (R5) is with Desired Features 2 and 3. The detailed analysis will be described in the next subsection.

---

**Algorithm 2** Generation of  $M_k[j, l], \forall l$  (Line 2 in Algorithm 1)

---

- // Obtain the local information*
  - 1: Compute  $\bar{\Theta}[j]$  using (5);
  - 2: Obtain  $\bar{\mathbf{x}}_k[j]$ ;
  - // Generate stochastic policies*
  - 3: Compute  $\bar{\xi}_k[j]$  [using (6)];
  - 4: Compute  $P_k[j, l], \forall l$ ;
  - 5: Compute  $S_k[j, l], \forall l$ ;
  - 6: Compute  $G_k[j]$ ;
  - 7: Compute  $\omega_k[j]$  using (9);
  - 8: Compute  $M_k[j, l]$  using (8),  $\forall l$ ;
- 

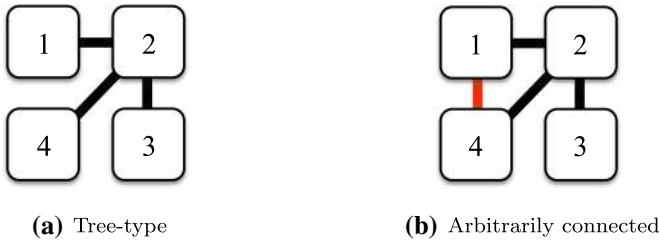
In a nutshell, our design guidelines are as follows:

- (i) Design  $\bar{\xi}_k[j]$  as a scalar function in  $(0, 1]$  that monotonically decreases as  $\bar{\mathbf{x}}_k[j]$  converges to  $\bar{\Theta}[j]$ , e.g. Eq. (6). Note that the shape of  $\bar{\xi}_k[j]$  is important so that it may cause high residual convergence error, as will be shown in Sect. 6.1.
- (ii) Design  $P_k[j, l]$  that satisfies (R1)–(R5) along with additional criteria from a given specific application.
- (iii) Design  $S_k[j, l]$  with consideration of the robotic swarm’s auxiliary but temporary behaviours that help the ultimate problem objective (if necessary).
- (iv) Design  $G_k[j]$  as a scalar function in  $[0, 1]$  in terms of  $\bar{\mathbf{x}}_k[j]$  and  $\bar{\Theta}[j]$  [e.g. Eqs. (18) or (28)], with consideration of when  $S_k$  is desired to be activated (if  $S_k$  is implemented).
- (v) Use  $M_k[j, l]$  and  $\omega_k[j]$  as shown in Eqs. (8) and (9), respectively.

We will apply the same guidelines when implementing the proposed framework into the specific examples in Sect. 4.

### 3.4 Analysis

We first show that the Markov process using Eq. (10) satisfies Desired Feature 1 under the assumption that  $P_k$  satisfies requirements (R1)–(R4) for every time instant. The swarm



**Fig. 3** Examples of simple bin topologies to help Lemmas 1 and 2: **a** tree-type; **b** arbitrarily connected. The red line in **(b)** indicates a newly added route between bin  $\mathcal{B}_1$  and  $\mathcal{B}_4$  based on the topology in **(a)** (Color figure online)

distribution at time instant  $k \geq k_0$ , governed by the Markov process from an arbitrary initial state  $\mathbf{x}_{k_0}$ , can be written as:

$$\mathbf{x}_k = \mathbf{x}_{k_0} U_{k_0,k} := \mathbf{x}_{k_0} M_{k_0} M_{k_0+1} \dots M_{k-1}. \tag{11}$$

**Theorem 1** *Provided that the requirements (R1)–(R4) are satisfied for all time instants  $k \geq k_0$ , it holds that  $\lim_{k \rightarrow \infty} \mathbf{x}_k = \Theta$  pointwise for all agents, irrespective of the initial condition.*

**Proof** Please refer to “Appendix A”. □

Theorem 1 implies that the ensemble of the agents eventually converges to the desired swarm distribution, regardless of  $S_k$ ,  $G_k[j]$ , and (R5). However, the system may induce unnecessary transitions of agents even after being close enough to  $\Theta$ , meaning that Desired Feature 2 does not hold yet.

Next, we argue that Desired Features 2 and 3 can also be obtained if requirement (R5) is additionally satisfied. Suppose that, for every bin  $\mathcal{B}_j$ ,  $\bar{\mathbf{x}}_k[j]$  converges to and eventually reaches  $\bar{\Theta}[j]$  at some time instant  $k$ . From Eqs. (4)–(5) and the supposition of  $\bar{\mathbf{x}}_k[j] = \bar{\Theta}[j]$ ,  $\forall j$ , it follows that  $1/\bar{\Theta}[j] \cdot \mathbf{n}_k[j] = \sum_{\forall \mathcal{B}_l \in \mathcal{N}_k(j)} \mathbf{n}_k[l]$ ,  $\forall j$ . This can be represented in matrix form as:

$$\mathbf{n}_k \cdot B := \mathbf{n}_k \cdot (\mathbf{A}_k - X) = \mathbf{0} \tag{12}$$

where  $X \in \mathbb{R}^{n_b \times n_b}$  is a diagonal matrix such that  $X = \text{diag}(1/\bar{\Theta}[1], \dots, 1/\bar{\Theta}[n_b])$ .

**Lemma 1** *Let the term tree-type (bidirectional) topology refer to as a graph such that any two vertices are connected by exactly one bidirectional path with no cycles (e.g. Fig. 3a). Given  $n_b$  bins connected as a tree-type topology, the rank of its corresponding matrix  $B$  in Eq. (12) is  $n_b - 1$ .*

**Proof** The matrix  $B \in \mathbb{R}^{n_b \times n_b}$  can be linearly decomposed into  $n_e$  of the same-sized matrices  $B_{(i,j)}$ , where  $n_e$  is the number of edges in the underlying graph of  $\mathbf{A}_k$ . Here,  $B_{(i,j)} \in \mathbb{R}^{n_b \times n_b}$  is a matrix such that  $B_{(i,j)}[i, i] = -\Theta[j]/\Theta[i]$  and  $B_{(i,j)}[j, j] = -\Theta[i]/\Theta[j]$ ;  $B_{(i,j)}[i, j] = B_{(i,j)}[j, i] = 1$ ; and all the other entries are zero. For example, consider that four bins are given and connected as shown in Fig. 3a. Clearly,  $B = B_{(1,2)} + B_{(2,3)} + B_{(2,4)}$ , where

$$B = \begin{pmatrix} -\frac{\Theta[2]}{\Theta[1]} & 1 & 0 & 0 \\ 1 & -\frac{\Theta[1]+\Theta[3]+\Theta[4]}{\Theta[2]} & 1 & 1 \\ 0 & 1 & -\frac{\Theta[2]}{\Theta[3]} & 0 \\ 0 & 1 & 0 & -\frac{\Theta[2]}{\Theta[4]} \end{pmatrix},$$

$$\begin{aligned}
 B_{(1,2)} &= \begin{pmatrix} -\frac{\vartheta[2]}{\vartheta[1]} & 1 & 0 & 0 \\ 1 & -\frac{\vartheta[1]}{\vartheta[2]} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \\
 B_{(2,3)} &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & -\frac{\vartheta[3]}{\vartheta[2]} & 1 & 0 \\ 0 & 1 & -\frac{\vartheta[2]}{\vartheta[3]} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \\
 B_{(2,4)} &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & -\frac{\vartheta[4]}{\vartheta[2]} & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -\frac{\vartheta[2]}{\vartheta[4]} \end{pmatrix}.
 \end{aligned}$$

It turns out that the rank of every  $B_{(i,j)}$  is one, and the matrix has only one linearly independent column vector, denoted by  $v_{(i,j)}$ . Without loss of generality, we consider  $v_{(i,j)} \in \mathbb{R}^{n_b}$  as a column vector such that the  $i$ th entry is  $-\frac{1}{\vartheta[i]}$ , the  $j$ th entry is  $\frac{1}{\vartheta[j]}$ , and the others are zero: for an instance,  $v_{(1,2)} = [-\frac{1}{\vartheta[1]}, \frac{1}{\vartheta[2]}, 0, 0]^T$ .

It follows that  $v_{(i,j)}$  and  $v_{(k,l)}$  are linearly independent when the bin pairs  $\{i, j\}$  and  $\{k, l\}$  are different. This implies that the number of linearly independent column vectors of  $B$  is the same as that of edges in the topology. Hence, for a tree-type topology of  $n_b$  bins, since there exist  $n_b - 1$  edges, the rank of the corresponding matrix  $B$  is  $n_b - 1$ . □

**Lemma 2** *Given a bidirectional and strongly connected topology of bins, the rank of its corresponding matrix  $B$  is not affected by adding a new bidirectional edge that connects any two existing bins.*

**Proof** We will show that this claim is valid even when a tree-type topology is given, as it is a sufficient condition for being bidirectional and strongly connected. Given the tree-type topology in Fig. 3a, suppose that bins  $B_1$  and  $B_4$  are newly connected. Then, the new topology becomes as shown in Fig. 3b, and it has a new corresponding matrix  $B_{\text{new}}$ , where  $B_{\text{new}} = B + B_{(1,4)}$ . As explained in the proof of Lemma 1, the rank of  $B_{(1,4)}$  is one and it has only a linearly independent vector  $v_{(1,4)}$ . However, this vector can be produced as a linear combination of the existing  $v$  vectors of  $B$  (i.e.  $v_{(1,4)} = v_{(1,2)} + v_{(2,4)}$ ). Thus, the rank of  $B_{\text{new}}$  retains that of  $B$ . Without loss of generality, this implies that the rank of  $B$  of a given bidirectional and strongly connected topology is not affected by adding a new edge that directly connects any two existing bins. □

Thanks to Lemmas 1 and 2, we end up with the following corollary and lemma:

**Corollary 1** *Given  $n_b$  bins that are at least bidirectional and strongly connected, the rank of its corresponding matrix  $B$  is  $n_b - 1$ .*

**Lemma 3** *Given  $n_b$  bins that are at least bidirectional and strongly connected, convergence of  $\bar{\mathbf{x}}_\infty$  to  $\bar{\Theta}$  is equivalent to convergence of  $\mathbf{x}_\infty$  to  $\Theta$ .*

**Proof** Assuming that  $\lim_{k \rightarrow \infty} \bar{\mathbf{x}}_k = \bar{\Theta}$ , it can be said that  $\lim_{k \rightarrow \infty} \mathbf{n}_k \cdot B = \mathbf{0}$ , as similar to the derivation of Eq. (12). From Eq. (5), it turns out that

$$\Theta \cdot B = \mathbf{0}. \tag{13}$$

Since the nullity of  $B$  is one due to Corollary 1, there exists only one linearly independent row vector  $\mathbf{a} \in \mathbb{R}^{n_b}$  such that  $\mathbf{a} \cdot B = \mathbf{0}$ . Hence, it follows that  $\lim_{k \rightarrow \infty} \mathbf{n}_k = \epsilon \Theta$ , where  $\epsilon$  is an arbitrary scalar value. This also implies that  $\lim_{k \rightarrow \infty} \mathbf{x}_k = \lim_{k \rightarrow \infty} \mathbf{n}_k/n_a = \Theta$ .

On the other hand, suppose that  $\lim_{k \rightarrow \infty} \mathbf{x}_k = \Theta$ , which can be rewritten as  $\lim_{k \rightarrow \infty} \mathbf{n}_k = n_a \Theta$ . By right multiplying  $B$  for both sides (i.e.  $\lim_{k \rightarrow \infty} \mathbf{n}_k \cdot B = n_a(\Theta \cdot B)$ ), it follows from Eq. (13) that  $\lim_{k \rightarrow \infty} \mathbf{n}_k \cdot B = \mathbf{0}$ . By the definition of  $B$ , it can be rearranged as  $\lim_{k \rightarrow \infty} \bar{\mathbf{x}}_k = \bar{\Theta}$ .

Therefore, convergence of  $\bar{\mathbf{x}}_\infty$  to  $\bar{\Theta}$  is equivalent to convergence of  $\mathbf{x}_\infty$  to  $\Theta$ . □

From this lemma and (R5), Desired Feature 2 finally holds as follows.

**Theorem 2** *If  $P_k$  satisfies (R1)–(R5) for all time instants  $k \geq k_0$ , the Markov process using  $M_k$  in Eq. (10) satisfies Desired Feature 2 as well as Desired Feature 1.*

**Proof** It was shown from Theorem 1 that requirements (R1)–(R4) guarantee the convergence of  $\mathbf{x}_\infty$  to  $\Theta$  (i.e. Desired Feature 1). From this and Lemma 3,  $\bar{\mathbf{x}}_\infty$  also converges to  $\bar{\Theta}$ . If  $P_k$  additionally complies with (R5), then  $P_k$  becomes close to  $I$  as  $k \rightarrow \infty$ . This is also the case for the Markov process  $M_k$ , which satisfies Desired Feature 2. □

**Corollary 2** *In order for every agent in bin  $\mathcal{B}_j$  to generate  $M_k[j, l], \forall l$  in Eq. (8), the agent only needs local information within  $\mathcal{A}_{\mathcal{N}_k(j)}$ . Therefore, Desired Feature 3 is also achieved.*

**Remark 4** (Robustness against dynamic changes of agents or bins) The proposed framework is robust against dynamic changes in the number of agents or bins. As each agent behaves based on its current bin location and local information in a memoryless manner, Desired Features 1–3 in the proposed framework will not be affected by inclusion or exclusion of agents in a swarm.

Besides, as long as changes on bins are perceived by nearby agents in the corresponding neighbour bins, robustness against those changes also holds in the proposed framework. This is because agents in bin  $\mathcal{B}_j$  utilise only local information such as  $\bar{\Theta}[j]$  and  $\bar{\mathbf{x}}_k[j]$ , and are not required to know information from other far-away bins. Moreover, the proposed framework does not need to recalculate  $\Theta$  (which has to be normalised in a GICA-based framework such that  $\sum_{\forall j} \Theta[j] = 1$  after reflecting such changes) because computing  $\bar{\Theta}[j]$  in Eq. (5) already includes a sense of normalisation based on local information.

## 4 Implementation examples

### 4.1 Example I: minimising travelling expenses

This section provides implementation examples of the proposed framework. In particular, this subsection addresses a problem of minimising travelling expenses of agents during convergence to a desired swarm distribution, as shown in Bandyopadhyay et al. (2017). The problem can be defined as follows:

**Problem 1** Given a cost matrix  $E_k \in \mathbb{R}^{n_b \times n_b}$  in which each element  $E_k[j, l]$  represents the travelling expense of an agent from bin  $\mathcal{B}_j$  to  $\mathcal{B}_l$ , find  $P_k$  such that

$$\min \sum_{j=1}^{n_b} \sum_{l=1}^{n_b} E_k[j, l] P_k[j, l], \tag{P1}$$

subject to (R1)–(R5) and

$$\Theta[l]f_{\xi}(\bar{\xi}_k[j], \bar{\xi}_k[l])f_E(E_k[j, l]) \leq P_k[j, l] \text{ if } \mathbf{A}_k[j, l] = 1, \quad \forall j \neq l, \quad (14)$$

where  $\Theta[l]$  enables agents in bin  $\mathcal{B}_j$  to be distributed over its neighbour bins in proportion to the desired swarm distribution;  $f_{\xi}(\bar{\xi}_k[j], \bar{\xi}_k[l]) \in (0, 1]$  and  $f_E(E_k[j, l]) \in (0, 1]$  control the lower bound of  $P_k[j, l]$  in Eq. (14), depending on the primary local-feedback gains and travelling expenses, respectively. Specifically, it is set that

$$f_{\xi}(\bar{\xi}_k[j], \bar{\xi}_k[l]) = \max(\bar{\xi}_k[j], \bar{\xi}_k[l]) \quad (15)$$

so that the value monotonically increases with regard to increase of either  $\bar{\xi}_k[j]$  or  $\bar{\xi}_k[l]$  and diminishes as  $\bar{\xi}_k[j]$  and  $\bar{\xi}_k[l]$  simultaneously reduce, meaning that it allows a larger number of transitioning agents between the two bins  $\mathcal{B}_j$  and  $\mathcal{B}_l$  when any one of them needs to be regulated.  $f_E(E_k[j, l]) \in (0, 1]$  monotonically decreases as  $E_k[j, l]$  increases [see Eq. (30) for an example of its explicit definition], preventing agents in bin  $\mathcal{B}_j$  from spending higher transition expenses. We assume that  $E_k$  is symmetric;  $E_k[j, l] > 0$  if  $\mathbf{A}_k[j, l] = 1$ ; and its diagonal entries are zero.

**Corollary 3** *The optimal matrix  $P_k$  of problem (P1) is given by:  $\forall j, l \in \{1, \dots, n_b\}$  and  $l \neq j$ ,*

$$P_k[j, l] = \begin{cases} \Theta[l]f_{\xi}(\bar{\xi}_k[j], \bar{\xi}_k[l])f_E(E_k[j, l]) & \text{if } \mathbf{A}_k[j, l] = 1 \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

and  $\forall j$ ,

$$P_k[j, j] = 1 - \sum_{\forall l \neq j} P_k[j, l]. \quad (17)$$

**Proof** Please refer to “Appendix B”. □

To reduce unnecessary transitions of agents during this process, it is desirable that agents in bin  $\mathcal{B}_j$  such that  $\bar{\mathbf{x}}_k[j] \leq \bar{\Theta}[j]$  (i.e. underpopulated) do not deviate. To this end, we set  $S_k = I$  and  $G_k[j]$  as follows:

$$G_k[j] := \frac{\exp(\beta(\bar{\Theta}[j] - \bar{\mathbf{x}}_k[j]))}{\exp(\beta|\bar{\Theta}[j] - \bar{\mathbf{x}}_k[j])}. \quad (18)$$

The gain value is depicted in Fig. 4a with regard to the sensitivity parameter  $\beta$ , which controls the steepness of  $G_k[j]$  at around when  $\bar{\mathbf{x}}_k[j] - \bar{\Theta}[j]$  is close to zero but positive. For example, at a lower  $\beta$ , a relatively higher number of agents tend to follow the secondary guidance matrix  $S_k$  (i.e. not to deviate) rather than  $P_k$  even when  $\bar{\mathbf{x}}_k[j] - \bar{\Theta}[j] > 0$  is not much close to zero.

**Remark 5** (Increase of Convergence Rate) Due to the fact that  $\sum_{\forall l \neq j} P_k[j, l] \leq \sum_{\forall \mathcal{B}_l \in \mathcal{N}_k(j) \setminus \{\mathcal{B}_j\}} \Theta[l]$  from Eq. (16), the total outflux of agents from bin  $\mathcal{B}_j$  becomes smaller as the bin has fewer connections with other bins. This eventually makes the convergence rate of the Markov process slower.

Adding an additional variable into  $P_k[j, l]$  in (16) does not affect the obtainment of Desired Features 1–3 as long as  $P_k$  satisfies (R1)–(R5). Thus, in order to enhance the convergence rate under the requirements, one can add

$$\epsilon_{\Theta} := \min \left\{ \frac{1}{\sum_{\forall s: \mathcal{B}_s \in \mathcal{N}_k(j) \setminus \{\mathcal{B}_j\}} \Theta[s]}, \frac{1}{\sum_{\forall s: \mathcal{B}_s \in \mathcal{N}_k(l) \setminus \{\mathcal{B}_l\}} \Theta[s]} \right\} \quad (19)$$



**Algorithm 3** Minimising Travelling Expenses (Lines 4–6 of Algorithm 2 for P1)

- 1: Compute  $P_k[j, l] \forall l$  using (16) [or (20)] and (17)
- 2: Set  $S_k[j, j] = 1; S_k[j, l] = 0, \forall l \neq j$
- 3: Compute  $G_k[j]$  using (18)

into  $P_k[j, l]$ , as follows:

$$P_k[j, l] = \begin{cases} \epsilon_{\Theta} \Theta[l] f_{\xi}(\bar{\xi}_k[j], \bar{\xi}_k[l]) f_E(E_k[j, l]) & \text{if } \mathbf{A}_k[j, l] = 1 \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

which can be substituted for Eq. (16).

**4.2 Example II: maximising convergence rate within flux upper limits**

This subsection presents an example in which the specific objective is to maximise the convergence rate under upper bounds regarding transition of agents between bins, referred to as *flux upper limits*. The bounds can be interpreted as safety constraints in terms of collision avoidance and congestion: higher congestions may induce higher collisions amongst agents, which may bring unfavourable effects on system performance. A similar problem is addressed by an open-loop algorithm in Berman et al. (2009), where transitions of agents are limited only after a desired swarm distribution is achieved. This restriction is not for considering the aforementioned safety constraints, but rather for mitigating the trade-off between convergence rate and long-term system efficiency.

For the sake of imposing flux upper limits *during the entire process*, we consider the following one-way flux constraint: for every time instant  $k$ ,

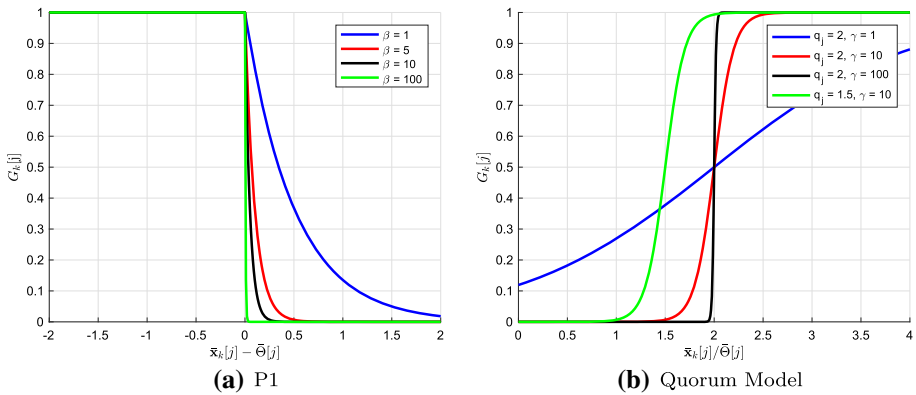
$$\mathbf{n}_k[j] P_k[j, l] \leq c_{(j,l)}, \quad \forall j, \quad \forall l \neq j. \tag{21}$$

This means that the number of agents moving from bin  $\mathcal{B}_j$  to  $\mathcal{B}_l$  is upper-bounded by  $c_{(j,l)}$ . The bound value is assumed to be very small with consideration of mission environments such as the number of agents, the number of bins, and their topology. Otherwise, all the agents can be distributed over the bins very soon so that the flux upper limits become meaningless and the corresponding problem can be trivial. Please note that the flux limits in this example should be considered as expected constraints. In the case where hard constraints are to be accommodated in practice, it is necessary to set a tighter value with consideration of a margin from the actual value. The level of the margin would be affected by the number of agents involved in the framework, as will be shown in Fig. 9c later.

Regarding the convergence rate of a Markov chain, there are respective analytical methods depending on whether it is time-homogeneous or time-inhomogeneous. For a time-homogeneous Markov chain, if the matrix is irreducible, the second largest eigenvalue of the matrix is used as an index indicating its asymptotic convergence rate (Bestaoui Sebbane 2014, p. 389). On the contrary, for a time-inhomogeneous Markov chain, *coefficients of ergodicity* can be utilised as a substitute for the second largest eigenvalue, which is not useful for this case (Ipsen and Selee 2011). Particularly, this paper uses the following *proper* coefficient of ergodicity, amongst others:

**Definition 9** [*Coefficient of Ergodicity* (Seneta 1981, pp. 136–137)] Given a stochastic matrix  $\mathcal{M} \in \mathbb{P}^{n \times n}$ , a (proper) coefficient of ergodicity  $0 \leq \tau(\mathcal{M}) \leq 1$  can be defined as:

$$\tau(\mathcal{M}) := \max_{\forall s} \max_{\forall j, \forall l} |\mathcal{M}[j, s] - \mathcal{M}[l, s]|. \tag{22}$$



**Fig. 4** The secondary feedback gains  $G_k[j]$  depending on the associated design parameters: **a** for P1 [i.e. Eq. (18)]; **b** for the quorum model [i.e. Eq. (28)]

A coefficient of ergodicity is said to be *proper* if  $\tau(\mathcal{M}) = 0$  is necessary and sufficient for  $\mathcal{M} = \mathbf{1}^T \cdot v$ , where  $v \in \mathbb{P}^n$  is a row-stochastic vector.

The convergence rate of a time-inhomogeneous Markov chain  $\mathcal{M}_k \in \mathbb{P}^{n \times n}, \forall k \geq 1$  can be maximised by minimising  $\tau(\mathcal{M}_k)$  at each time instant  $k$ , thanks to Seneta (1981, Theorem 4.8, p. 137):  $\tau(\mathcal{M}_1 \mathcal{M}_2 \dots \mathcal{M}_r) \leq \prod_{k=1}^r \tau(\mathcal{M}_k)$ . Hence, the objective of the specific problem considered in this subsection can be defined as: find  $P_k$  such that

$$\min \tau(P_k) \tag{23}$$

subject to (R1)–(R5) and (21).

**Remark 6** [Advantages of the coefficient of ergodicity in (22)] Other proper coefficients in Seneta (1981, p. 137) such as

$$\tau_1(\mathcal{M}) = 1 - \min_{\forall j, \forall l} \sum_{\forall s} \min\{\mathcal{M}[j, s], \mathcal{M}[l, s]\}$$

or

$$\tau_2(\mathcal{M}) = 1 - \sum_{\forall j} \min\{\mathcal{M}[j, s]\}$$

may have the trivial case such that  $\tau_1(P_k) = 1$  (or  $\tau_2(P_k) = 1$ ) for some time instant  $k$ , when they are applied to this problem. For example, given a topology of bins  $\mathbf{A}_k$ , there may exist a pair of bins  $\mathcal{B}_j$  and  $\mathcal{B}_l$  such that  $P_k[j, s] = 0$  or  $P_k[l, s] = 0, \forall s$ . To avoid this trivial case, Bandyopadhyay et al. (2017) instead utilises  $\tau_1((P_k)^{d_{\mathbf{A}_k}})$  as the proper coefficient of ergodicity, where  $d_{\mathbf{A}_k}$  denotes the diameter of the underlying graph of  $\mathbf{A}_k$ . However, this implies that agents in bin  $\mathcal{B}_j$  are required to additionally access the information from other bins beside  $\mathcal{N}_k(j)$ , causing additional communicational costs. The coefficient of ergodicity in (22) does not suffer from this issue. Note that  $\tau(\mathcal{M}) \leq \tau_1(\mathcal{M}) \leq \tau_2(\mathcal{M})$  (Seneta 1981, p. 137).

Finding the optimal solution for problem (23) is another challenging issue, called *fastest mixing Markov chain problem*. Since the purpose of this section is to show an example of how to implement our proposed framework, we heuristically address this problem at this moment.

Suppose that a matrix  $P_k$  satisfying (R1)–(R5) is given and that the topology of bins is at least connected without any bin being connected to all the others. Since the matrix is non-negative and there exists at least one zero-value entry in each column, the coefficient of ergodicity can be said as  $\tau(P_k) = \max_{\forall s, \forall j} (P_k[j, s])$ . Assuming that  $\max_{\forall l \neq j} P_k[j, l] \leq 1/|\mathcal{N}_k(j)|$ , which is generally true due to the small values of  $c_{(j,l)}$ , it turns out that each diagonal element of  $P_k$  is the largest value in each row. Thus, we can say that  $\tau(P_k) = \max_{\forall j} P_k[j, j]$ . Eventually, the objective function in Eq. (23) can be rewritten as

$$\max \min_{\forall j} \sum_{\forall l \neq j} P_k[j, l]$$

because minimising the maximum diagonal element of a stochastic matrix is equivalent to maximising the minimum row sum of its off-diagonal elements.

We turn now to constraints (R1)–(R5) and (21). In order to comply with (R3), we initially set  $P_k[j, l] = \Theta[l]Q_k[j, l]$ , where  $Q_k$  is a symmetric matrix that we will design now. Constraint (21), (R4), and the symmetricity of  $Q_k$  are necessary conditions for the following constraint:  $\forall j, \forall l \neq j$ ,

$$\min \left( \frac{c_{(j,l)}}{\mathbf{n}_k[j]\Theta[l]}, \frac{c_{(l,j)}}{\mathbf{n}_k[l]\Theta[j]} \right) \geq Q_k[j, l] > 0 \quad \text{if } \mathbf{A}_k[j, l] = 1 \tag{24}$$

$$Q_k[j, l] = 0 \quad \text{otherwise.}$$

For (R2) and (R5), we set the diagonal entries of  $P_k$  as

$$P_k[j, j] \geq 1 - \bar{\xi}_k[j], \quad \forall j. \tag{25}$$

Note that the non-strict inequality is not troublesome to (R2) because  $\bar{\xi}_k[j] = 1$  only when  $\bar{\mathbf{x}}_k[j] = 0$ , in which there exists no agent in bin  $\mathcal{B}_j$ , and thus effectively  $P_k[j, j] = 1$ . Equation (25) can be rewritten, with consideration of (R1) (i.e.  $\sum_{l=1}^{n_b} \Theta[l]Q_k[j, l] = 1, \forall j$ ), as

$$\sum_{\forall l \neq j} \Theta[l]Q_k[j, l] \leq \bar{\xi}_k[j], \quad \forall j. \tag{26}$$

In summary, Eq. (24) is a sufficient condition for (R3), (R4), and (21); and Eq. (26) is for (R1), (R2), and (R5). Hence, the reduced problem can be defined as:

**Problem 2** Find  $Q_k$  such that

$$\max \min_{\forall j} \sum_{\forall l \neq j} \Theta[l]Q_k[j, l] \tag{P2}$$

subject to (24) and (26).

The algorithm for problem (P2) is shown in Algorithm 4. If we neglect (26), an optimal solution can be obtained by making  $Q_k[j, l]$  equal to its upper bound in (24) (Line 2). However, this solution may not satisfy (26). Thus, we lower the entries of  $Q_k$  to satisfy (26), while keeping them symmetric and as high as possible (Lines 3–8). In detail, Line 3 (or Line 6) ensures constraint (26) for each bin  $\mathcal{B}_j$  in a way that, if this is not the case, obtains the necessary lowering factor  $\bar{\epsilon}'_Q[j]$  (or  $\bar{\epsilon}_Q[j]$ ). In order to keep  $Q_k$  as high as possible, we temporarily take  $\epsilon'_Q[j, l]$  as the maximum value of  $\{\bar{\epsilon}'_Q[j], \bar{\epsilon}'_Q[l]\}$  (Line 4). After curtailing  $Q_k[j, l]$  by applying  $\epsilon'_Q[j, l]$ , we obtain the corresponding lowering factor again (Lines 5–6). For now, the minimum value is taken to maintain  $Q_k$ 's symmetricity and satisfy (26)

**Algorithm 4** Max Convergence with Flux Limits (Line 4 of Algorithm 2 for P2)

```

// Initialise  $P_k$ 
1:  $P_k[j, l] = 0, \forall l \in \{1, 2, \dots, n_b\}$ ;
// Compute  $Q_k$  satisfying constraint (24) only
2:  $Q_k[j, l] = \min\left(\frac{c(j,l)}{\mathbf{n}_k[j]\Theta[l]}, \frac{c(l,j)}{\mathbf{n}_k[l]\Theta[j]}\right), \forall \mathcal{B}_l \in \mathcal{N}_k(j) \setminus \{\mathcal{B}_j\}$ ;
// Lower  $Q_k$  to satisfy constraint (26) additionally
3:  $\bar{\epsilon}'_Q[j] = \min\left(\frac{\bar{\epsilon}_k[j]}{\sum_{\forall l \neq j} \Theta[l]Q_k[j, l]}, 1\right)$ ;
4:  $\epsilon'_Q[j, l] = \max(\bar{\epsilon}'_Q[j], \bar{\epsilon}'_Q[l]), \forall \mathcal{B}_l \in \mathcal{N}_k(j) \setminus \{\mathcal{B}_j\}$ ;
5:  $Q_k[j, l] := \epsilon'_Q[j, l]Q_k[j, l], \forall \mathcal{B}_l \in \mathcal{N}_k(j) \setminus \{\mathcal{B}_j\}$ ;
6:  $\bar{\epsilon}_Q[j] = \min\left(\frac{\bar{\epsilon}_k[j]}{\sum_{\forall l \neq j} \Theta[l]Q_k[j, l]}, 1\right)$ ;
7:  $\epsilon_Q[j, l] = \min(\bar{\epsilon}_Q[j], \bar{\epsilon}_Q[l]), \forall \mathcal{B}_l \in \mathcal{N}_k(j) \setminus \{\mathcal{B}_j\}$ ;
8:  $Q_k[j, l] := \epsilon_Q[j, l]Q_k[j, l], \forall \mathcal{B}_l \in \mathcal{N}_k(j) \setminus \{\mathcal{B}_j\}$ ;
// Compute  $P_k$ 
9:  $P_k[j, l] = \Theta[l]Q_k[j, l], \forall \mathcal{B}_l \in \mathcal{N}_k(j) \setminus \{\mathcal{B}_j\}$ ;
10:  $P_k[j, j] = 1 - \sum_{\forall l \neq j} P_k[j, l]$ ;
    
```

simultaneously (Line 7). Then, the corresponding stochastic policy is generated based on the resultant  $Q_k$  (Lines 8–10).

Note that we set  $G_k[j] = 0$  for all time instants and all bins, so  $M_k = P_k$ .

**4.3 Example III: local information-based quorum model**

This subsection shows that the proposed framework is able to incorporate a quorum model, which is introduced in Halasz et al. (2007) and Hsieh et al. (2008). In this model, if a bin is overpopulated above a certain level of predefined threshold called *quorum*, the probabilities that agents in the bin move to neighbour bins are temporarily increased, rather than consistently following given  $P_k$ . This feature eventually brings an advantage to the convergence performance of the swarm.

To this end, we set the secondary guidance matrix  $S_k$  as follows:  $\forall j, l \in \{1, \dots, n_b\}$ ,

$$S_k[j, l] := \begin{cases} 1/|\mathcal{N}_k(j)| & \text{if } \mathbf{A}_k[j, l] = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{27}$$

This matrix makes agents in a bin equally disseminated over its neighbour bins. In addition, the secondary feedback gain  $G_k[j]$  is defined as

$$G_k[j] := \left(1 + \exp\left(\gamma\left(q_j - \frac{\bar{\mathbf{x}}_k[j]}{\Theta[j]}\right)\right)\right)^{-1}, \tag{28}$$

where  $\gamma > 0$  is a design parameter, and  $q_j > 1$  is the quorum for bin  $\mathcal{B}_j$ . Figure 4b shows the gain value varying depending on  $\gamma$  and  $q_j$ . As  $\bar{\mathbf{x}}_k[j]/\Theta[j]$  becomes higher than the quorum,  $G_k[j]$  gets close to 1 (i.e.  $S_k[j, l]$  becomes more dominant than  $P_k[j, l]$ ). The steepness of  $G_k[j]$  around the quorum value is regulated by  $\gamma$ .

The existing quorum models in Halasz et al. (2007) and Hsieh et al. (2008) require each agent to know  $\mathbf{x}_k[j]$ , which implies that the total number of agents  $n_a$  should be tracked in real time. It could be possible that some agents in a swarm unexpectedly become faulted by internal or external effects during a mission, which hinders for other alive agents from keeping track of  $n_a$  in a timely manner. On the contrary, this requirement is not the case for

---

**Algorithm 5** The Quorum Model (Lines 5–6 of Algorithm 2)

---

- 1: Compute  $S_k[j, l]$  using (27),  $\forall l$ ;
  - 2: Compute  $G_k[j]$  using (28);
- 

the proposed quorum model, and it works by only using local information available from  $\mathcal{A}_{\mathcal{N}_k(j)}$ .

### 5 Asynchronous implementation

A synchronous process induces extra time delays and inter-agent communication to make the entire agents, each of which may have different timescales for obtaining new information, remain in sync. Such unnecessary waiting time and communication may cause unfavourable effects on mission performance or may not even be realisable in practice (Johnson et al. 2011).

In the previous sections, it was assumed that a swarm of agents act synchronously at every time instant. Here, we show that the proposed framework can accommodate asynchronous behaviours of the agents, assuming that the union of underlying graphs of the corresponding Markov matrices across some time intervals is frequently and infinitely strongly connected.

Suppose that an algorithm to compute  $P_k$  that satisfies (R1)–(R5) in a synchronous environment is given (e.g. Algorithm 3 or 4). We propose an asynchronous implementation as shown in Algorithm 6, which substitutes Line 4 in Algorithm 2. We refer to a set of bins where agents are ready to use their respective local information (e.g.  $\mathbf{n}_k[j]$ ) as  $\mathcal{R}_k^+$ , and a set of the other bins as  $\mathcal{R}_k^-$ . For each bin  $\mathcal{B}_j$ , we denote  $\mathcal{N}_k^+(j) := \mathcal{N}_k(j) \cap \mathcal{R}_k^+$  and  $\mathcal{N}_k^-(j) := \mathcal{N}_k(j) \cap \mathcal{R}_k^-$ . It is assumed that each agent in bin  $\mathcal{B}_j \in \mathcal{R}_k^+$  knows the local information of its neighbour bin  $\mathcal{B}_l \in \mathcal{N}_k^+(j)$ .

In the asynchronous algorithm, each agent in bin  $\mathcal{B}_j \in \mathcal{R}_k^+$  follows the existing procedure of generating  $P_k[j, l]$  for each neighbour bin  $\mathcal{B}_l$  whose local information is also available (Line 2). Then, the probabilities to transition to all the other bins (except  $\mathcal{B}_j$ ) are set to be zero (Line 3). In the meantime, each agent for whom local information is not ready does not deviate but remains at the bin it belongs to. Equivalently, it can be said that  $P_k[j, j] = 1$  and  $P_k[j, l] = 0, \forall l \neq j$  (Line 6).

Hereafter, for the sake of differentiation from the original  $P_k$  generated in a synchronous environment, let us refer to the matrix resultant from Algorithm 6 as *asynchronous primary guidance matrix*, denoted by  $\tilde{P}_k$ . Accordingly, the *asynchronous Markov matrix* can be defined as:

$$\tilde{M}_k := (I - W_k)\tilde{P}_k + W_k S_k.$$

---

**Algorithm 6** Asynchronous Construction of  $P_k[j, l]$  (Substitute for Line 4 of Algorithm 2)

---

- 1: **if**  $\mathcal{B}_j \in \mathcal{R}_k^+$  & **isnonempty**( $\mathcal{N}_k^+(j) \setminus \{\mathcal{B}_j\}$ ) **then**
  - 2:   Compute  $P_k[j, l]$  as usual,  $\forall \mathcal{B}_l \in \mathcal{N}_k^+(j) \setminus \{\mathcal{B}_j\}$
  - 3:    $P_k[j, l] = 0, \forall \mathcal{B}_l \in \mathcal{B} \setminus \mathcal{N}_k^+(j)$
  - 4:    $P_k[j, j] = 1 - \sum_{\forall l \neq j} P_k[j, l]$
  - 5: **else**
  - 6:    $P_k[j, j] = 1; P_k[j, l] = 0, \forall l \neq j$
  - 7: **end if**
-

We show that this asynchronous Markov process also converges to the desired swarm distribution.

**Lemma 4** *The matrix  $\tilde{P}_k$ , for every time instant  $k$ , satisfies the following properties: (1) row stochastic; (2) all diagonal elements are positive and all other elements are non-negative; and (3)  $\sum_{l=1}^{n_b} \Theta[l] \tilde{P}_k[l, j] = \Theta[j], \forall j$ .*

**Proof** The matrix  $\tilde{P}_k$  is row stochastic because of Lines 4 and 6 in Algorithm 6. Furthermore, given that  $P_k$  satisfies (R2), property (2) is also valid for  $\tilde{P}_k$  because  $\tilde{P}_k[j, j] \geq P_k[j, j]$  for  $\forall j$ .

Let us now turn to the property (3), and firstly consider the case where  $\forall \mathcal{B}_j \in \mathcal{R}_k^+$ . For any two bins  $\mathcal{B}_{j_1}$  and  $\mathcal{B}_{j_2}$  ( $j_1 \neq j_2$ ), Algorithm 6 yields that  $\tilde{P}_k[j_1, j_2] = P_k[j_1, j_2]$  and  $\tilde{P}_k[j_2, j_1] = P_k[j_2, j_1]$  if  $\mathcal{B}_{j_1}, \mathcal{B}_{j_2} \in \mathcal{R}_k^+$  and  $\mathbf{A}_k[j_1, j_2] = 1$  simultaneously, otherwise  $\tilde{P}_k[j_1, j_2] = \tilde{P}_k[j_2, j_1] = 0$ . For  $\forall \mathcal{B}_j \in \mathcal{R}_k^+$ , this fact implies the following: (i)  $\tilde{P}_k[l, j] = P_k[l, j]$  for  $\forall \mathcal{B}_l \in \mathcal{N}_k^+(j) \setminus \{\mathcal{B}_j\}$ ; (ii)  $\tilde{P}_k[j, l] = \tilde{P}_k[l, j] = 0$  for  $\forall \mathcal{B}_l \in \mathcal{B} \setminus \mathcal{N}_k^+(j)$ ; and (iii)  $\tilde{P}_k[j, j] = P_k[j, j] + \sum_{\forall \mathcal{B}_l \in \mathcal{N}_k^-(j)} P_k[j, l]$ . We apply the findings into the following equation:

$$\begin{aligned} \sum_{l=1}^{n_b} \Theta[l] \tilde{P}_k[l, j] &= \sum_{\forall \mathcal{B}_l \in \mathcal{B} \setminus \mathcal{N}_k^+(j)} \Theta[l] \tilde{P}_k[l, j] \\ &+ \sum_{\forall \mathcal{B}_l \in \mathcal{N}_k^+(j) \setminus \{\mathcal{B}_j\}} \Theta[l] \tilde{P}_k[l, j] + \Theta[j] \tilde{P}_k[j, j]. \end{aligned} \tag{29}$$

The first term of the right hand side becomes zero because of (ii). Due to (i) and the fact that  $\Theta[l] P_k[l, j] = \Theta[j] P_k[j, l] \forall l$  (from Requirement 3), the second term becomes  $\Theta[j] \sum_{\forall \mathcal{B}_l \in \mathcal{N}_k^+(j) \setminus \{\mathcal{B}_j\}} P_k[j, l]$ . The last term becomes  $\Theta[j] P_k[j, j] + \Theta[j] \sum_{\forall \mathcal{B}_l \in \mathcal{N}_k^-(j)} P_k[j, l]$  because of (iii). By putting all of them together and adding  $\sum_{\forall \mathcal{B}_l \in \mathcal{B} \setminus \mathcal{N}_k(j)} P_k[j, l] = 0$ , the right hand side of Eq. (29) is equivalent to

$$\begin{aligned} &\Theta[j] \left( \sum_{\forall \mathcal{B}_l \in \mathcal{N}_k^+(j) \setminus \{\mathcal{B}_j\}} P_k[j, l] + P_k[j, j] + \sum_{\forall \mathcal{B}_l \in \mathcal{N}_k^-(j)} P_k[j, l] + \sum_{\forall \mathcal{B}_l \in \mathcal{B} \setminus \mathcal{N}_k(j)} P_k[j, l] \right) \\ &= \Theta[j] \sum_{l=1}^{n_b} P_k[j, l] = \Theta[j]. \end{aligned}$$

On the other hand, for the case where  $\forall \mathcal{B}_j \in \mathcal{R}_k^-$ , it follows from Algorithm 6 that  $\tilde{P}_k[l, j] = 0, \forall l \neq j$  and  $\tilde{P}_k[j, j] = 1$ . Thus,  $\sum_{l=1}^{n_b} \Theta[l] \tilde{P}_k[l, j] = \Theta[j]$ . □

**Lemma 5** *If the union of a set of underlying graphs of  $\{\tilde{P}_{k_1}, \tilde{P}_{k_1+1}, \dots, \tilde{P}_{k_2-1}\}$  is strongly connected, then the matrix product  $\tilde{P}_{k_1, k_2} := \tilde{P}_{k_1} \tilde{P}_{k_1+1} \dots \tilde{P}_{k_2-1}$  is irreducible.*

**Proof** Since the union of a set of underlying graphs of  $\{\tilde{P}_{k_1}, \tilde{P}_{k_1+1}, \dots, \tilde{P}_{k_2-1}\}$  is strongly connected, the underlying graph of  $\sum_{k=k_1}^{k_2-1} \tilde{P}_k$  is also strongly connected. Noting that every  $\tilde{P}_k, \forall k \in \{k_1, k_1 + 1, \dots, k_2 - 1\}$  is a non-negative  $n_b \times n_b$  matrix and its diagonal elements are positive (by Lemma 4), it follows from (Jadbabaie et al. 2003, Lemma 2) that  $\tilde{P}_{k_1, k_2} \geq \gamma \sum_{k=k_1}^{k_2-1} \tilde{P}_k$ , where  $\gamma > 0$ . This implies that the underlying graph of  $\tilde{P}_{k_1, k_2}$  is strongly connected, and thus the matrix  $\tilde{P}_{k_1, k_2}$  is irreducible. □

**Theorem 3** *Suppose that there exists an infinite sequence of non-overlapping time intervals  $[k_i, k_{i+1})$ ,  $i = 0, 1, 2, \dots$ , such that the union of underlying graphs of  $\{\tilde{P}_{k_i}, \tilde{P}_{k_{i+1}}, \dots, \tilde{P}_{k_{i+1}-1}\}$  in each interval is strongly connected. Let the swarm distribution at time instant  $k \geq k_0$ , governed by the corresponding Markov process from an arbitrary state  $\mathbf{x}_{k_0}$ , be  $\mathbf{x}_k = \mathbf{x}_{k_0} \tilde{U}_{k_0,k} := \mathbf{x}_{k_0} \tilde{M}_{k_0} \tilde{M}_{k_0+1} \dots \tilde{M}_{k-1}$ . Then, it holds that  $\lim_{k \rightarrow \infty} \mathbf{x}_k = \Theta$  pointwise for all agents, irrespective of the initial condition.*

**Proof** Thanks to Lemma 4 and 5, the matrix product  $\tilde{P}_{k_i, k_{i+1}}$  for each time interval  $[k_i, k_{i+1})$  satisfies (R1)–(R4). Therefore, one can prove this theorem by similarly following the proof of Theorem 1. □

## 6 Numerical experiments

### 6.1 Effects of primary local-feedback gain $\tilde{\xi}_k[j]$

This section first investigates the sensitivity of the primary feedback gain  $\tilde{\xi}_k[j]$  using Algorithm 3 [with Eq. (20)]. We show that, depending on the shape of the gain, the performance of the proposed framework changes with respect to convergence rate, fraction of transitioning agents, and residual convergence error.

We consider the scenario having a set of 2000 agents and an arena consisting of  $10 \times 10$  bins, as depicted in Fig. 2. There are vertical and horizontal paths between adjacent bins. The agents are allowed to move at most 3 paths away within a unit time instant. All the agents start from a bin, which reflects the fact that they are generally deployed from a base station at the beginning of a mission. The desired swarm distribution  $\Theta$  is uniform-randomly generated at each scenario. The agents are assumed to estimate necessary local information correctly.

The performance of the proposed algorithm will be compared with that of the GICA-based algorithm (Bandyopadhyay et al. 2017). To this end,  $f_E(E_k[j, l])$  is set to be the same as the corresponding coefficient in the existing work:

$$f_E(E_k[j, l]) := 1 - \frac{E_k[j, l]}{E_{k, \max} + \epsilon_E}, \tag{30}$$

where  $E_{k, \max}$  is the maximum element of the travelling expense matrix  $E_k$ , and  $\epsilon_E$  is a design parameter.  $E_k[j, l]$  is defined as a linear function based on the distance between bin  $\mathcal{B}_j$  and  $\mathcal{B}_l$ :

$$E_k[j, l] := \epsilon_{E_1} \cdot \Delta_{S(j,l)} + \epsilon_{E_0}, \tag{31}$$

where  $\Delta_{S(j,l)}$  is the minimum required number of paths from  $\mathcal{B}_j$  to  $\mathcal{B}_l$ ;  $\epsilon_{E_1}$  and  $\epsilon_{E_0}$  are design parameters. The agents are assumed to follow any shortest route when they transition between two bins. The design parameters are set as follows:  $\epsilon_{E_1} = 1$  and  $\epsilon_{E_0} = 0.5$  in (31);  $\epsilon_E = 0.1$  in (30);  $\epsilon_\xi = 10^{-9}$  in (6);  $\beta = 1.8 \times 10^5$  in (18); and  $\lambda = 10^{-6}$  in (9).

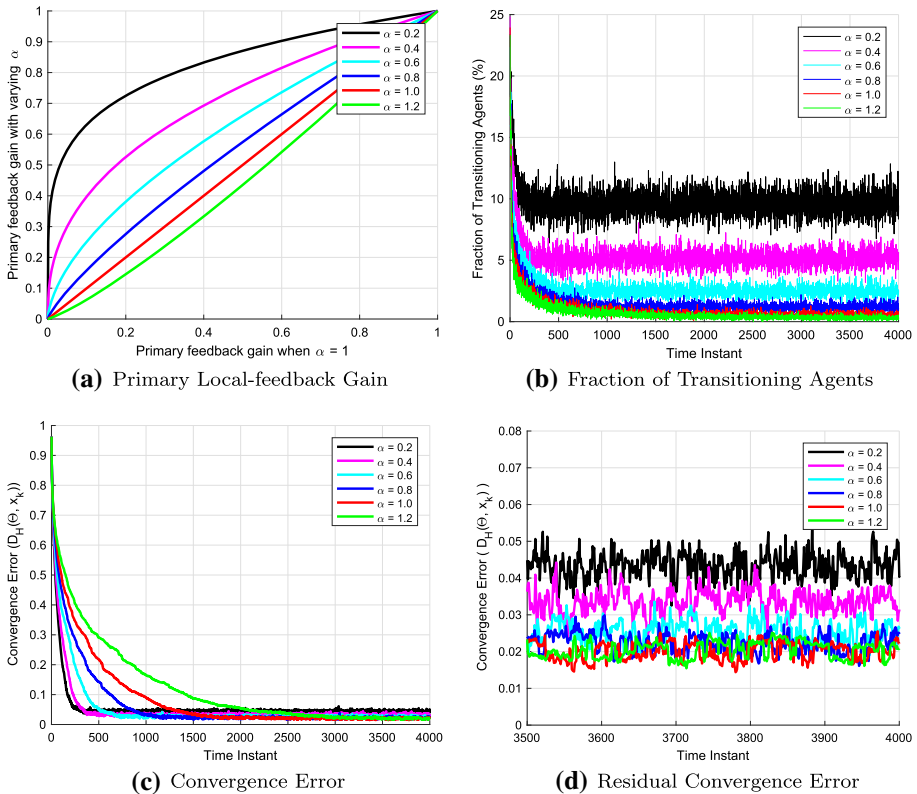
As a performance index for the closeness between  $\mathbf{x}_k$  and  $\Theta$ , we use *Hellinger Distance*, i.e.

$$D_H(\Theta, \mathbf{x}_k) := \frac{1}{\sqrt{2}} \sqrt{\sum_{j=1}^{n_b} (\sqrt{\Theta[j]} - \sqrt{\mathbf{x}_k[j]})^2},$$

which is known as a “concept of measuring similarity between two distributions” (Chung et al. 1989) and is utilised as a feedback gain in the existing work.

More importantly, to examine the effects of  $\tilde{\xi}_k[j]$ , we set  $\alpha$  in (6) as 0.2, 0.4, 0.6, 0.8, 1 and 1.2.





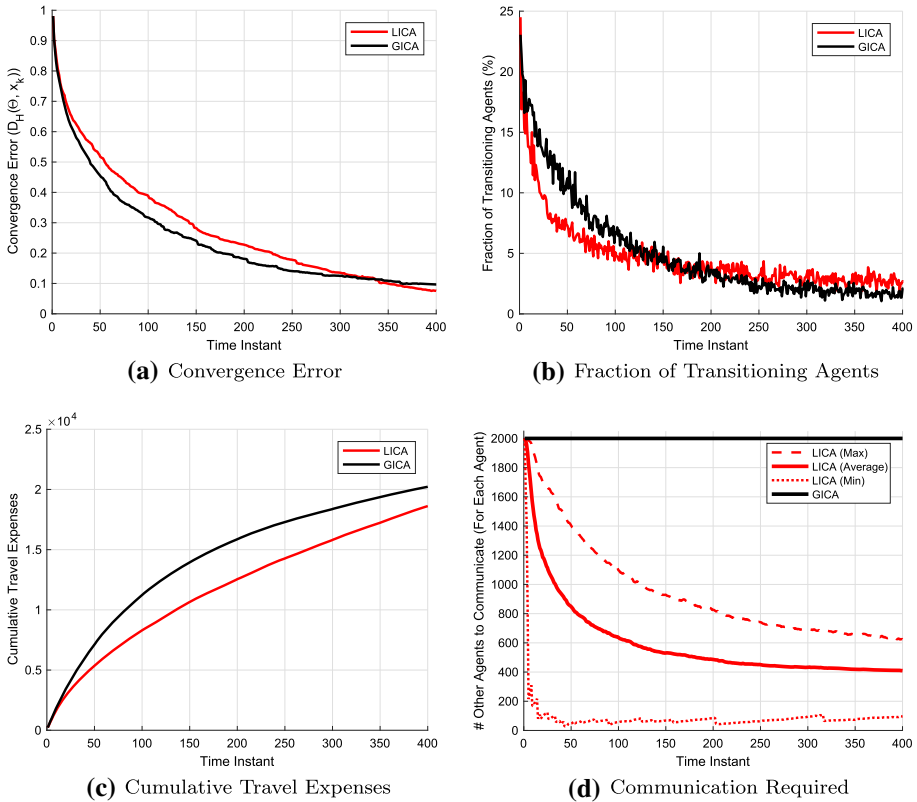
**Fig. 5** Sensitivity analysis depending on the primary local-feedback gain  $\bar{\xi}_k[j]$  in Eq. (6) with different setting of  $\alpha$ : **a** the value of  $\bar{\xi}_k[j]$ ; **b** the fraction of transitioning agents; **c** the convergence performance; **d** the convergence performance (zoomed-in for time instant between 3500 and 4000)

Figure 5 reveals that the convergence rate can be traded off against the fraction of transitioning agents and the residual convergence error. As  $\bar{\xi}_k[j]$  becomes more concave (i.e. the value of  $\alpha$  decreases), the summation of off-diagonal entries of  $P_k$  becomes higher, leading to more transitioning agents but faster convergence rate. At the same time, such unnecessarily higher off-diagonal entries of  $P_k$  even at a low value of  $|\bar{\theta}[j] - \bar{x}_k[j]|$  prevent the swarm from properly converging to the desired swarm distribution, resulting in higher residual convergence error.

### 6.2 Comparison with the GICA-based method

Let us now compare the LICA-based method for (P1) with the GICA-based method. The scenario considered is the same as the one in the previous subsection except for  $\alpha = 0.6$ . Note that  $\epsilon_\theta$  in Remark 5 improves convergence rate, but is not discussed in the existing work. For the fair comparison,  $\epsilon_\theta$  is applied to both methods. We conduct 100 runs of Monte Carlo experiments. Figure 6 presents the results of one representative scenario, and Fig. 7 shows the statistical results of the Monte Carlo experiments.

According to Fig. 6a, the convergence rate of the proposed method is comparable to that of the GICA-based method. Specifically, the former is slower at the initial phase, but becomes

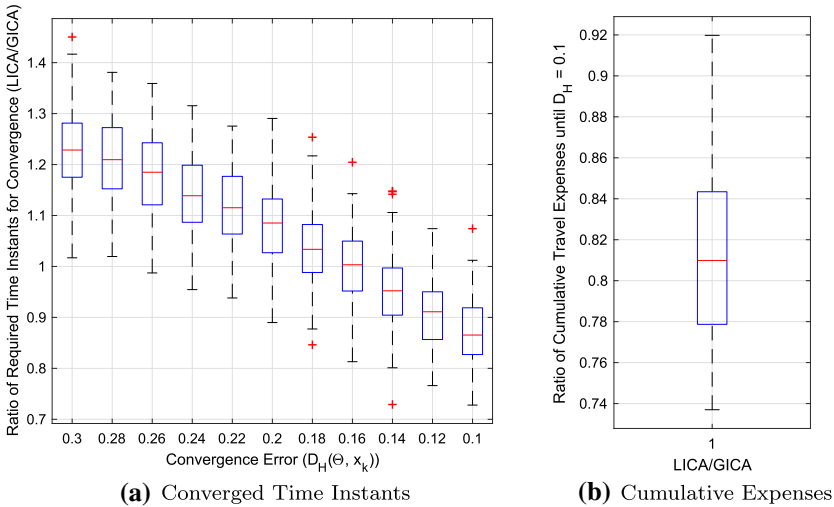


**Fig. 6** Performance comparison between the proposed method (LICA) with the existing method (GICA) (Bandyopadhyay et al. 2017): **a** the convergence error between the current swarm status and the desired status; **b** the fraction of agents transitioning between any two bins; **c** the cumulative travel expenses of all the agents from the beginning; **d** the number of other agents whose information is necessary for each agent

similar to that of the GICA-based method as reaching  $D_H(\theta, \mathbf{x}_k) = 0.10$ . This is confirmed by the statistical results in Fig. 7a, which presents the ratio of the required time instants for converging to  $D_H(\theta, \mathbf{x}_k) \in \{0.30, 0.28, \dots, 0.12, 0.10\}$  in the LICA-based method to those of the GICA-based method.

Figure 6c shows that the cumulative travel expenses are smaller in the proposed method than in the existing method. Until achieving  $D_H(\theta, \mathbf{x}_k) = 0.1$ , the expenses by the proposed method and those by the compared method are  $1.72 \times 10^4$  and  $1.96 \times 10^4$ , respectively, and their ratio is 0.878. This is also confirmed by the statistical result in Fig. 7b. A possible explanation is that when some of the bins do not meet their desired swarm densities, the entire swarm in the GICA-based method would obtain higher feedback gains, leading to unnecessary transitions. On the contrary, this is not the case in the LICA-based method since agents are only affected by their neighbour bins.

More importantly, Fig. 6d indicates that agents in the proposed framework require much less information from other agents. This figure shows the number of other agents whose information is necessary for each agent to generate its stochastic policies. For the LICA-based framework, the red dashed line and the red dotted line represent the maximum case (i.e. the agent who needs the largest amount of information) and the minimum case (i.e. the agent



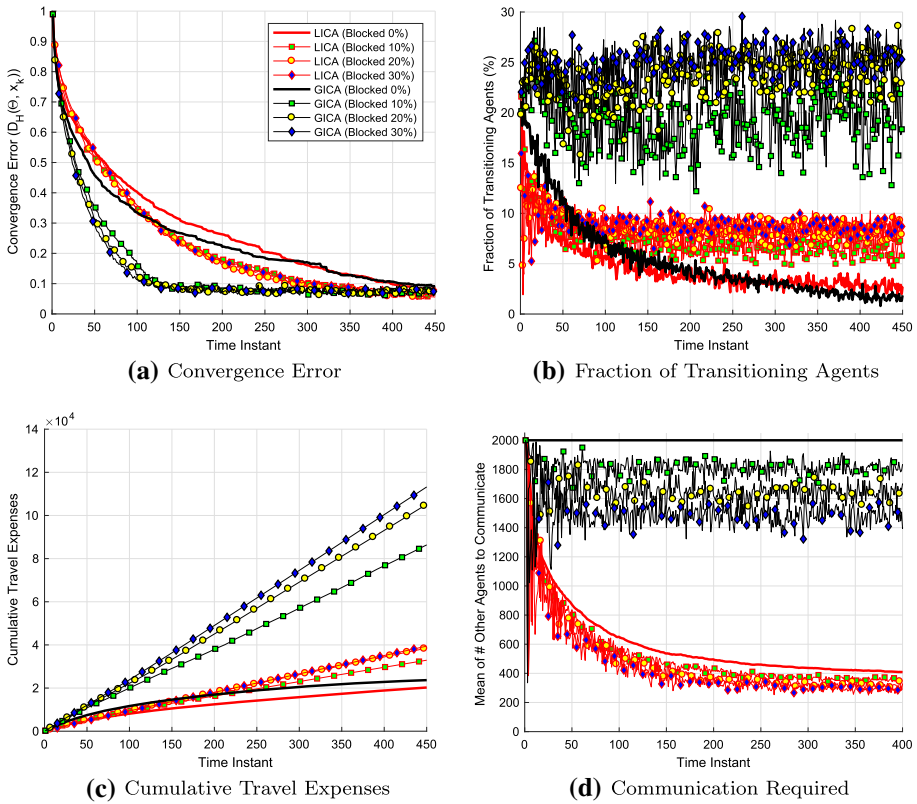
**Fig. 7** Performance comparison (Monte Carlo experiments) between the proposed method (LICA) and the existing method (GICA) (Bandyopadhyay et al. 2017): **a** the required time instants to converge to  $D_H(\theta, \mathbf{x}_k) \in \{0.30, 0.28, \dots, 0.12, 0.10\}$  (i.e. convergence rate); **b** the ratio of the cumulative travel expenses by LICA to those by GICA until converging to  $D_H(\theta, \mathbf{x}_k) = 0.1$

who needs the smallest amount of information) amongst all the agents, respectively. This results show that just 20 % of information are averagely required in the proposed method after the system converges such that  $D_H(\theta, \mathbf{x}_k) < 0.1$ , compared with the GICA-based method. This also implies that the LICA-based framework has a shorter timescale for each time instant interval and that its convergence performance in practice would be better. Note that the convergence comparison result in Fig. 6a is presented in respect to time instants of each Markov process.

### 6.3 Robustness in asynchronous environments

This subsection investigates the effects of asynchronous environments in the proposed LICA-based method for (P1) and compares them with those in the GICA-based method. We consider a realistic scenario where an asynchronous process is required: agents in some bins cannot communicate or cannot operate temporarily for some reasons (such bins are called *blocked*), and thus other agents in normal bins have to perform their own process without waiting for them. The proportion of blocked bins to the entire bins is set to be 0%, 10%, 20% and 30%. At each time instant, the corresponding proportion of bins become randomly blocked. Despite the absence of information from the blocked bins, we set that agents in normal bins anyway compute  $\bar{\mathbf{x}}_k[j]$  in the proposed method (or  $\mathbf{x}_k$  in the GICA-based method). For the proposed method, the asynchronous implementation in Sect. 5 is built with Algorithm 3. The rest of scenario setting are the same as those in Sect. 6.2.

Figure 8 illustrates the performance of each method: convergence rate, fraction of transitioning agents, cumulative travel expenses, and the amount of information to communicate. As the proportion of the blocked bins increases, the GICA-based method tends to have faster convergence speed, whereas it loses Desired Feature 2 and thus increases cumulative traveling expenses (as shown in Fig. 8a–c, respectively). On the contrary, the LICA-based method shows a graceful degradation in terms of Desired Feature 2 (as shown in Fig. 8b). A possible

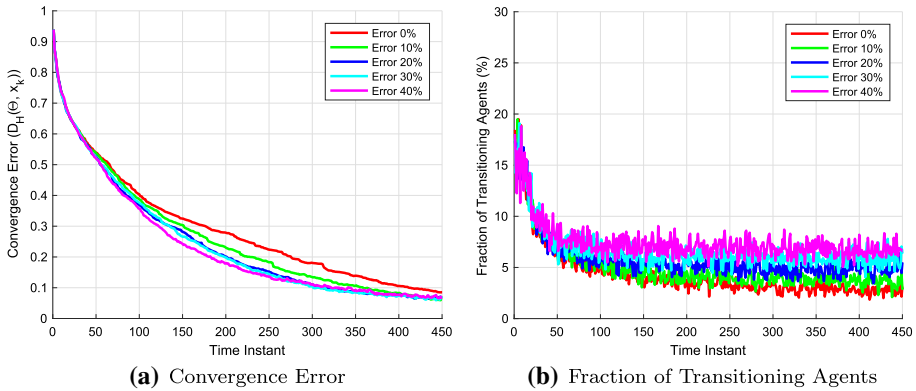


**Fig. 8** Performance comparison in communication-disconnected situations: **a** the convergence error between the current swarm status and the desired status (the legend applies to all the other subfigures); **b** the fraction of agents transitioning between any two bins; **c** the cumulative travel expenses of all the agents from the beginning; **d** the average number of other agents whose information is necessary for each agent

explanation for these results could be that higher feedback gains due to the communication disconnection induce faster convergence performance in each method than the normal situation. This effect is dominant for the GICA-based method because it affects the entire agents, who use global information. However, in the LICA-based framework, the communication disconnection only locally influences so that its effectiveness is relatively modest. Figure 8d shows that the proposed framework still only relies on much less information (e.g. averagely 20% after the system reasonably converges), compared with the existing method.

### 6.4 Effect of local information estimation error

Let us now examine the performance degradation of the proposed method when there exist estimation errors on local information. In this experiment, agent  $a_i$  in bin  $\mathcal{B}_j$  is set to locally perceive  $\mathbf{n}_k[l]$  as  $\mathbf{n}_k^i[l]$ ,  $\forall \mathcal{B}_l \in \mathcal{N}_k(j)$ , which is generated from a uniform random distribution  $[(1 - \eta) \cdot \mathbf{n}_k[l], (1 + \eta) \cdot \mathbf{n}_k[l]]$ , where  $\eta$  is the predefined error level. Then,  $\mathbf{n}_k^i[l]$  is used to compute  $\bar{\mathbf{x}}_k[j]$  as in Eq. (4). Apart from that, we use the same scenario setting in Sect. 6.2, while varying  $\eta \in \{0\%, 10\%, 20\%, 30\%, 40\%\}$ .



**Fig. 9** Performance degradation of the proposed framework in the existence of estimation error (from 10 to 40%) on local information  $\mathbf{n}_k[l]$  about neighbour bins: **a** the convergence behaviour; **b** the fraction of transitioning agents

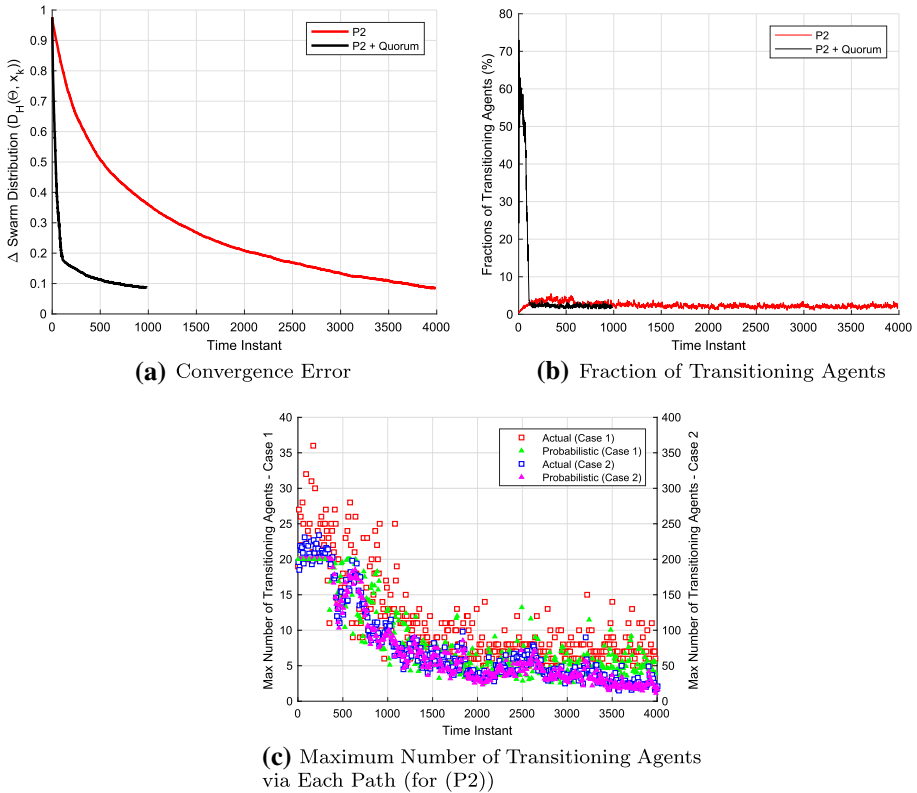
Figure 9 shows that despite the estimation errors, the proposed LICA-based method still achieves Desired Features 1 and 2 with graceful performance degradation. The uncertainty induces faster convergence but keeps a higher number of transitioning agents even at the last phase. A possible reason behind this would be similar to that for the results in Sect. 6.3. In practice, such faster convergence behaviour caused by uncertainties provide obvious benefits. The increased transitioning agents at the last phase could be addressed by allowing them to utilise more time to estimate the local information more accurately as the system converges (e.g. by setting variable time instants), considering the fact that the costs of physical transition between bins are in general much more expensive than those for communication. Alternatively, we could also utilise global information once in a while and make the agents forcefully settle down when a certain level of the desired global status is achieved, as similar to Bandyopadhyay et al. (2017).

### 6.5 Demonstration of examples II and III

This subsection demonstrates the LICA-based method for (P2) (i.e. Algorithm 4) and the quorum model (i.e. Algorithm 5). For the former, we consider a scenario where 10,000 agents and an arena consisting of  $10 \times 10$  bins are given. The arena is as depicted in Fig. 2, where the agents are allowed to move only one path away within a unit time instant. For each one-way path, the flux upper limit per time instant is set as 20 agents (i.e.  $c_{(j,l)} = 20, \forall j, \forall l \neq j$ ). All the agents start from a bin, and the desired swarm distribution is uniformly-randomly generated.

For the latter, we build the quorum model upon the LICA-based method for (P2). Combining the two may be a good strategy for a user who wants to achieve not only faster convergence rate but also lower unnecessary transitions after equilibrium, which are regulated by the flux upper limits. In the same scenario described above, we will demonstrate the combined algorithm that computes  $S_k$  and  $G_k$  by Algorithm 5 and  $P_k$  by Algorithm 4. We set  $q_j = 1.3$  and  $\gamma = 30$  for (28);  $\alpha = 1$  and  $\epsilon_\xi = 10^{-9}$  for (6); and  $\lambda = 10^{-6}$  for (9).

Figure 10a shows that both approaches make the swarm converge to the desired swarm distribution. It is observed from Fig. 10b that the number of transitioning agents in the method for (P2) is restricted because of the upper flux bound during the entire process. Meanwhile, the quorum-based method very quickly disseminates the agents, who are initially at one bin, over



**Fig. 10** Comparison results between the method for (P2) and the quorum-based model: **a** the convergence error between the current swarm status and the desired status; **b** the fraction of agents transitioning between any two bins; **c** the maximum number of transitioning agents via each path in the method for (P2) (Case 1:  $|\mathcal{A}| = 10,000$  and  $c_{(j,l)} = 20, \forall j, \forall l \neq j$ ; Case 2:  $|\mathcal{A}| = 100,000$  and  $c_{(j,l)} = 200, \forall j, \forall l \neq j$ )

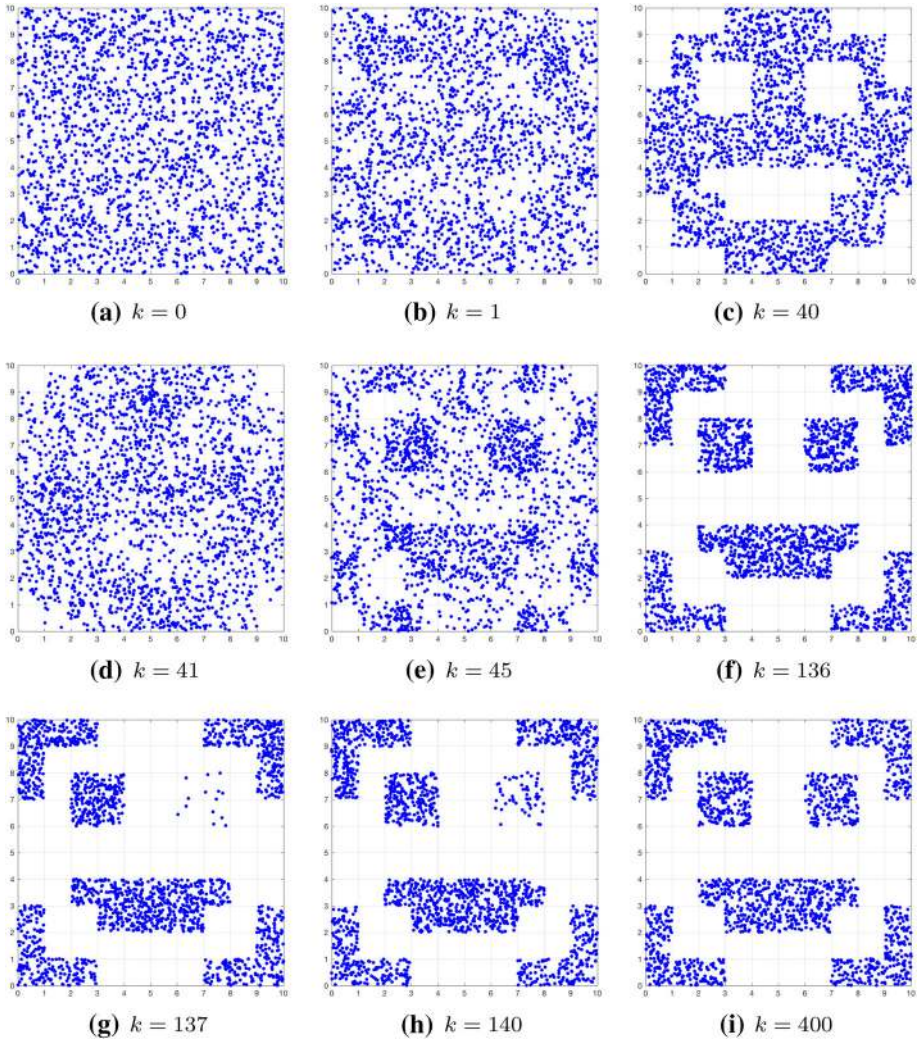
the other bins, and thus the fraction of transitioning agents is very high in the initial phase. After that, the population fraction drops and remains as low as the resultant transitioning for the method for (P2).

Figure 10c reports the maximum value amongst the number of transitioning agents via each (one-way) path. Note that the results are plotted every 10 time steps for clarity. The red squares indicate the actual result by the method for (P2), while the green triangles indicate the corresponding probabilistic values (i.e.  $\max_{\forall j} \max_{\forall l \neq j} \mathbf{n}_k[j] P_k[j, l]$ ). It is shown that the stochastic policies reflect the given upper bound, even though this bound is often violated in practice due to the finite-number agents’ randomness. However, the result in the same scenario with setting  $|\mathcal{A}| = 100,000$  and  $c_{(j,l)} = 200, \forall j, l \neq j$  (denoted by Case 2), depicted by the blue squares and the magenta triangles in Fig. 10c, suggests that such violation can be mitigated as the number of given agents increases.

### 6.6 Visualised adaptiveness test

We now consider a scenario in which the robots must be distributed spatially to resemble a reference image. Every pixel of an area is regarded as a bin. We have  $n_a = 2000$  agents





**Fig. 11** Visualisation results: 2000 agents are deployed into 100 pixels (bins) to configure images

and  $n_b = 10 \times 10$  bins, and all the bins are connected as shown in Fig. 2. The scenario considered is as follows. Initially (i.e. at  $k = 0$ ), all the agents are randomly distributed over the area, and they know the desired distribution vector, which is a smile icon. At  $k = 41$ , the inverted-colour icon is given to them as a new desired distribution vector. Then (i.e. at  $k = 137$ ), it happens that some agents for the right eye of the smiling face are somehow unexpectedly eliminated. The disappearance of agents is apparent only to the remaining neighbouring agents, but not to farther agents. We use the proposed algorithm for (P1) (i.e. Algorithm 3) and additionally adopt a subroutine whereby agents in zero-desired-density bins randomly move to one of its closest neighbour bins. Due to this subroutine and the constraint  $\Theta[j]P_k[j, l] = \Theta[l]P_k[l, j]$  in Requirement 3 (i.e. not allowed to move to a zero-desired-density bin from a positive-desired-density one), all the agents eventually remain in



the positive-desired-density bins. The rest of the dynamics of system are the same as those in Sect. 6.2. The visualised results are shown in Fig. 11.

## 7 Conclusion

In this paper, we propose a LICA(Local Information Consistency Assumption)-based closed-loop Markov chain framework for probabilistic swarm distribution guidance. To generate feedback, our framework only requires the agents to employ local information. Consequently, the agents exhibit reduction in communication, shorter timescales for obtaining new information, asynchronous decision-making, and deployability without a priori mission knowledge. One of the major features of our approach is that, even if the information used is limited, the agents can converge to a desired density distribution, while ensuring scalability, robustness, and long-term system efficiency. The numerical experiments reveal that the proposed framework is more robust when arbitrary parts of the inter-agent communication is sporadically disconnected. We discuss the design requirements for the Markov matrix to hold all these advantages, and we provide specific implementation examples.

Future work will investigate optimisation of  $\bar{\xi}_k[j]$ , which can mitigate the trade-off between convergence rate and residual error. In addition, it is expected that the communication cost can be further reduced by incorporating a vision-based local density estimation (Saleh et al. 2015). Another natural progression would be to extend this study to accommodate heterogeneous agents as shown in Prorok et al. (2017), which addressed how to distribute different agents having various multiple traits into a desired trait distribution rather than a desired population density. It would be also interesting to compare this Markov chain framework with others (e.g. a game-theoretical approach in Jang et al. (2018b)). Some results were preliminarily shown in Jang et al. (2018), but will include real-robot experiments using MONA robots (Arvin et al. 2018) or Kilobots (Rubenstein et al. 2014).

**Acknowledgements** The authors gratefully acknowledge that this research was supported by International Joint Research Programme with Chungnam National University (No. EFA3004Z). Thanks to Matthew Nancekievill, Saptarshi Bandyopadhyay, Namhoon Cho, and Sangjun Bae for their valuable and constructive discussions.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## Appendix

### A. Proof for Theorem 1

**Definition 10** (*Irreducible*) A matrix is *reducible* if and only if its associated digraph is not strongly connected. A matrix that is not reducible is *irreducible*.

**Definition 11** (*Primitive*) A *primitive* matrix is a square non-negative matrix  $\mathcal{M}$  such that for every  $i, j$  there exists  $k > 0$  such that  $\mathcal{M}^k[i, j] > 0$ .

**Definition 12** (*Regular*) A *regular* matrix is a stochastic matrix such that all the entries of some power of the matrix are positive.

**Definition 13** (Seneta 1981, pp. 92, 149) (*Asymptotic Homogeneity*) “A sequence of stochastic matrices  $\mathcal{M}_k \in \mathbb{M}^{n \times n}$ ,  $k \geq k_0$ , is said to be *asymptotically homogeneous* (with respect to  $d$ ) if there exists a row-stochastic vector  $d \in \mathbb{P}^n$  such that  $\lim_{k \rightarrow \infty} d\mathcal{M}_k = d$ .” (Bandyopadhyay et al. 2017)

**Definition 14** (Seneta 1981, pp. 92, 149) (*Strong Ergodicity*) “The forward matrix product  $\mathcal{U}_{k_0,k} := \mathcal{M}_{k_0}\mathcal{M}_{k_0+1} \dots \mathcal{M}_{k-1}$ , formed from a sequence of stochastic matrices  $\mathcal{M}_r \in \mathbb{P}^{n \times n}$ ,  $r \geq k_0$ , is said to be *strongly ergodic* if for each  $j, l, r$  we get  $\lim_{r \rightarrow \infty} \mathcal{U}_{k_0,r}[j, l] = \mathbf{v}[l]$ ”, where  $\mathbf{v} \in \mathbb{P}^n$  is a row-stochastic vector. Here,  $\mathbf{v}$  is called its unique limit vector (i.e.  $\lim_{r \rightarrow \infty} \mathcal{U}_{k_0,r} = \mathbf{1}^\top \mathbf{v}$ ). (Bandyopadhyay et al. 2017)

**Lemma 6** *Given the requirements (R1)–(R4) are satisfied,  $M_k$  in Eq. (10) has the following properties:*

1. row stochastic;
2. irreducible;
3. all diagonal elements are positive, and all other elements are non-negative;
4. there is a positive lower bound  $\kappa$  such that  $0 < \kappa \leq \min_{j,l}^+ M_k[j, l]$  (note that  $\min^+$  refers to the minimum of the positive elements);
5. asymptotically homogeneous with respect to  $\Theta$ .  
 In addition,  $U_{k_0,k}$  in Eq. (11) has the following properties:
6. irreducible;
7. primitive.

**Proof** This lemma can be proved by similarly following the mathematical development for Bandyopadhyay (2017, Theorem 4).  $M_k$  is row stochastic because  $M_k \cdot \mathbf{1}^\top = (I - W_k)P_k \cdot \mathbf{1}^\top + W_k S_k \cdot \mathbf{1}^\top = (I - W_k) \cdot \mathbf{1}^\top + W_k \cdot \mathbf{1}^\top = \mathbf{1}^\top$ .  $P_k$  is irreducible and  $\omega_k[j]$  is always less than 1; thus,  $M_k$  is also irreducible (i.e.  $M_k[j, l] > 0$  if  $P_k[j, l] > 0$ ). The property 3 is true because  $\text{diag}(I - W_k) > \mathbf{0}$ ,  $W_k \geq \mathbf{0}$ , and  $P_k$  is also a non-negative matrix such that its diagonal elements are positive. The property 4 is implied by either the property 2 or 3. From the definition of  $W_k$  [i.e. Eq. (9)], it follows that  $\lim_{k \rightarrow \infty} W_k = \mathbf{0}$  because of  $\exp(-\lambda k)$ , and thereby  $\lim_{k \rightarrow \infty} M_k = \lim_{k \rightarrow \infty} P_k$ . Hence,  $\lim_{k \rightarrow \infty} \Theta M_k = \lim_{k \rightarrow \infty} \Theta P_k = \Theta$ , and the property 5 is valid.

Let us now turn to  $U_{k_0,k}$ . If  $M_r[j, l] > 0$  for some  $r \in \{k_0, \dots, k - 1\}$  and  $j, l \in \{1, \dots, n_b\}$ , then the corresponding element  $U_{k_0,k}[j, l] > 0$  (Bandyopadhyay et al. 2017, Theorem 4). Thus, due to the property 2,  $U_{k_0,k}$  is irreducible as well. Besides, it follows from (Horn and Johnson 2012, Lemma 8.5.4, p. 541) that  $U_{k_0,k}$  is primitive: “if a square matrix is irreducible, non-negative and all its main diagonal entries are positive, then the matrix is primitive”. □

**Proof for Theorem 1** Theorem 1 can be proved by following similar steps in proving (Bandyopadhyay et al. 2017, Theorem 4). The claim in the thorem is true if  $\lim_{k \rightarrow \infty} \mathbf{x}_k = \mathbf{x}_{k_0} \cdot \lim_{k \rightarrow \infty} U_{k_0,k} = \mathbf{x}_{k_0} \cdot \mathbf{1}^\top \Theta = \Theta$ . In order for that, the matrix product  $U_{k_0,k}$  should (i) be *strongly ergodic* and (ii) have  $\Theta$  as its unique limit vector, i.e.  $\lim_{k \rightarrow \infty} U_{k_0,k} = \mathbf{1}^\top \Theta$ . We will show that the two conditions are valid under the assumption that (R1)–(R4) are satisfied.

From Lemma 6, we found that (a)  $U_{k_0,k}$  is primitive (thus, regular); (b) there is a positive lower bound  $\kappa$  such that  $0 < \kappa \leq \min_{j,l}^+ M_k[j, l]$ ,  $\forall k$ ; and (c)  $M_k$  is asymptotically homogeneous. Then, it follows from (Seneta 1981, Theorem 4.15, p. 150) that  $U_{k_0,k}$  is strongly ergodic with respect to a certain vector  $\mathbf{v} \in \mathbb{P}^{n_b}$ , which fulfils the condition (i).

Let  $\mathbf{e}_k \in \mathbb{P}^{n_b}$  be the unique stationary distribution vector corresponding to  $M_k$  (i.e.  $\mathbf{e}_k M_k = \mathbf{e}_k$ ). Due to the prior condition (b) and the fact that (d)  $M_k$  is irreducible for  $\forall k \geq k_0$ , it

follows from (Seneta 1981, Theorem 4.12, p. 149) that the asymptotical homogeneity of  $M_k$  with respect to  $\Theta$  (i.e.  $\lim_{k \rightarrow \infty} \Theta M_k = \Theta$ ), given by Lemma 6, is equivalent to both  $\lim_{k \rightarrow \infty} \mathbf{e}_k = \mathbf{e}$ , where  $\mathbf{e}$  is a limit vector, and  $\Theta = \mathbf{e}$ . According to Seneta (1981, Corollary, p. 150), under the prior conditions (b) and (d) and if  $U_{k_0, k}$  is strongly ergodic with its unique limit vector  $\mathbf{v}$ , then  $\mathbf{v} = \mathbf{e}$ . Hence, it turns out that the unique limit vector of  $U_{k_0, k}$  is  $\mathbf{v} = \mathbf{e} = \Theta$  (i.e.,  $\lim_{k \rightarrow \infty} U_{k_0, k} = \mathbf{1}^T \Theta$ ). Thereby, the condition (ii) is also fulfilled.  $\square$

## B. Proof for Corollary 3

**Proof for Corollary 3** We can prove this by following the proof of Bandyopadhyay (2017, Corollary 1). Suppose that the problem is only subject to (R4) and (14), without (R1)–(R3) and (R5). Then, the off-diagonal elements of an optimal matrix should be their corresponding lower bounds in (14) if  $A_k[j, l] = 1$ . The diagonal elements of the matrix do not affect the objective function due to the fact that  $E_k[j, j] = 0, \forall j$ . Accordingly, the matrix  $P_k$  that holds (16) and (17) is also an optimal matrix for the simplified problem.

Let us now consider (R1)–(R3) and (R5). Since  $f_{\xi}(\bar{\xi}_k[j], \bar{\xi}_k[l])$ , and  $f_E(E_k[j, l])$  are upper-bounded by 1 and  $\sum_{\forall l \neq j} \Theta[l] < 1$ ,  $P_k[j, j]$  in (17) is always positive for all  $j$ , which fulfils (R2). It also follows that (R1) is satisfied by Eq. (17). From Eq. (16), it holds that  $\Theta[j]P_k[j, l] = \Theta[l]P_k[l, j]$ , complying with (R3). Since (R1)–(R4) are satisfied, the Markov process is converging to a desired distribution due to Theorem 1. Noting that  $f_{\xi}(\bar{\xi}_k[j], \bar{\xi}_k[l])$  diminishes as  $\bar{\xi}_k$  gets close to  $\mathbf{0}$  (i.e.  $\bar{\mathbf{x}}_k \rightarrow \bar{\Theta}$ ), (R5) is also fulfilled by Eqs. (16) and (17). Hence,  $P_k$  is the optimal solution for problem (P1).  $\square$

## References

- Acikmese, B., & Bayard, D. S. (2012). A Markov chain approach to probabilistic swarm guidance. In *2012 American control conference* (pp. 6300–6307). IEEE.
- Acikmese, B., & Bayard, D. S. (2014). Probabilistic swarm guidance for collaborative autonomous agents. In *2014 American control conference* (pp. 477–482). IEEE.
- Acikmese, B., & Bayard, D. S. (2015). Markov chain approach to probabilistic guidance for swarms of autonomous agents. *Asian Journal of Control*, *17*(4), 1105–1124.
- Arvin, F., Espinosa, J., Bird, B., West, A., Watson, S., & Lennox, B. (2018). Mona: An affordable open-source mobile robot for education and research. *Journal of Intelligent and Robotic Systems*. <https://doi.org/10.1007/s10846-018-0866-9>.
- Bandyopadhyay, S., & Chung, S. J. (2014). Distributed estimation using Bayesian consensus filtering. In *2014 American control conference* (pp. 634–641). IEEE.
- Bandyopadhyay, S., Chung, S. J., & Hadaegh, F. Y. (2017). Probabilistic and distributed control of a large-scale swarm of autonomous agents. *IEEE Transactions on Robotics*, *33*(5), 1103–1123.
- Becco, C., Vandewalle, N., Delcourt, J., & Poncin, P. (2006). Experimental evidences of a structural and dynamical transition in fish school. *Physica A: Statistical Mechanics and its Applications*, *367*, 487–493.
- Berman, S., Halasz, A., Hsieh, M. A., & Kumar, V. (2008). Navigation-based optimization of stochastic strategies for allocating a robot swarm among multiple sites. In *2008 IEEE conference on decision and control* (pp. 4376–4381). IEEE.
- Berman, S., Halasz, A., Hsieh, M. A., & Kumar, V. (2009). Optimized stochastic policies for task allocation in swarms of robots. *IEEE Transactions on Robotics*, *25*(4), 927–937.
- Bestaoui Sebbane, Y. (2014). *Planning and decision making for aerial robots, intelligent systems, control and automation: Science and engineering* (Vol. 71). Cham: Springer.
- Chattopadhyay, I., & Ray, A. (2009). Supervised self-organization of homogeneous swarms using ergodic projections of Markov chains. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *39*(6), 1505–1515.
- Chung, J., Kannappan, P., Ng, C., & Sahoo, P. (1989). Measures of distance between probability distributions. *Journal of Mathematical Analysis and Applications*, *138*(1), 280–292.

- Couzin, I. D., Krause, J., James, R., Ruxton, G. D., & Franks, N. R. (2002). Collective memory and spatial sorting in animal groups. *Journal of Theoretical Biology*, 218(1), 1–11.
- Couzin, I. D., Krause, J., Franks, N. R., & Levin, S. A. (2005). Effective leadership and decision-making in animal groups on the move. *Nature*, 433(7025), 513–516.
- Demir, N., & Acikmese, B. (2015). Probabilistic density control for swarm of decentralized ON–OFF agents with safety constraints. In *2015 American control conference* (pp. 5238–5244). IEEE.
- Demir, N., Eren, U., & Açkmece, B. (2015). Decentralized probabilistic density control of autonomous swarms with safety constraints. *Autonomous Robots*, 39(4), 537–554.
- Gautrais, J., Jost, C., & Theraulaz, G. (2008). Key behavioural factors in a self-organised fish school model. *Annales Zoologici Fennici*, 45(5), 415–428.
- Halasz, A., Hsieh, M. A., Berman, S., & Kumar, V. (2007). Dynamic redistribution of a swarm of robots among multiple sites. In *2007 IEEE/RSJ international conference on intelligent robots and systems* (pp. 2320–2325). IEEE.
- Hoare, D. J., Couzin, I. D., Godin, J. G. J., & Krause, J. (2004). Context-dependent group size choice in fish. *Animal Behaviour*, 67(1), 155–164.
- Horn, R. A., & Johnson, C. R. (2012). *Matrix analysis* (2nd ed.). Cambridge: Cambridge University Press.
- Hsieh, M. A., Halasz, A., Berman, S., & Kumar, V. (2008). Biologically inspired redistribution of a swarm of robots among multiple sites. *Swarm Intelligence*, 2(2–4), 121–141.
- Ipsen, I. C. F., & Selee, T. M. (2011). Ergodicity coefficients defined by vector norms. *SIAM Journal on Matrix Analysis and Applications*, 32(1), 153–200.
- Jadbabaie, A., Lin, J., & Morse, A. (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6), 988–1001.
- Jang, I., Shin, H. S., & Tsourdos, A. (2018b). Anonymous hedonic game for task allocation in a large-scale multiple agent system. *IEEE Transactions on Robotics*. <https://doi.org/10.1109/TRO.2018.2858292>.
- Jang, I., Shin, H. S., & Tsourdos, A. (2018a). A comparative study of game-theoretical and Markov-chain-based approaches to division of labour in a robotic swarm. *IFAC-PapersOnLine*, 51(12), 62–68.
- Johnson, L. B., Ponda, S., Choi, H. L., & How, J. P. (2011). Asynchronous decentralized task allocation for dynamic environments. In *Infotech@Aerospace 2011*. <https://doi.org/10.2514/6.2011-1441>.
- Johnson, L. B., Choi, H. L., & How, J. P. (2016). The role of information assumptions in decentralized task allocation: A tutorial. *IEEE Control Systems*, 36(4), 45–58.
- Keller, L., Krieger, M. J. B., & Billeter, J. B. (2000). Ant-like task allocation and recruitment in cooperative robots. *Nature*, 406(6799), 992–995.
- Lerman, K., Martinoli, A., & Galstyan, A. (2005). A review of probabilistic macroscopic models for swarm robotic systems. *Swarm robotics* (pp. 143–152). Berlin: Springer.
- Luo, R., Chakraborty, N., & Sycara, K. (2014). Supervisory control for cost-effective redistribution of robotic swarms. In *2014 IEEE international conference on systems, man, and cybernetics* (pp. 596–601). IEEE.
- Mather, T. W., & Hsieh, M. A. (2011). Macroscopic modeling of stochastic deployment policies with time delays for robot ensembles. *The International Journal of Robotics Research*, 30(5), 590–600.
- Morgan, D., Subramanian, G. P., Bandyopadhyay, S., Chung, S. J., & Hadaegh, F. Y. (2014). Probabilistic guidance of distributed systems using sequential convex programming. In *2014 IEEE/RSJ international conference on intelligent robots and systems* (pp. 3850–3857). IEEE.
- Partridge, B. L. (1982). The structure and function of fish schools. *Scientific American*, 246(6), 114–123.
- Prorok, A., Hsieh, M. A., & Kumar, V. (2017). The impact of diversity on optimal control policies for heterogeneous robot swarms. *IEEE Transactions on Robotics*, 33(2), 346–358.
- Rubenstein, M., Ahler, C., Hoff, N., Cabrera, A., & Nagpal, R. (2014). Kilobot: A low cost robot with scalable operations designed for collective behaviors. *Robotics and Autonomous Systems*, 62(7), 966–975.
- Saleh, S. A. M., Suandi, S. A., & Ibrahim, H. (2015). Recent survey on crowd density estimation and counting for visual surveillance. *Engineering Applications of Artificial Intelligence*, 41, 103–114.
- Seeley, T. D. (1995). *The wisdom of the hive*. Cambridge, MA: Harvard University Press.
- Seneta, E. (1981). *Non-negative matrices and Markov chains*. Springer series in statistics. New York: Springer.