An Adaptive Fusion Strategy for Distributed Information Estimation Over Cooperative Multi-Agent Networks

Daxin Tian, *Senior Member, IEEE,* Jianshan Zhou, and Zhengguo Sheng

Abstract—In this paper, we study the problem of distributed information estimation that is closely relevant to some network-based applications such as distributed surveillance, cooperative localization and optimization. We consider a problem where an application area containing multiple information sources of interest is divided into a series of subregions in which only one information source exists. The information is presented as a signal variable which has finite states associated with certain probabilities. The probability distribution of information states of all the subregions constitutes a global information picture for the whole area. Agents with limited measurement and communication ranges are assumed to monitor the area, and cooperatively create a local estimate of the global information. To efficiently approximate the actual global information using individual agents' own estimates, we propose an adaptive distributed information fusion strategy and use it to enhance the local Bayesian rule based updating procedure. Specifically, this adaptive fusion strategy is induced by iteratively minimizing a Jensen-Shannon divergence based objective function. A constrained optimization model is also presented to derive minimum Jensen-Shannon divergence weights at each agent for fusing local neighbors' individual estimates. Theoretical analysis and numerical results are supplemented to show the convergence performance and effectiveness of the proposed solution.

Index Terms—Cooperative information estimation, adaptive distributed fusion, nonlinear constrained optimization, multi-agent networks, Jensen-Shannon divergence

I. INTRODUCTION

NETWORK-TYPE systems are general in both nature (such as fish schools, ants and honeybee swarms) and engineering (such as unmanned aerial vehicles, mobile robots, and other wireless sensor networks). In these systems, information estimation and fusion over multi-agent networks is of great significance, which can support individual agents to achieve some common tasks in a distributed manner such as environmental monitoring, global localization, self-defending or attacking invaders, etc. However, some challenges exist to be dealt with for practically realizing highly-scalable information estimation and fusion paradigms such as limited individual detection and interaction, lack of centralized control, and dynamic and noisy nature of measurements obtained

by every agent. In some application scenarios relevant to object detection or target locating, the goal of an information estimation model is to estimate the actual probability that a certain target is present in a given closed region. The probability of a target existence within a given surveillance region is usually assumed to follow the Bernoulli distribution and all of the probabilities corresponding to different regions constitute a so-called probability map [1]–[3]. The estimation of the individual probability map can be iteratively updated by following Bayesian rule. For example, in [3], to realize a distributed strategy for probability map estimation, Bayesian updating is combined with the traditional consensus protocol, which is used for fusing different individual probability maps of the neighbors of an agent. Nevertheless, although the proposed estimation fusion strategy based on Bayesian updating is useful in the static object detection, it may fail to be applied in a more general scenario where the probability of a subregion state or an object state does not follow the Bernoulli distribution.

Additionally, many distributed solutions have been proposed in the context of adaptive distributed LMS (Least-Mean-Square) estimation, which include the incremental adaptive strategies [4]–[6], the consensus based strategies [7]–[9] and the adaptive diffusion strategies [10]–[14]. Specifically, two diffusion strategies ATC (Adapt-Then-Combine) and CTA (Combine-Then-Adapt) have been proven to be powerful to realize distributed optimization and cooperative learning over networks [12], [13], [15]–[19]. In most of these studies, the distributed optimization is always modeled as an unconstrained LMS estimation problem, in which the global objective function is formulated as a sum of all individual components. The global function has to be localized so that the distributed optimization procedure can be induced by adopting the steepest descent algorithm. However, the unconstrained LMS estimation solutions are not suitable in some specific application situations, where a global task should be formulated as a constrained distributed optimization problem. In this context, distributed solutions are required to satisfy some certain estimation constraints at each agent, which could be more challenging and beyond the conventional unconstrained LMS estimation schemes. Since the information estimation over a distributed system can be modeled as a distributed optimization problem, some distributed optimization algorithms with consideration of certain constraints have been proposed [20], [21]. In [21], two distributed optimization cases are considered: the first case does not take into account

the equality constraints and employs Lagrangian relaxation approach to devise a distributed Lagrangian primal-dual subgradient algorithm; the second case takes into account the equality constraints and adopts a penalty relaxation approach. Although both of equality and inequality constraints are taken into account in the second case, it require identical local constraint sets to guarantee the Slater's condition due to the nature of penalty relaxation approach.

In this paper, we present an ATC-type distributed framework, which includes a nonlinear constrained optimization model and a fusion-weight optimization model. The objective function of the optimization model is defined based on the Jensen-Shannon divergence [22], which allows the individual fusion to iteratively approach to the actual information in terms of minimizing the information-theoretic divergence measure based on the Shannon entropy. Besides, the fusion-weight optimization model is proposed to iteratively adapt the fusion weights of each agent according to estimation results collected from the other neighbors. Different from the existing work, the distributed optimization algorithm proposed in this paper does not need the assumption of identical constraint sets. Furthermore, another essential difference between our work and [21] is the distributed framework, where we adopt a projection gradient approach and the distributed processing is based on ATC. By resorting to the probabilistic theory, the distributed information estimation is generalized as the process of learning and approaching to probability distributions over the multi-agent network. We do not assume any specific given distribution (for instance, the Bernoulli distribution adopted in [1], [3], [23], [24]) for the probabilities of the information states. Hence, the proposed solution can be deployed for a wide range of distributed applications once the detection information of interest is appropriately represented as a certain discrete probability distribution.

The remainder of the paper is organized as follows. In Section II, relevant preliminaries are briefly outlined for the distributed information estimation problem, which include main mathematical notations and definitions. In Section III, we introduce the non-cooperative individual information estimation scheme, where the Bayesian rule is adopted to incorporate the individual measurements. In Section IV, the nonlinear constrained optimization model is proposed for improving individual estimation. Section V presents the experimental results of the adaptive fusion strategy. Finally, Section VI concludes this paper.

II. PRELIMINARIES

A. Notations and Definitions

Notations: Throughput this paper, we use $\text{col}\{x_1, \ldots, x_n\}$ to represent a column vector constructed by stacking entries $x_1, \ldots, x_n$ on top of each other, and $\text{diag}\{x_1, \ldots, x_n\}$ to represent a diagonal matrix with diagonal entries $x_1, \ldots, x_n$. Besides, let $\mathbf{1}_{n \times 1}$ be a column vector of n dimensions all of whose entries are equal to 1, and $\mathbf{0}_{n \times 1}$ be a full-zero column vector of n dimensions. The identity matrix of size n is denoted by $\boldsymbol{I}_n$. Unless otherwise specified, all vectors are column vectors and denoted by boldface lowercase letters, while matrices are denoted by boldface capital letters.

Definitions: Given a multi-agent network, we use a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ to represent its communication topology with $\mathcal{V} = \{1, 2, \ldots, n\}$ denoting a node set and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ denoting an edge set consisting of unordered pairs $\mathcal{E} = \{(i,j) | i, j \in \mathcal{V}\}$ excluding self-loop (i,i). $(i,j) \in \mathcal{E}$ represents a mutual communication between the agents i and j, and any agent $i \in \mathcal{V}$ is supposed to be periodically communicating with its immediate neighbors $\{j | j \in \mathcal{V}, (j,i) \in \mathcal{E}\}$ through one-hop broadcasting-based communication. We assume that the communication graph is a connected graph, i.e., the communication is bidirectional and there is always a path between any two agents in the network.

B. Problem Formulation

The multi-agent network is assumed to be deployed to detect a geographical region that contains multiple information sources. The set of the whole information sources is denoted by Π, whose cardinality is m, i.e., $m = |\Pi|$. Then, the entire surveillance region can be divided into a series of surveillance subregions, each of which corresponds to one information source. An agent i can only detect a part of subregions, i.e., a fraction of information sources. The set of partial information sources in i's detection range is defined as π_i, $\pi_i \subsetneq \Pi$ for all $i \in \mathcal{V}$ and $\bigcup_{i \in \mathcal{V}} \pi_i = \Pi$. Without loss of generality, we also assume that the detection regions of any two different agents $i_1 \neq i_2$ $(i_1, i_2 \in \mathcal{V})$ are not identical, i.e., $\pi_{i_1} \neq \pi_{i_2}$. This implies that two general situations are considered in our study: i) some information sources can be only observed by any single agent, and ii) some others can be observed by multiple (at least two) agents simultaneously. We need to point out that in the first situation, only a single agent's observation contributes to information gain of the overall network in estimation of an information source without overlapped detection. As for any other agent who is blind in observation of this information source, the sense of a fusion strategy is reduced to the point that the agent simply needs to collect the other's useful observation information diffused over the network and incorporates it into its own individual estimation, at the meanwhile keeping silence in order to avoid diffusing its blindness. By contrast, in the second situation, several agents with overlapped detection can contribute to information gain of the network in estimation of an information source in their common detection region through diffusing and fusing the multiple observation information with a certain fusion strategy. The sense of the fusion strategy lies in that an agent with overlapped detection region can combine several others' observation information with its own to enhance its own individual estimation, at the same time diffusing its own observation information for others' fusion. In this paper, we provide a unified algorithmic information fusion framework to deal with both of the two considered situations.

Furthermore, we point out that the information released by the sources can be some parameters of interest such as temperature field, multi-target locations, or some phenomenons of interest in the corresponding subregions. We divide the detection time into discrete time intervals $t \in \mathbb{Z}_{\geq 0}$. Considering the dynamic and noisy nature of information detected by agents, we

formulate the detection signal at time interval t corresponding to an information source $k \in \Pi$ as a random variable $v_k(t)$, such that its time-dependent variability can be modeled by a certain discrete probability distribution. The ranges of the detection signal $v_k(t)$ are represented as various information states. Let the total number of the information states associated with the source k be L_k. The l-th information state is denoted by $S_{k,l} = [a_{k,l}, b_{k,l})$ for $l = 1, \ldots, L_k - 1$, while $S_{k,L_k} = [a_{k,L_k}, b_{k,L_k}]$. $a_{k,l}$ and $b_{k,l}$ are the lower and the upper bounds of the detection signal $v_k(t)$ in the state $S_{k,l}$, which satisfy $b_{k,l} = a_{k,l+1}$ for $l = 1, \ldots, L_k - 1$. Each information state $S_{k,l}$ is associated with a certain probability, denoted by $p(v_k(t) \in S_{k,l})$, which indicates the possibility of the detection signal $v_k(t)$ currently ranging within $S_{k,l}$. Hence, for any $k \in \Pi$, we have $\sum_{l=1}^{L_k} p(v_k(t) \in S_{k,l}) = 1$. The objective of the information estimation over the multi-agent network is to enable each individual agent to approach the actual probability distributions of information states of the entire surveillance region, $\{p(v_k(t) \in S_{k,l})|l = 1, 2, \ldots, L_k; \forall k \in \Pi\}$, through local measurements, estimations and interactions.

Define by $v_k(i,t)$ the detection signal of an agent i received from the information source $k \in \pi_i$ at any time interval t. Then, this agent is assumed to take multiple measurements on $v_k(i,t)$ during this time interval. We collect these real-time measurements into a vector $\boldsymbol{v}_k(i,t) = \{v_{k,s}(i,t)|s = 1, 2, ..., V_i\}$, where $v_{k,s}(i,t)$ is the s-th sample on the information source k and V_i denotes i's sampling number. With the individual cumulative observations $\{\boldsymbol{v}_k(i,\tau)|\tau = 0, 1, \ldots, t\}$, the agent i can plot a discrete histogram of $v_k(i,t)$, which implies a discrete conditional probability distribution of the information states of k, $\{p_{k,l}(i,t) = p(v_k(i,t) \in S_{k,l}|v_k(t) \in S_{k,l})\}$, where $\forall p_{k,l}(i,t) \geq 0$ and $\sum_{l=1}^{L_k} p_{k,l}(i,t) = 1$. Indeed, this conditional probability $p_{k,l}(i,t)$ representing the possibility that the real-time detection signal range estimated by i is in the l-th information state of k given that the actual detection signal value exactly belongs to the same state. The conditional probability $p_{k,l}(i,t)$ indicates the accuracy of observed information at i. Correspondingly, the false detection probability $\overline{p_{k,l}(i,t)}$ is

$$\overline{p_{k,l}(i,t)} = p(v_k(i,t) \in S_{k,l}|v_k(t) \in S_{k,l'}, l' \neq l)$$
$$= \frac{\sum_{\forall l' \neq l}^{L_k} p_{k,l'}(i,t)}{\sum_{\forall l' \neq l}^{L_k} 1} = \frac{1 - p_{k,l}(i,t)}{L_k - 1} \quad (1)$$

We further denote the actual probability distribution of information states of any subregion $k \in \Pi$ at time interval t as $\boldsymbol{p}_k(t) = \text{col}\{p(v_k(t) \in S_{k,l})|l = 1, \ldots, L_k\}$, and then define $\boldsymbol{p}(t) = \text{col}\{\boldsymbol{p}_k(t)|k = 1, \ldots, m\}$ to collect all of the probability distributions. Similarly, for any individual $i \in \mathcal{V}$, we denote $\boldsymbol{p}_k(i,t) = \text{col}\{p(v_k(i,t) \in S_{k,l}|v_k(t) \in S_{k,l})|l = 1, \ldots, L_k\}$, and $\boldsymbol{p}(i,t) = \text{col}\{\boldsymbol{p}_k(i,t)|k = 1, \ldots, m\}$. In addition, we use q to denote the total number of information states of the entire region, i.e., $q = \sum_{k=1}^{m} L_k$. We note that the sizes of $\boldsymbol{p}_k(t)$ and $\boldsymbol{p}_k(i,t)$ are identical to L_k, while the sizes of $\boldsymbol{p}(t)$ and $\boldsymbol{p}(i,t)$ are identical to q.

III. INFORMATION ESTIMATION BASED ON INDIVIDUAL OBSERVATION

The agents can yield the real-time posterior information by combining the information accuracy of their real-time observations, $p_{k,l}(i,t)$, based on the well-known Bayesian rule, and have

$$p(v_k(t) \in S_{k,l}|v_k(i,t) \in S_{k,l})$$
$$= \frac{p(v_k(i,t) \in S_{k,l}|v_k(t) \in S_{k,l})p(v_k(t) \in S_{k,l})}{p(v_k(i,t) \in S_{k,l}|v_k(t) \in S_{k,l})p(v_k(t) \in S_{k,l}) + \Delta_{k,l}(i,t)}$$
$$= \frac{p_{k,l}(i,t)p(v_k(t) \in S_{k,l})}{p_{k,l}(i,t)p_{k,l}(v_k(t) \in S_{k,l}) + \Delta_{k,l}(i,t)}$$
$$\qquad (2)$$

where $\Delta_{k,l}(i,t)$ is:

$$\Delta_{k,l}(i,t)$$
$$= p(v_k(i,t) \in S_{k,l}|v_k(t) \in S_{k,l'}, l' \neq l)p(v_k(t) \in S_{k,l}, l' \neq l)$$
$$= \sum_{\forall l' \neq l}^{L_k} p(v_k(i,t) \in S_{k,l}|v_k(t) \in S_{k,l'})p(v_k(t) \in S_{k,l'})$$
$$\qquad (3)$$

By using (1), $\Delta_i^k(n,t)$ can be further expressed as:

$$\Delta_{k,l}(i,t) = \sum_{\forall l' \neq l}^{L_k} \overline{p_{k,l}(i,t)}p(v_k(t) \in S_{k,l'})$$
$$= \frac{1 - p_{k,l}(i,t)}{L_k - 1} \sum_{\forall l' \neq l}^{L_k} p(v_k(t) \in S_{k,l'}) \quad (4)$$
$$= \frac{1 - p_{k,l}(i,t)}{L_k - 1} (1 - p(v_k(t) \in S_{k,l}))$$

Generally, because of the existence of noises in agents' observations, the actual probability distribution of information states corresponding to any $k \in \Pi$, $\{p(v_k(t) \in S_{k,l})|l = 1, \ldots, L_k\}$, is unknown to these agents. At this point, it is unpractical to directly apply the equation (2) to distributed estimation since the calculation of this formula requires the exact knowledge of the parameter $p(v_k(t) \in S_{k,l})$. To obtain the current posterior information, we adopt the recursive Bayesian updating method to combine the past posterior information and the current observation information. For simplicity, we denote i's real-time estimation on the posterior probability by $\phi_{k,l}(i,t)$, i.e., $\phi_{k,l}(i,t) = p(v_k(t) \in S_{k,l}|v_k(i,t) \in S_{k,l})$. Then, we use the previous posterior estimation, represented by $\phi_{k,l}(i,t-1)$, to substitute the unknown $p(v_k(t) \in S_{k,l})$ in the right term of (2) and get

$$\phi_{k,l}(i,t)$$
$$= \frac{p_{k,l}(i,t)\phi_{k,l}(i,t-1)}{p_{k,l}(i,t)\phi_{k,l}(i,t-1) + \frac{1-p_{k,l}(i,t)}{L_k-1}(1 - \phi_{k,l}(i,t-1))}$$
$$\qquad (5)$$

In this paper, we generalize the term $p_{k,l}(i,t)$ in equation (5) such that it can reflect any distribution pattern captured by the individual observation information. The essential concept of (5) is that it allows us to effectively incorporate the past estimated information, $\phi_{k,l}(i,t-1)$, and the real-time observed information, $p_{k,l}(i,t)$, into the current individual estimation,

$\phi_{k,l}(i,t)$. To analyze the convergence of the Bayesian updating (5), we rearrange the equation (5) as

$$
\left(\frac{1}{\phi_{k,l}(i,t)} - 1 \right)
$$
$$
= \frac{1}{L_k - 1} \left(\frac{1 - p_{k,l}(i,t)}{p_{k,l}(i,t)} \right) \left(\frac{1}{\phi_{k,l}(i,t-1)} - 1 \right) \qquad (6)
$$

and further derive a closed-form expression on $\phi_{k,l}(i,t)$:

$$
\phi_{k,l}(i,t) = \frac{1}{1 + \left(\frac{1 - \phi_{k,l}(i,1)}{\phi_{k,l}(i,1)} \right) \left(\frac{1}{L_k - 1} \right)^{t-1} \prod_{\tau=2}^{t} \frac{1 - p_{k,l}(i,\tau)}{p_{k,l}(i,\tau)}} \qquad (7)
$$

where $t \geq 2$. By introducing an auxiliary parameter $a(t)$

$$
a(t) = \frac{1 - p_{k,l}(i,t)}{p_{k,l}(i,t) (L_k - 1)} \qquad (8)
$$

we can simplify (7) as

$$
\phi_{k,l}(i,t) = \frac{1}{1 + \left(\frac{1 - \phi_{k,l}(i,1)}{\phi_{k,l}(i,1)} \right) \prod_{\tau=2}^{t} a(\tau)} \qquad (9)
$$

From (9), it can be found that the accuracy of the individual real-time estimation, $\phi_{k,l}(i,t)$, depends on the number of the information states of the subregion k, L_k, and the accuracy of the individual observation, $p_{k,l}(i,t)$. Specifically, the influences of L_k and $p_{k,l}(i,t)$ on the convergence of $\phi_{k,l}(i,t)$ are summarized as the **Lemma 1**.

Lemma 1 (The convergence of the individual Bayesian updating): Given the individual prior information state probability of any agent $i \in \mathcal{V}$, $p_{k,l}(i,t) \in (0,1)$ for $l = 1, \ldots, L_k$, and the finite number of information states of any $k \in \Pi$, $L_k \geq 2$, the following conclusions are held when calculating (9):

1) If $\frac{1}{L_k} < p_{k,l}(i,t)$ for $\forall t$, $\lim_{t \to +\infty} \phi_{k,l}(i,t) = 1$ holds.

2) If $\frac{1}{L_k} = p_{k,l}(i,t)$ for $\forall t$, $\phi_{k,l}(i,t)$ does not converge. Instead, it constantly equals to the initial posterior information estimation over all time intervals, i.e., $\phi_{k,l}(i,1) = \phi_{k,l}(i,t)$ for $\forall t$.

3) If $\frac{1}{L_k} > p_{k,l}(i,t)$ for $\forall t$, $\lim_{t \to +\infty} \phi_{k,l}(i,t) = 0$ holds.

Proof: The conditions required in the three cases correspondingly result in three possible parameter $a(t)$: $0 < a(t) < 1$, $a(t) = 1$, and $a(t) > 1$ for $\forall t$. Then, we examine the time-dependent cumulative product on $a(t)$ under the three cases:

$$
\lim_{t \to +\infty} \prod_{\tau=2}^{t} a(\tau) = \begin{cases} 0, & 0 < a(\tau) < 1; \\ 1, & a(\tau) = 1; \\ +\infty, & a(\tau) > 1 \end{cases} \qquad (10)
$$

Substituting the results in (10) into (9) leads to the result. ∎

From **Lemma 1**, it can be seen that the individual estimation can not be improved despite of its further measurements if the observed information accuracy stays at uniform level, i.e., $p_{k,l}(i,t) = \frac{1}{L_k}$ for $\forall t$. Besides, if the quality of the individual observed information from the individual measurements is good enough, i.e., $p_{k,l}(i,t) > \frac{1}{L_k}$, the individual estimation can converge to 1. Otherwise, it leads to failure in the estimation of the information state probability distribution.

IV. ADAPTIVE FUSION STRATEGY

A. Global Optimization Model

As discussed in Section II, some subregions may be out of the detection range of an agent i, which can be lumped in a set $\bar{\pi}_i = \Pi - \pi_i$. The prior information corresponding to the subregion $k' \in \bar{\pi}_i$, $p_{k',l}(i,t)$, can not be obtained from the individual measurements. From **Lemma 1**, it can be found that the recursive Bayesian updating scheme based on (5) can not be implemented in this situation. Therefore, a distributed cooperative solution for information sharing and fusion among local agents is needed to improve the accuracy of individual observed information. In order to propose a cooperative distributed information estimation, we first develop a nonlinear constrained distributed optimization model. Since we formulate the information of interest over the geographic region by a series of finite discrete probability distributions of information states of subregions, we can model the optimization objective based on the information theory. Specifically, the *Jensen-Shannon* divergence, as an information metric (also called information radius (IRad) [25]), is adopted to represent an estimation objective. It can measure the disparity between two finite random graphs and can reflect the mutual information between two related random variables [22]. Given a finite discrete probability distribution $\boldsymbol{p} = (p_1, p_2, \ldots, p_U)^{\mathrm{T}}$ where $\sum_{u=1}^{U} p_u = 1$ and $\forall p_u > 0$, the amount of uncertainty of this distribution $\boldsymbol{p}$, namely the entropy, can be calculated based on Shannon's information entropy function $E(\boldsymbol{p})$ [26]:

$$
E(\boldsymbol{p}) = -\sum_{u=1}^{U} (p_u \log_2 p_u) \qquad (11)
$$

From Jensen's inequality theorem, it shows that this Shannon information entropy (11) is a concave function of the multiple probabilities $p_1, p_2, \ldots, p_U$ [22]. However, the equation (11) may not be applied under some discrete probability distributions. For example, when one entry in $\boldsymbol{p}$ is equal to zero, i.e., $p_u = 0$, (11) is not valid for numerical computation. Hence, to overcome the drawback of the logarithmic function $\log(\cdot)$, we do not directly adopt (11) in our following mathematical model. Instead, we establish a modified Shannon entropy by introducing a parameter ϵ into (11):

$$
H(\boldsymbol{p}) = -\sum_{u=1}^{U} ((p_u + \epsilon) \log_2(p_u + \epsilon)) \qquad (12)
$$

where $\epsilon \in (0,1)$ is a positive constant but should be small enough. In (12), the range of the value of any p_u is expanded to be $[0,1]$ rather than $(0,1)$.

Letting $\boldsymbol{\omega}_k = (\omega_{k,1}, \omega_{k,2})^{\mathrm{T}}$ where $\omega_{k,1}$ and $\omega_{k,2}$ are weights of the two information state probability distributions $\boldsymbol{p}_k(t)$ and $\boldsymbol{p}_k(i,t)$, respectively, satisfying $\omega_{k,1} + \omega_{k,2} = 1$ and $\omega_{k,1}, \omega_{k,2} \geq 0$, we can define the Jensen-Shannon divergence of weights $\boldsymbol{\omega}_k$ between $\boldsymbol{p}_k(t)$ and $\boldsymbol{p}_k(i,t)$ by [22]:

$$
\begin{aligned}
& JSD_k \left(\boldsymbol{p}_k(t), \boldsymbol{p}_k(i,t) \right) \\
& = H \left(\omega_{k,1} \boldsymbol{p}_k(t) + \omega_{k,2} \boldsymbol{p}_k(i,t) \right) \\
& \quad - \left(\omega_{k,1} H(\boldsymbol{p}_k(t)) + \omega_{k,2} H(\boldsymbol{p}_k(i,t)) \right)
\end{aligned} \qquad (13)
$$

Based on the Jensen-Shannon divergence (13), we further formulate an individual objective function for any agent i as follows:

$$f_i(\boldsymbol{p}(t)) = \sum_{\forall k \in \Pi} JSD_k\left(\boldsymbol{p}_k(t), \boldsymbol{p}_k(i,t)\right) \qquad (14)$$

Subsequently, a global objective function can be defined by collecting all the individual components:

$$f(\boldsymbol{p}(t)) = \sum_{i=1}^{n} f_i(\boldsymbol{p}(t)) \qquad (15)$$

For any $k \in \Pi$, the probabilities in $\boldsymbol{p}_k(t)$ should satisfy an equality constraint, $\sum_{l=1}^{L_k} p(v_k(t) \in S_{k,l}) = 1$, and inequality constraints, $\forall p(v_k(t) \in S_{k,l}) \geq 0$. To provide the compact forms of these constraints, we introduce an equality constraint coefficient matrix, $\boldsymbol{C} = \text{diag}\left\{\mathbf{1}_{L_1}, \mathbf{1}_{L_2}, \ldots, \mathbf{1}_{L_m}\right\}^{\mathrm{T}}$, and an inequality constraint coefficient matrix, $\boldsymbol{E} = \boldsymbol{I}_q$. It should be noted that the dimension of $\boldsymbol{C}$ is $m \times q$ while $\boldsymbol{E}$ is indeed a $q \times q$ identity matrix. The constraints on $\boldsymbol{p}(t)$ are defined as:

$$s.t. \begin{cases} \boldsymbol{E}\boldsymbol{p}(t) \geq \mathbf{0}_q \\ \boldsymbol{C}\boldsymbol{p}(t) = \mathbf{1}_m \end{cases} \qquad (16)$$

where the inequality constraint is element-wise.

Because the Jensen-Shannon divergence is a nonnegative measure [22], $f(\boldsymbol{p}(t))$ is also a nonnegative real-value function, i.e., $f(\boldsymbol{p}(t)) \geq 0$ for any input $\boldsymbol{p}(t)$. Additionally, the smaller the Jensen-Shannon divergence $f(\boldsymbol{p}(t))$ is, the less the difference between the actual and the observed information distributions achieves. That is, the function $f(\boldsymbol{p}(t))$ can reach zero, if and only if any observed distribution $\boldsymbol{p}_k(i,t)$ at the individual agent i totally matches the actual distribution $\boldsymbol{p}_k(t)$. Therefore, treating the unknown $\boldsymbol{p}(t)$ as the decision variable and the prior probability distributions estimated from the observations, $\{\boldsymbol{p}(i,t)|i \in \mathcal{V}\}$, as input parameters, we can develop an optimization model where the Jensen-Shannon divergence based objective function (15) is expected to be minimized under the constraints (16):

$$\min \ f(\boldsymbol{p}(t)) = \sum_{i=1}^{n} f_i(\boldsymbol{p}(t))$$
$$s.t. \begin{cases} \boldsymbol{E}\boldsymbol{p}(t) \geq \mathbf{0}_q \\ \boldsymbol{C}\boldsymbol{p}(t) = \mathbf{1}_m. \end{cases} \qquad (17)$$

From (17), we see that the optimization model has linear constraints while its objective function is nonlinear. The following **Lemma 2** shows the convexity of its objective function $f(\boldsymbol{p}(t))$.

Lemma 2 (The convexity of the optimization objective function): Given $\boldsymbol{\omega}_k = (\omega_{k,1}, \omega_{k,2})^{\mathrm{T}}$ satisfying $\omega_{k,1} + \omega_{k,2} = 1$ and $\omega_{k,1}, \omega_{k,2} \geq 0$ for $k = 1, \ldots, m$, the optimization objective $f(\boldsymbol{p}(t))$ in the global model (17) is a strictly convex function of $\boldsymbol{p}(t)$.

Proof: According to the Lemma 1 in the work [27], the Jensen-Shannon divergence based function is strictly convex, so that the objective function $f(\boldsymbol{p}(t))$ summing the Jensen-Shannon divergence functions of all the agents must be also strictly convex. Thus, **Lemma 2** is proved. ∎

B. Local Optimization Model

From the **Lemma 2**, it can be found that if an algorithm existing for the nonlinear optimization model can converge to a stationary point of $f(\boldsymbol{p}(t))$, this algorithm can converge to a global minimum of the model. In the application context of information estimation, such a global minimum indeed corresponds to the actual probability distribution of information states of the entire region. Hence, we consider that every individual agent's goal is to approach the same actual global distribution, denoted by $\boldsymbol{p}^*(t)$. As each individual has a common goal (determining the global distribution $\boldsymbol{p}^*(t)$), they are expected to share local independent observed information and perform local cooperative interaction with other neighbors. Two essential issues arises when a distributed processing is considered: 1) *how to enable each agent to adapt their individual estimation in real time according to its own and neighbors' continuous measurements*; 2) *how to enable a better local fusion of each agent's and its neighbors' information to improve individual estimation performance rather than solely solving the global $\boldsymbol{p}^*(t)$ on its own information*. To address these issues, we first propose a localized Jensen-Shannon based objective function depending on interactions among the neighboring agents. Then, each agent can minimize the localized objective function under the same constraints in the global model (17) through processing a gradient-projection procedure. Finally, an adaptive fusion strategy also based on Jensen-Shannon divergence is proposed to combine the local intermediate estimations of each agent and its neighbors.

Let $\mathcal{V}_i$ denote the immediate neighborhood of an agent i (including the agent i itself). By introducing some spatial coefficients $\{x_{j,i}\}$ $(j, i \in \mathcal{V})$, we present an objective function for the agent i via a weighted sum strategy as:

$$g_i(\boldsymbol{p}(t)) = \sum_{j=1}^{n} x_{j,i} f_j(\boldsymbol{p}(t)) \qquad (18)$$

where the coefficients $\{x_{j,i}\}$ are nonnegative and satisfy $\sum_{i=1}^{n} x_{j,i} = 1$ and $x_{j,i} = 0$ if and only if $j \notin \mathcal{V}_i$. Considering that each agent can only exchange its information with its immediate neighbors $\mathcal{V}_i$, we define $g_i(\boldsymbol{p}(t))$ as an optimization objective for the individual i. Thus, the local optimization model is defined as:

$$\min \ g_i(\boldsymbol{p}(t)) = \sum_{j=1}^{n} x_{j,i} f_j(\boldsymbol{p}(t))$$
$$s.t. \begin{cases} \boldsymbol{E}\boldsymbol{p}(t) \geq \mathbf{0}_q \\ \boldsymbol{C}\boldsymbol{p}(t) = \mathbf{1}_m \end{cases} \qquad (19)$$

where the local objective function $g_i(\boldsymbol{p}(t))$ combines the neighbors' individual cost functions. This enables the intermediate estimation information to be diffused among neighboring agents in the network. Iteratively solving the localized model (19) at every i can lead to a continuous local information diffusion over the multi-agent network, which turns out to improve the individual information estimation. Additionally,

we remark that the nonnegative coefficients $\{x_{j,i}\}$ can lead to

$$\begin{aligned}
\sum_{i=1}^{n} g_i(\boldsymbol{p}(t)) &= \sum_{i=1}^{n}\sum_{j=1}^{n} x_{j,i} f_j(\boldsymbol{p}(t)) \\
&= \sum_{j=1}^{n}\sum_{i=1}^{n} x_{j,i} f_j(\boldsymbol{p}(t)) \\
&= \sum_{j=1}^{n} f_j(\boldsymbol{p}(t))
\end{aligned} \tag{20}$$

which shows that the sum of all individual objective functions $\{g_i(\boldsymbol{p}(t))\}$ is identical to the global objective function $f(\boldsymbol{p}(t))$. Since local observed information is diffused over the network and fused at individual nodes to enhance their $\boldsymbol{p}(i,t)$, these individuals can gradually arrive at a consensus on the probability distributions of information states of the whole surveillance region. That is, their $\boldsymbol{p}(i,t)$ is expected to approach the common distribution under $t \to +\infty$. Note the local objective function $g_i(\boldsymbol{p}(t))$ is constructed with local neighbors' $\boldsymbol{p}(i,t)$ and the global objective $f(\boldsymbol{p}(t))$ in (17) indeed collects every individual objectives $\{g_i(\boldsymbol{p}(t))\}$ (shown in (20)). Along with $t \to +\infty$, minimizing the global objective function in (17) is approximately equivalent to minimizing the local objective (18) at every node of the network $\mathcal{G}(\mathcal{V},\mathcal{E})$ in a decentralized manner.

C. Gradient Projection Solution

Each agent can solve the local optimization model (19) by the gradient projection algorithm. Let $\boldsymbol{p}_{i,\bar{t}}(t)$ be the agent's individual estimation of the global $\boldsymbol{p}^*(t)$ at the iteration $\bar{t}$, i.e., a feasible iterator for the local model (19). Without loss of generality, at any $\boldsymbol{p}_{i,\bar{t}}(t)$, the inequality constraints $\boldsymbol{E}\boldsymbol{p}_{i,\bar{t}}(t) \geq \boldsymbol{0}_q$ can be decomposed into two parts, one of which is called the active constraints, represented by $\boldsymbol{E}_{i,1}\boldsymbol{p}_{i,\bar{t}}(t) = \boldsymbol{0}_{q_{i,1}}$, another is called the nonactive constraints, represented by $\boldsymbol{E}_{i,2}\boldsymbol{p}_{i,\bar{t}}(t) > \boldsymbol{0}_{q_{i,2}}$. The coefficient matrices $\boldsymbol{E}_{i,1}$ and $\boldsymbol{E}_{i,2}$ are sub-blocks of the matrix $\boldsymbol{E}$, i.e., $\boldsymbol{E} = \mathrm{col}\{\boldsymbol{E}_{i,1}, \boldsymbol{E}_{i,2}\}$. The full-zero vectors $\boldsymbol{0}_{q_{i,1}}$ and $\boldsymbol{0}_{q_{i,2}}$ are also sub-blocks of $\boldsymbol{0}_q$, i.e., $\boldsymbol{0}_q = \mathrm{col}\{\boldsymbol{0}_{q_{i,1}}, \boldsymbol{0}_{q_{i,2}}\}$, where the dimensions satisfy $q_{i,1} + q_{i,2} = q$. We further point out that since all of the probabilities in a discrete distribution cannot be zero simultaneously (the sum of their values constantly equals to 1), all the inequality constraints can not be active simultaneously. That is, the dimension of $\boldsymbol{E}_{i,1}$ can not be q, i.e., $q_{i,1} < q$ being always held. In fact, because there totally exist m discrete information state distributions, we can see $q_{i,2} \geq m$, which equivalently implies $q_{i,1} \leq (q-m)$. With the active constraint coefficient matrix $\boldsymbol{E}_{i,1}$ and the equality constraint coefficient matrix $\boldsymbol{C}$, we can construct a new constraint matrix associated with the iterator $\boldsymbol{p}_{i,\bar{t}}(t)$:

$$\boldsymbol{M}_{i,\bar{t}}(t) = \begin{bmatrix} \boldsymbol{E}_{i,1} \\ \boldsymbol{C} \end{bmatrix} \tag{21}$$

where the dimension of $\boldsymbol{M}_{i,\bar{t}}(t)$ is $(q_{i,1}+m) \times q$.

The basic idea behind the gradient-projection iterative scheme is that a new iterator is generated in a feasible direction starting from the current feasible iterator. When the current feasible iterator is within the feasible region, the negative gradient direction can be employed for searching a new point; otherwise, when the current iterator is on the boundary of the feasible region, a new feasible direction is generated by projecting the negative gradient direction at the current point to the null space constituted by the active constraints. Thus, considering $\boldsymbol{M}_{i,\bar{t}}(t)$ with full row rank, we are allowed to establish another new matrix $\boldsymbol{P}_{i,\bar{t}}(t)$, called projection matrix

$$\boldsymbol{P}_{i,\bar{t}}(t) = \boldsymbol{E} - \left(\boldsymbol{M}_{i,\bar{t}}(t)\right)^{\mathrm{T}}\left(\boldsymbol{M}_{i,\bar{t}}(t)\left(\boldsymbol{M}_{i,\bar{t}}(t)\right)^{\mathrm{T}}\right)^{-1}\boldsymbol{M}_{i,\bar{t}}(t) \tag{22}$$

It is worth pointing out that when the matrix $\boldsymbol{M}_{i,\bar{t}}(t)$ is empty, we can simply set $\boldsymbol{P}_{i,\bar{t}}(t) = \boldsymbol{E}$.

Once the projection matrix $\boldsymbol{P}_{i,\bar{t}}(t)$ is achieved at the agent i, it can apply a steepest-descent iterative method to optimize their individual objective functions $g_i(\boldsymbol{p}(t))$ in a negative direction of projected gradient $\boldsymbol{d}_{i,\bar{t}}(t)$:

$$\boldsymbol{d}_{i,\bar{t}}(t) = -\boldsymbol{P}_{i,\bar{t}}(t)\nabla_{\boldsymbol{p}(t)}g_i(\boldsymbol{p}_{i,\bar{t}}(t))) \tag{23}$$

where $\nabla_{\boldsymbol{p}(t)}g_i(\boldsymbol{p}_{i,\bar{t}}(t))$ denotes the gradient of the individual objective function $g_i(\boldsymbol{p}(t))$ evaluated at the point $\boldsymbol{p}_{i,\bar{t}}(t)$. Based on the equation (23), we can get a new iterator at the current point $\boldsymbol{p}_{i,\bar{t}}(t)$ by

$$\boldsymbol{u}_{i,\bar{t}+1}(t) = \boldsymbol{p}_{i,\bar{t}}(t) + \lambda_i \boldsymbol{d}_{i,\bar{t}}(t) \tag{24}$$

where the parameter $\lambda_i > 0$ is a nonnegative step size.

According to the equation (24), the iterative procedure can be proceeded only when $\boldsymbol{d}_{i,\bar{t}}(t) \neq \boldsymbol{0}$. To solve the problem when $\boldsymbol{d}_{i,\bar{t}}(t) = \boldsymbol{0}$, we define an auxiliary vector $\boldsymbol{s}_{i,\bar{t}}(t)$ as

$$\boldsymbol{s}_{i,\bar{t}}(t) = \left(\boldsymbol{M}_{i,\bar{t}}(t)\left(\boldsymbol{M}_{i,\bar{t}}(t)\right)^{\mathrm{T}}\right)^{-1}\boldsymbol{M}_{i,\bar{t}}(t)\nabla_{\boldsymbol{p}(t)}g_i(\boldsymbol{p}_{i,\bar{t}}(t)) \tag{25}$$

Then, we divide this vector $\boldsymbol{s}_{i,\bar{t}}(t)$ into two sub-blocks as follows

$$\boldsymbol{s}_{i,\bar{t}}(t) = \begin{bmatrix} \boldsymbol{s}_{i,\bar{t},1}(t) \\ \boldsymbol{s}_{i,\bar{t},2}(t) \end{bmatrix} \tag{26}$$

where the row indexes of $\boldsymbol{s}_{i,\bar{t},1}(t)$ and $\boldsymbol{s}_{i,\bar{t},2}(t)$ correspond to those of the blocks $\boldsymbol{E}_{i,1}$ and $\boldsymbol{C}$ in $\boldsymbol{M}_{i,\bar{t}}(t)$, respectively. The **Lemma 3** is presented for proceeding the iteration when $\boldsymbol{d}_{i,\bar{t}}(t) = \boldsymbol{0}$.

Lemma 3 (Conditions on gradient projection matrix for iteration): Given that the active-constraint coefficient matrix $\boldsymbol{M}_{i,\bar{t}}(t)$, the projection matrix $\boldsymbol{P}_{i,\bar{t}}(t)$ and the iteration direction $\boldsymbol{d}_{i,\bar{t}}(t)$ are derived from (21), (22) and (23), respectively, and $\boldsymbol{d}_{i,\bar{t}}(t) = \boldsymbol{0}$ is satisfied at the iterator $\boldsymbol{p}_{i,\bar{t}}(t)$, the following two conclusions are held:

1) If $\boldsymbol{s}_{i,\bar{t},1}(t)$ is element-wisely nonnegative, i.e., $\boldsymbol{S}_{i,\bar{t},1}(t) \geq \boldsymbol{0}$, then the current point $\boldsymbol{p}_{i,\bar{t}}(t)$ is a minimizer for the optimization model (19);

2) If $\boldsymbol{s}_{i,\bar{t},1}(t)$ has at least a negative element, denoted by $s_{i,\bar{t},1}^r(t) < 0$ where r is its row index, then one can remove the r-th row from $\boldsymbol{E}_{i,1}$ to get a new active inequality constraint coefficient matrix $\widehat{\boldsymbol{E}_{i,1}}$ and to construct a new active constraint matrix $\widehat{\boldsymbol{M}_{i,\bar{t}}(t)}$ by $\widehat{\boldsymbol{M}_{i,\bar{t}}(t)} = \mathrm{col}\left\{\widehat{\boldsymbol{E}_{i,1}}, \boldsymbol{C}\right\}$. With the new $\widehat{\boldsymbol{M}_{i,\bar{t}}(t)}$, a new projection matrix $\widehat{\boldsymbol{P}_{i,\bar{t}}(t)}$ can also be

established by (22):

$$\widehat{\boldsymbol{P}_{i,\bar{t}}(t)}$$
$$= \boldsymbol{E} - \left(\widehat{\boldsymbol{M}_{i,\bar{t}}(t)}\right)^{\mathrm{T}} \left(\widehat{\boldsymbol{M}_{i,\bar{t}}(t)} \left(\widehat{\boldsymbol{M}_{i,\bar{t}}(t)}\right)^{\mathrm{T}}\right)^{-1} \left(\widehat{\boldsymbol{M}_{i,\bar{t}}(t)}\right) \tag{27}$$

then $\boldsymbol{d}_{i,\bar{t}}(t)$ can be re-calculated by:

$$\boldsymbol{d}_{i,\bar{t}}(t) = -\widehat{\boldsymbol{P}_{i,\bar{t}}(t)} \nabla_{\boldsymbol{p}(t)} g_i(\boldsymbol{p}_{i,\bar{t}}(t)) \tag{28}$$

such that it can satisfy $\boldsymbol{d}_{i,\bar{t}}(t) \neq \boldsymbol{0}$ to proceed the iteration (24).

Proof: When $\boldsymbol{s}_{i,\bar{t},1}(t) \geq \boldsymbol{0}$ and $\boldsymbol{d}_{i,\bar{t}}(t) = \boldsymbol{0}$, we can see

$$\boldsymbol{0}$$
$$= \left(\boldsymbol{E} - \left(\boldsymbol{M}_{i,\bar{t}}(t)\right)^{\mathrm{T}} \left(\boldsymbol{M}_{i,\bar{t}}(t) \left(\boldsymbol{M}_{i,\bar{t}}(t)\right)^{\mathrm{T}}\right)^{-1} \boldsymbol{M}_{i,\bar{t}}(t)\right)$$
$$\nabla_{\boldsymbol{p}(t)} g_i(\boldsymbol{p}_{i,\bar{t}}(t))$$
$$= \nabla_{\boldsymbol{p}(t)} g_i(\boldsymbol{p}_{i,\bar{t}}(t)) - \left(\boldsymbol{M}_{i,\bar{t}}(t)\right)^{\mathrm{T}} \left(\boldsymbol{M}_{i,\bar{t}}(t) \left(\boldsymbol{M}_{i,\bar{t}}(t)\right)^{\mathrm{T}}\right)^{-1}$$
$$\boldsymbol{M}_{i,\bar{t}}(t) \nabla_{\boldsymbol{p}(t)} g_i(\boldsymbol{p}_{i,\bar{t}}(t))$$
$$= \nabla_{\boldsymbol{p}(t)} g_i(\boldsymbol{p}_{i,\bar{t}}(t)) - \left(\boldsymbol{E}_{i,1}\right)^{\mathrm{T}} \boldsymbol{s}_{i,\bar{t},1}(t) - \left(\boldsymbol{C}\right)^{\mathrm{T}} \boldsymbol{s}_{i,\bar{t},2}(t) \tag{29}$$

The equation (29) is exactly the Kuhn-Tucker condition [28]. Thus, under the condition of 1) in **Lemma 3**, $\boldsymbol{p}_{i,\bar{t}}(t)$ is indeed a local minimizer of the optimization model (19). Recall that **Lemma 2** states the convexity of the Jensen-Shannon based objective function. The model (19) has a convex objective function and linear constraints, which indicates that $\boldsymbol{p}_{i,\bar{t}}(t)$ is also a global optimum of this model.

Otherwise, there exists at least an element $s_{i,\bar{t},1}^r(t)$ in $\boldsymbol{s}_{i,\bar{t},1}(t)$ that is negative, i.e., $s_{i,\bar{t},1}^r(t) < 0$. Hence, we denote the corresponding r-th row of $\boldsymbol{E}_{i,1}$ as $\boldsymbol{E}_{i,1}^r$. To proceed this proof, we turn to induce a contradiction given $\widehat{\boldsymbol{P}_{i,\bar{t}}(t)} \nabla_{\boldsymbol{p}(t)} g_i(\boldsymbol{p}_{i,\bar{t}}(t)) = \boldsymbol{0}$.

It can be found that (noting $\widehat{\boldsymbol{M}_{i,\bar{t}}(t)} = \mathrm{col}\left\{\widehat{\boldsymbol{E}_{i,1}}, \boldsymbol{C}\right\}$)

$$\left(\boldsymbol{E}_{i,1}\right)^{\mathrm{T}} \boldsymbol{s}_{i,\bar{t}}(t) + \boldsymbol{C}^{\mathrm{T}} \boldsymbol{s}_{i,\bar{t}}(t)$$
$$= \left(\widehat{\boldsymbol{E}_{i,1}}\right)^{\mathrm{T}} \widehat{\boldsymbol{s}_{i,\bar{t},1}(t)} + s_{i,\bar{t},1}^r(t) \left(\boldsymbol{E}_{i,1}^r\right)^{\mathrm{T}} + \boldsymbol{C}^{\mathrm{T}} \boldsymbol{s}_{i,\bar{t},2}(t) \tag{30}$$
$$= \left(\widehat{\boldsymbol{M}_{i,\bar{t}}(t)}\right)^{\mathrm{T}} \overline{\boldsymbol{s}_{i,\bar{t}}(t)} + s_{i,\bar{t},1}^r(t) \left(\boldsymbol{E}_{i,1}^r\right)^{\mathrm{T}}$$

where $\widehat{\boldsymbol{s}_{i,\bar{t},1}(t)}$ is composed of all the entries in $\boldsymbol{s}_{i,\bar{t},1}(t)$ except the r-th one, and $\overline{\boldsymbol{s}_{i,\bar{t}}(t)}$ is composed of $\widehat{\boldsymbol{s}_{i,\bar{t},1}(t)}$ and $\boldsymbol{s}_{i,\bar{t},2}(t)$. Substituting (30) into (29) gets

$$\boldsymbol{0} = \nabla_{\boldsymbol{p}(t)} g_i(\boldsymbol{p}_{i,\bar{t}}(t)) - \left(\widehat{\boldsymbol{M}_{i,\bar{t}}(t)}\right)^{\mathrm{T}} \overline{\boldsymbol{s}_{i,\bar{t}}(t)} - s_{i,\bar{t},1}^r(t) \left(\boldsymbol{E}_{i,1}^r\right)^{\mathrm{T}} \tag{31}$$

Furthermore, $\widehat{\boldsymbol{P}_{i,\bar{t}}(t)} \nabla_{\boldsymbol{p}(t)} g_i(\boldsymbol{p}_{i,\bar{t}}(t)) = \boldsymbol{0}$ can lead to

$$\boldsymbol{0}$$
$$= \widehat{\boldsymbol{P}_{i,\bar{t}}(t)} \nabla_{\boldsymbol{p}(t)} g_i(\boldsymbol{p}_{i,\bar{t}}(t))$$
$$= \left(\boldsymbol{E} - \left(\widehat{\boldsymbol{M}_{i,\bar{t}}(t)}\right)^{\mathrm{T}} \left(\widehat{\boldsymbol{M}_{i,\bar{t}}(t)} \left(\widehat{\boldsymbol{M}_{i,\bar{t}}(t)}\right)^{\mathrm{T}}\right)^{-1} \widehat{\boldsymbol{M}_{i,\bar{t}}(t)}\right)$$
$$\nabla_{\boldsymbol{p}(t)} g_i(\boldsymbol{p}_{i,\bar{t}}(t))$$
$$= \nabla_{\boldsymbol{p}(t)} g_i(\boldsymbol{p}_{i,\bar{t}}(t)) - \left(\widehat{\boldsymbol{M}_{i,\bar{t}}(t)}\right)^{\mathrm{T}} \boldsymbol{s}_{i,\bar{t}}(t) \tag{32}$$

Thus, subtracting (32) from (31) yields

$$\boldsymbol{0} = \left(\widehat{\boldsymbol{M}_{i,\bar{t}}(t)}\right)^{\mathrm{T}} \left(\overline{\boldsymbol{s}_{i,\bar{t}}(t)} - \boldsymbol{s}_{i,\bar{t}}(t)\right) + s_{i,\bar{t},1}^r(t) \left(\boldsymbol{E}_{i,1}^r\right)^{\mathrm{T}} \tag{33}$$

The right side of (33) illustrates a linear combination of the row vectors of $\boldsymbol{M}_{i,\bar{t}}(t)$. Since $s_{i,\bar{t},1}^r(t) \neq 0$, the row vectors of $\boldsymbol{M}_{i,\bar{t}}(t)$ are linearly dependent. This is incompatible with the fact that $\boldsymbol{M}_{i,\bar{t}}(t)$ has full row rank. Therefore, $\widehat{\boldsymbol{P}_{i,\bar{t}}(t)} \nabla_{\boldsymbol{p}(t)} g_i(\boldsymbol{p}_{i,\bar{t}}(t)) \neq \boldsymbol{0}$, and the conclusion 2) of **Lemma 3** is proven. ∎

Indeed, **Lemma 3** gives a theoretical condition that indicates when to stop iterating at a given point. Specifically, according to the proof of **Lemma 3**, an iterator $\boldsymbol{p}_{i,\bar{t}}(t)$ satisfying both of $\boldsymbol{d}_{i,\bar{t}}(t) = \boldsymbol{0}$ and $\boldsymbol{s}_{i,\bar{t},1}(t) \geq \boldsymbol{0}$ is a Kuhn-Tucker point, it is also an optimum since the objective function of the model (19) is strictly convex as presented in **Lemma 2**.

On the other hand, since the objective function of the model (19), $g_i(\boldsymbol{p}(t))$, collects the agent i's neighboring components, $\{f_j(\boldsymbol{p}(t)) | j \in \mathcal{V}_i\}$, its gradient with respect to $\boldsymbol{p}(t)$ also combines the gradient information of the neighbors, i.e., $\nabla_{\boldsymbol{p}(t)} g_i(\boldsymbol{p}(t)) = \sum_{j \in \mathcal{V}_i} x_{j,i} \nabla_{\boldsymbol{p}(t)} f_j(\boldsymbol{p}(t))$. Recall that in the equation (14), the real-time observation information of any neighbor $j \in \mathcal{V}_i$, $\boldsymbol{p}(j,t)$, has been introduced into its Jensen-Shannon based function $f_j(\boldsymbol{p}(t))$. The agent i's gradient formula $\nabla_{\boldsymbol{p}(t)} g_i(\boldsymbol{p}(t))$ not only incorporates its own but also the neighbors' real-time observation information. In this way, the iterative procedure based on (24) can adapt the individual intermediate estimation, $\boldsymbol{u}_{i,\bar{t}+1}(t)$, to local real-time observation information. Furthermore, once the computation of (24) is accomplished at each individual agent, the agent i can enhance its estimation at iteration $(\bar{t}+1)$ by combining the intermediate estimations of all its neighbors based on a linear-weighted fusion strategy. That is, we compute i's next iterator $\boldsymbol{p}_{i,\bar{t}+1}(t)$ by

$$\boldsymbol{p}_{i,\bar{t}+1}(t) = \sum_{j=1}^n y_{j,i} \boldsymbol{u}_{j,\bar{t}+1}(t) \tag{34}$$

where $\{y_{j,i} | j = 1, \ldots, n\}$ are some non-negative weights that satisfy $\sum_{j=1}^n y_{j,i} = 1$ and $y_{j,i} = 0$ if and only if $j \notin \mathcal{V}_i$.

Essentially, the equation (34) can lead to the fusion of agents' intermediate estimation information over the network, which can further benefit the estimation performance of each individual. Combining (24) and (34) induces a distributed

cooperative estimation solution as:

$$
\begin{cases}
\boldsymbol{u}_{i,\bar{t}+1}(t) = \boldsymbol{p}_{i,\bar{t}}(t) - \lambda_i \sum_{j=1}^{n} x_{j,i} \boldsymbol{P}_{i,\bar{t}}(t) \nabla_{\boldsymbol{p}(t)} f_j(\boldsymbol{p}_{i,\bar{t}}(t)) \\
\boldsymbol{p}_{i,\bar{t}+1}(t) = \sum_{j=1}^{n} y_{j,i} \boldsymbol{u}_{j,\bar{t}+1}(t)
\end{cases}
\tag{35}
$$

for $\forall i \in \mathcal{V}$.

The mathematical structure of the iterative formulas in (35) coincides with the ATC-type (Adapt-Then-Combine) computation framework [10], [11]. The first iteration equation in (35) can be treated as the individual adaptation to real-time observation information, while the second is the combination of multiple agents' intermediate information. In (35), considering the constraints in the optimization model (19), it is required to guarantee that the estimations $\{\boldsymbol{p}_{i,\bar{t}}(t)\}$ at each iteration $\bar{t}$ is feasible. The following theorem is presented to guarantee the feasibility of the proposed iterative algorithm based on (35). Let $\widehat{\boldsymbol{b}}_{j,\bar{t}}(t) = \boldsymbol{0}_{q_{j,2}} - \boldsymbol{E}_{j,2} \boldsymbol{p}_{j,\bar{t}}(t)$ and $\widehat{\boldsymbol{d}}_{j,\bar{t}}(t) = \boldsymbol{E}_{j,2} \boldsymbol{d}_{j,\bar{t}}(t)$ for any $j \in \mathcal{V}_i$, and denote the l-th ($l = 1, \ldots, q_{j,2}$) elements of the vectors $\widehat{\boldsymbol{b}}_{j,\bar{t}}(t)$ and $\widehat{\boldsymbol{d}}_{j,\bar{t}}(t)$ by $\widehat{b}_{j,\bar{t},l}(t)$ and $\widehat{d}_{j,\bar{t},l}(t)$, respectively. We establish the following

Theorem 1 (The feasibility of the iterative algorithm): Suppose $\boldsymbol{p}_{j,\bar{t}}(t)$ is a feasible point of the optimization model (19) for $\forall j \in \mathcal{V}_i$ and $\lambda_j \in [0, \lambda_j^{\max}]$. Then, $\boldsymbol{p}_{i,\bar{t}+1}(t)$ obtained by (35) is also a feasible point. $\lambda_j^{\max}$ is defined by

$$
\lambda_j^{\max} = \begin{cases}
\min \left\{ \left. \dfrac{\widehat{b}_{j,\bar{t},l}(t)}{\widehat{d}_{j,\bar{t},l}(t)} \right| \widehat{d}_{j,\bar{t},l}(t) < 0 \right\}, & \widehat{\boldsymbol{d}}_{j,\bar{t}}(t) \not\geq \boldsymbol{0}_{q_{j,2}} \\
+\infty, & \widehat{\boldsymbol{d}}_{j,\bar{t}}(t) \geq \boldsymbol{0}_{q_{j,2}}
\end{cases}
\tag{36}
$$

Proof: Substituting the first equation of (35) into the second one leads to

$$
\boldsymbol{p}_{i,\bar{t}+1}(t) = \sum_{j=1}^{n} y_{j,i} \boldsymbol{p}_{j,\bar{t}}(t) - \sum_{j=1}^{n} y_{j,i} \lambda_j \boldsymbol{P}_{j,\bar{t}}(t) \nabla_{\boldsymbol{p}(t)} g_j(\boldsymbol{p}_{j,\bar{t}}(t))
\tag{37}
$$

The results of $\boldsymbol{E} \boldsymbol{p}_{i,\bar{t}+1}(t)$ and $\boldsymbol{C} \boldsymbol{p}_{i,\bar{t}+1}(t)$ are:

$$
\begin{aligned}
& \boldsymbol{E} \boldsymbol{p}_{i,\bar{t}+1}(t) \\
&= \sum_{j=1}^{n} y_{j,i} \boldsymbol{E} \boldsymbol{p}_{j,\bar{t}}(t) \\
& \quad - \sum_{j=1}^{n} y_{j,i} \lambda_j \boldsymbol{E} \boldsymbol{P}_{j,\bar{t}}(t) \nabla_{\boldsymbol{p}(t)} g_j(\boldsymbol{p}_{j,\bar{t}}(t)) \\
&= \sum_{j=1}^{n} y_{j,i} \boldsymbol{E} \boldsymbol{p}_{j,\bar{t}}(t) \\
& \quad - \sum_{j=1}^{n} y_{j,i} \lambda_j \begin{bmatrix} \boldsymbol{E}_{j,1} \\ \boldsymbol{E}_{j,2} \end{bmatrix} \boldsymbol{P}_{j,\bar{t}}(t) \nabla_{\boldsymbol{p}(t)} g_j(\boldsymbol{p}_{j,\bar{t}}(t)) \\
&= \sum_{j=1}^{n} y_{j,i} \boldsymbol{E} \boldsymbol{p}_{j,\bar{t}}(t) \\
& \quad - \sum_{j=1}^{n} y_{j,i} \begin{bmatrix} \lambda_j \boldsymbol{E}_{j,1} \boldsymbol{P}_{j,\bar{t}}(t) \nabla_{\boldsymbol{p}(t)} g_j(\boldsymbol{p}_{j,\bar{t}}(t)) \\ \lambda_j \boldsymbol{E}_{j,2} \boldsymbol{P}_{j,\bar{t}}(t) \nabla_{\boldsymbol{p}(t)} g_j(\boldsymbol{p}_{j,\bar{t}}(t)) \end{bmatrix}
\end{aligned}
\tag{38}
$$

$$
\begin{aligned}
& \boldsymbol{C} \boldsymbol{p}_{i,\bar{t}+1}(t) \\
&= \sum_{j=1}^{n} y_{j,i} \boldsymbol{C} \boldsymbol{p}_{j,\bar{t}}(t) - \sum_{j=1}^{n} y_{j,i} \lambda_j \boldsymbol{C} \boldsymbol{P}_{j,\bar{t}}(t) \nabla_{\boldsymbol{p}(t)} g_j(\boldsymbol{p}_{j,\bar{t}}(t))
\end{aligned}
\tag{39}
$$

Firstly, according to the definition of the projection matrix $\boldsymbol{P}_{j,\bar{t}}(t)$, we can get

$$
\begin{aligned}
& \lambda_j \boldsymbol{M}_{j,\bar{t}}(t) \boldsymbol{d}_{j,\bar{t}}(t) \\
&= -\lambda_j \boldsymbol{M}_{j,\bar{t}}(t) \boldsymbol{P}_{j,\bar{t}}(t) \nabla_{\boldsymbol{p}(t)} g_j(\boldsymbol{p}_{j,\bar{t}}(t)) \\
&= -\lambda_j \boldsymbol{M}_{j,\bar{t}}(t) \\
& \quad \left(\boldsymbol{E} - \left(\boldsymbol{M}_{j,\bar{t}}(t) \right)^{\mathrm{T}} \left(\boldsymbol{M}_{j,\bar{t}}(t) \left(\boldsymbol{M}_{j,\bar{t}}(t) \right)^{\mathrm{T}} \right)^{-1} \boldsymbol{M}_{j,\bar{t}}(t) \right) \\
& \quad \nabla_{\boldsymbol{p}(t)} g_j(\boldsymbol{p}_{j,\bar{t}}(t)) \\
&= -\lambda_j \\
& \quad \left(\boldsymbol{M}_{j,\bar{t}}(t) - \boldsymbol{M}_{j,\bar{t}}(t) \left(\boldsymbol{M}_{j,\bar{t}}(t) \right)^{\mathrm{T}} \left(\boldsymbol{M}_{j,\bar{t}}(t) \left(\boldsymbol{M}_{j,\bar{t}}(t) \right)^{\mathrm{T}} \right)^{-1} \boldsymbol{M}_{j,\bar{t}}(t) \right) \\
& \quad \nabla_{\boldsymbol{p}(t)} g_j(\boldsymbol{p}_{j,\bar{t}}(t)) \\
&= -\lambda_j \left(\boldsymbol{M}_{j,\bar{t}}(t) - \boldsymbol{M}_{j,\bar{t}}(t) \right) \nabla_{\boldsymbol{p}(t)} g_j(\boldsymbol{p}_{j,\bar{t}}(t)) \\
&= \boldsymbol{0}_{q_{j,1}+m}
\end{aligned}
\tag{40}
$$

Note $\boldsymbol{M}_{j,\bar{t}}(t) = \mathrm{col}\{\boldsymbol{E}_{j,1}, \boldsymbol{C}\}$. (40) is equivalent to

$$
\begin{cases}
\lambda_j \boldsymbol{E}_{j,1} \boldsymbol{P}_{j,\bar{t}}(t) \nabla_{\boldsymbol{p}(t)} g_j(\boldsymbol{p}_{j,\bar{t}}(t)) = \boldsymbol{0}_{q_{j,1}} \\
\lambda_j \boldsymbol{C} \boldsymbol{P}_{j,\bar{t}}(t) \nabla_{\boldsymbol{p}(t)} g_j(\boldsymbol{p}_{j,\bar{t}}(t)) = \boldsymbol{0}_m
\end{cases}
\tag{41}
$$

Substituting the second equation of (41) into (39) gets

$$
\boldsymbol{C} \boldsymbol{p}_{i,\bar{t}+1}(t) = \sum_{j=1}^{n} y_{j,i} \boldsymbol{C} \boldsymbol{p}_{j,\bar{t}}(t)
\tag{42}
$$

Since $\boldsymbol{p}_{j,\bar{t}}(t)$ is a feasible point satisfying $\boldsymbol{C} \boldsymbol{p}_{j,\bar{t}}(t) = \boldsymbol{1}_m$ and $\sum_{j=1}^{n} y_{j,i} = 1$, (42) is equivalent to $\boldsymbol{C} \boldsymbol{p}_{i,\bar{t}+1}(t) = \boldsymbol{1}_m$, which indicates that the new iterator $\boldsymbol{p}_{i,\bar{t}+1}(t)$ satisfies the equality constraints of the model (17).

To investigate the equation (38), we consider two cases:

1) If $\widehat{\boldsymbol{d}}_{j,\bar{t}}(t)$ is element-wisely non-negative, i.e., $\widehat{\boldsymbol{d}}_{j,\bar{t}}(t) \geq \boldsymbol{0}_{q_{j,2}}$, then the upper bound of the step size, $\lambda_j^{\max}$, is set to $+\infty$ accordingly. In this case, we can see that

$$
\begin{aligned}
\lambda_j \widehat{\boldsymbol{d}}_{j,\bar{t}}(t) &= \lambda_j \boldsymbol{E}_{j,2} \boldsymbol{d}_{j,\bar{t}}(t) \\
&= -\lambda_j \boldsymbol{E}_{j,2} \boldsymbol{P}_{j,\bar{t}}(t) \nabla_{\boldsymbol{p}(t)} g_j(\boldsymbol{p}_{j,\bar{t}}(t)) \geq \boldsymbol{0}_{q_{j,2}}
\end{aligned}
\tag{43}
$$

is held for $\lambda_j \geq 0$.

Thus, according to (41) and (43), we have

$$
\begin{bmatrix} -\lambda_j \boldsymbol{E}_{j,1} \boldsymbol{P}_{j,\bar{t}}(t) \nabla_{\boldsymbol{p}(t)} g_j(\boldsymbol{p}_{j,\bar{t}}(t)) \\ -\lambda_j \boldsymbol{E}_{j,2} \boldsymbol{P}_{j,\bar{t}}(t) \nabla_{\boldsymbol{p}(t)} g_j(\boldsymbol{p}_{j,\bar{t}}(t)) \end{bmatrix} \geq \begin{bmatrix} \boldsymbol{0}_{q_{j,1}} \\ \boldsymbol{0}_{q_{j,2}} \end{bmatrix} = \boldsymbol{0}_q
\tag{44}
$$

Note that the feasible point $\boldsymbol{p}_{j,\bar{t}}(t)$ satisfies $\boldsymbol{E} \boldsymbol{p}_{j,\bar{t}}(t) \geq \boldsymbol{0}_q$. Substituting (44) into (38) derives

$$
\boldsymbol{E} \boldsymbol{p}_{i,\bar{t}+1}(t) \geq \sum_{j=1}^{n} y_{j,i} \boldsymbol{E} \boldsymbol{p}_{j,\bar{t}}(t) \geq \boldsymbol{0}_q
\tag{45}
$$

At this point, the new iterator $\boldsymbol{p}_{j,\bar{t}+1}(t)$ also satisfies the inequality constraints of the optimization model (17).

2) If there exists at least one element in $\widehat{\boldsymbol{d}}_{j,\bar{t}}(t)$, i.e., $\widehat{\boldsymbol{d}}_{j,\bar{t}}(t) \not\geq \boldsymbol{0}_{q_{j,2}}$, we can divide $\widehat{\boldsymbol{d}}_{j,\bar{t}}(t)$ into two sub-blocks

$\widehat{\boldsymbol{d}}_{j,\bar{t},1}(t)$ and $\widehat{\boldsymbol{d}}_{j,\bar{t},2}(t)$ where the first sub-block $\widehat{\boldsymbol{d}}_{j,\bar{t},1}(t)$ contains the negative elements of $\widehat{\boldsymbol{d}}_{j,\bar{t}}(t)$ and the second $\widehat{\boldsymbol{d}}_{j,\bar{t},2}(t)$ contains the nonnegative elements, i.e., $\widehat{\boldsymbol{d}}_{j,\bar{t},1}(t) < \boldsymbol{0}_{r_{j,1}}$ and $\widehat{\boldsymbol{d}}_{j,\bar{t},2}(t) \geq \boldsymbol{0}_{r_{j,2}}$. $r_{j,1}$ and $r_{j,2}$ are the dimensions of these two sub-blocks, respectively, and satisfy $r_{j,1} + r_{j,2} = q_{j,2}$. Thus, $\widehat{\boldsymbol{d}}_{j,\bar{t}}(t)$ can be rearranged as $\widehat{\boldsymbol{d}}_{j,\bar{t}}(t) = \text{col}\{\widehat{\boldsymbol{d}}_{j,\bar{t},1}(t), \widehat{\boldsymbol{d}}_{j,\bar{t},2}(t)\}$. In addition, we also re-express the matrix $\boldsymbol{E}_{j,2}$ as $\boldsymbol{E}_{j,2} = \text{col}\{\boldsymbol{E}_{j,2,1}, \boldsymbol{E}_{j,2,2}\}$ where $\boldsymbol{E}_{j,2,1} \in \mathbb{R}^{r_{j,1} \times q}$ and $\boldsymbol{E}_{j,2,2} \in \mathbb{R}^{r_{j,2} \times q}$.

In the second case where the upper bound of the step size is limited by (36), it can be found that for any negative element $\widehat{d}_{j,\bar{t},1,l}(t) \in \widehat{\boldsymbol{d}}_{j,\bar{t},1}(t)$, the following inequality is held

$$\lambda_j \leq \lambda_j^{\max} = \min\left\{\frac{\widehat{b}_{j,\bar{t},l}(t)}{\widehat{d}_{j,\bar{t},1,l}(t)}\right\} \leq \frac{\widehat{b}_{j,\bar{t},l}(t)}{\widehat{d}_{j,\bar{t},1,l}(t)} \quad (46)$$

Additionally, noting $\widehat{d}_{j,\bar{t},1,l}(t) < 0$, (46) is equivalent to

$$\lambda_j \widehat{d}_{j,\bar{t},1,l}(t) \geq \widehat{b}_{j,\bar{t},l}(t) \quad (47)$$

We can further induce

$$\lambda_j \widehat{\boldsymbol{d}}_{j,\bar{t},1}(t) \geq \widehat{\boldsymbol{b}}_{j,\bar{t},1}(t) = \boldsymbol{0}_{r_{j,1}} - \boldsymbol{E}_{j,2,1}\boldsymbol{p}_{j,\bar{t}}(t) \quad (48)$$

where $\widehat{\boldsymbol{b}}_{j,\bar{t},1}(t)$ is a sub-block of $\widehat{\boldsymbol{b}}_{j,\bar{t}}(t)$ whose row indexes correspond to those of $\widehat{\boldsymbol{d}}_{j,\bar{t},1}(t)$. Since $\widehat{\boldsymbol{d}}_{j,\bar{t},2}(t)$ is element-wisely nonnegative, we can also get

$$\lambda_j \widehat{\boldsymbol{d}}_{j,\bar{t},2}(t) \geq \boldsymbol{0}_{r_{j,2}} \quad (49)$$

Combining (48) and (49), we can get

$$\lambda_j \widehat{\boldsymbol{d}}_{j,\bar{t}}(t) = \lambda_j \begin{bmatrix} \widehat{\boldsymbol{d}}_{j,\bar{t},1}(t) \\ \widehat{\boldsymbol{d}}_{j,\bar{t},2}(t) \end{bmatrix} \geq \begin{bmatrix} \boldsymbol{0}_{r_{j,1}} - \boldsymbol{E}_{j,2,1}\boldsymbol{p}_{j,\bar{t}}(t) \\ \boldsymbol{0}_{r_{j,2}} \end{bmatrix} \quad (50)$$

According to the definition of $\widehat{\boldsymbol{d}}_{j,\bar{t}}(t)$, i.e., $\widehat{\boldsymbol{d}}_{j,\bar{t}}(t) = \boldsymbol{E}_{j,2}\boldsymbol{d}_{j,\bar{t}}(t)$, (50) is equivalent to

$$\lambda_j \boldsymbol{E}_{j,2}\boldsymbol{d}_{j,\bar{t}}(t)$$
$$= -\lambda_j \boldsymbol{E}_{j,2}\boldsymbol{P}_{j,\bar{t}}(t)\nabla_{\boldsymbol{p}(t)}g_j(\boldsymbol{p}_{j,\bar{t}}(t)) \geq \begin{bmatrix} \boldsymbol{0}_{r_{j,1}} - \boldsymbol{E}_{j,2,1}\boldsymbol{p}_{j,\bar{t}}(t) \\ \boldsymbol{0}_{r_{j,2}} \end{bmatrix} \quad (51)$$

According to (41) and (51), the following inequality is held in this case:

$$\begin{bmatrix} -\lambda_j \boldsymbol{E}_{j,1}\boldsymbol{P}_{j,\bar{t}}(t)\nabla_{\boldsymbol{p}(t)}g_j(\boldsymbol{p}_{j,\bar{t}}(t)) \\ -\lambda_j \boldsymbol{E}_{j,2}\boldsymbol{P}_{j,\bar{t}}(t)\nabla_{\boldsymbol{p}(t)}g_j(\boldsymbol{p}_{j,\bar{t}}(t)) \end{bmatrix} \geq \begin{bmatrix} \boldsymbol{0}_{q_{j,1}} \\ \boldsymbol{0}_{r_{j,1}} - \boldsymbol{E}_{j,2,1}\boldsymbol{p}_{j,\bar{t}}(t) \\ \boldsymbol{0}_{r_{j,2}} \end{bmatrix} \quad (52)$$

Then, substituting (52) into (38), we further derive

$$\boldsymbol{E}\boldsymbol{p}_{i,\bar{t}+1}(t)$$
$$\geq \sum_{j=1}^{n} y_{j,i} \begin{bmatrix} \boldsymbol{E}_{j,1} \\ \boldsymbol{E}_{j,2,1} \\ \boldsymbol{E}_{j,2,2} \end{bmatrix} \boldsymbol{p}_{j,\bar{t}}(t)$$
$$+ \begin{bmatrix} \sum_{j=1}^{n} y_{j,i}\boldsymbol{0}_{q_{j,1}} \\ \sum_{j=1}^{n} y_{j,i}\left(\boldsymbol{0}_{r_{j,1}} - \boldsymbol{E}_{j,2,1}\boldsymbol{p}_{j,\bar{t}}(t)\right) \\ \sum_{j=1}^{n} y_{j,i}\boldsymbol{0}_{r_{j,2}} \end{bmatrix} \quad (53)$$
$$= \begin{bmatrix} \sum_{j=1}^{n} y_{j,i}\boldsymbol{E}_{j,1}\boldsymbol{p}_{j,\bar{t}}(t) \\ \sum_{j=1}^{n} y_{j,i}\left(\boldsymbol{E}_{j,2,1}\boldsymbol{p}_{j,\bar{t}}(t) + \boldsymbol{0}_{r_{j,1}} - \boldsymbol{E}_{j,2,1}\boldsymbol{p}_{j,\bar{t}}(t)\right) \\ \sum_{j=1}^{n} y_{j,i}\boldsymbol{E}_{j,2,2}\boldsymbol{p}_{j,\bar{t}}(t) \end{bmatrix}$$
$$\geq \boldsymbol{0}_q$$

Therefore, in the second case, $\boldsymbol{p}_{i,\bar{t}+1}(t)$ also satisfies the inequality constraints. To sum up, we have proven **Theorem 1**. ∎

Based on **Theorem 1**, we can enhance the adaption step in (35) by designing an appropriate step size λ_j. Specifically, we consider to optimize λ_j at each iteration $\bar{t}$ so as to minimize the objective function $g_j(\boldsymbol{p})$. Once the iteration direction $\boldsymbol{d}_{j,\bar{t}}(t)$ is obtained by $j \in \mathcal{V}_i$ at any iteration $\bar{t}$, we can treat the step size as a decision variable, denoted by λ, and formulate an objective function with respect to λ:

$$\phi_j(\lambda) = g_j\left(\boldsymbol{p}_{j,\bar{t}}(t) + \lambda\boldsymbol{d}_{j,\bar{t}}(t)\right) \quad (54)$$

Thus, an optimal searching step at $\bar{t}$, denoted by $\lambda_{j,\bar{t}}$, can be derived as

$$\lambda_{j,\bar{t}} = \underset{\lambda \in [0,\lambda_j^{\max}]}{\text{argmin}} \{\phi_j(\lambda)\} \quad (55)$$

where the upper bound $\lambda_j^{\max}$ is given according to **Theorem 1**.

D. Optimization of fusion weights

We represent $\boldsymbol{u}_{i,\bar{t}+1}(t) = \text{col}\{\boldsymbol{u}_{i,k,\bar{t}+1}(t)|k=1,\ldots,m\}$ where $\boldsymbol{u}_{i,k,\bar{t}+1}(t)$ is the intermediate estimation on the probability distribution of information states of the subregion k. From (35), it can be found that not only the step size λ_j but also those fusion weights $\{y_{j,i}|j \in \mathcal{V}_i\}$ have significant influence on the performance of (35). For simplicity, let $\boldsymbol{y}_i$ denote the set of fusion weights of the agent i, i.e., $\boldsymbol{y}_i = \{y_{j,i}|j \in \mathcal{V}_i\}$, and $\boldsymbol{U}_{i,k,\bar{t}+1}(t) = \{\boldsymbol{u}_{j,k,\bar{t}+1}(t)|j \in \mathcal{V}_i\}$. The Jensen-Shannon based cost function with respect to $\boldsymbol{y}_i$ can be defined by

$$JSD\left(\boldsymbol{U}_{i,k,\bar{t}+1}(t)\big|\boldsymbol{y}_i\right)$$
$$= H\left(\sum_{j \in \mathcal{V}_i} y_{j,i}\boldsymbol{u}_{j,k,\bar{t}+1}(t)\right)$$
$$- \left(\sum_{j \in \mathcal{V}_i} y_{j,i}H\left(\boldsymbol{u}_{j,k,\bar{t}+1}(t)\right)\right) \quad (56)$$

Subsequently, we can obtain optimal fusion weights by solving the following model

$$\min \ W_i(\boldsymbol{y}_i) = \sum_{k \in \Pi} JSD\left(\boldsymbol{U}_{i,k,\bar{t}+1}(t)\big|\boldsymbol{y}_i\right)$$
$$s.t. \begin{cases} \sum_{j \in \mathcal{V}_i} y_{j,i} = 1 \\ y_{j,i} \geq 0 \text{ for } \forall j \in \mathcal{V}_i \end{cases} \quad (57)$$

In (57), $\boldsymbol{y}_i$ represents the decision variable. This model takes $\boldsymbol{U}_{i,k,\bar{t}+1}(t)$ as input parameters, which implies that once $\{\boldsymbol{u}_{j,\bar{t}+1}(t)|j \in \mathcal{V}_i\}$ are aggregated at the agent i, the minimizer of this model can be solely solved by the individual agent. Thus, this model can be well integrated with the previous model (19). Besides, according to **Lemma 2**, the weight optimization model (57) also has a convex objective function. It is a typical convex optimization problem which can be solved by many existing efficient numerical algorithms such as
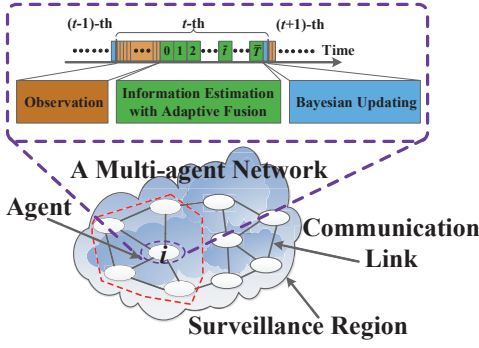
Fig. 1.   The deployment of a multi-agent network.

Fig. 2.   The framework of the adaptive fusion method.

augmented Lagrangian methods and the sequential quadratic programming techniques [29]–[34].

A schematic diagram of the deployment of a multi-agent network is given in Fig.1. The implementation of the algorithm presented by (35) mainly involves four steps: in the first step, any agent $i \in \mathcal{V}$ collects the gradient information from its immediate neighbors, $\{\nabla_{\boldsymbol{p}(t)} f_j(\boldsymbol{p}_{i,\bar{t}}(t)) | j \in \mathcal{V}_i\}$, so that it can construct a projection matrix $\boldsymbol{P}_{i,\bar{t}}(t)$ as well as the iteration direction $\boldsymbol{d}_{i,\bar{t}}(t)$; Then, the agent i can calculate an optimal searching step $\lambda_{i,\bar{t}}$ according to (55), and obtains its own intermediate estimation $\boldsymbol{u}_{i,\bar{t}+1}(t)$ based on the adaptation equation in (35); Thirdly, i obtains an optimal fusion weights $\boldsymbol{y}_i$ by solving the model (57) constructed based on its own and its neighbors' intermediate estimations; Finally, a next iterator $\boldsymbol{p}_{i,\bar{t}+1}(t)$ can be calculated based on the optimal fusion weights according to the combination equation in (35). These four steps are also performed at other agents, such that the overall network can achieve a cooperative information estimation.

To be specific, Fig.2 details the process of the overall proposed solution for cooperative information estimation over a multi-agent network, which combines the Bayesian updating formula (5) with (35), (55) and (57). Fig.2 shows that each agent i needs to broadcast its own intermediate estimation, $\boldsymbol{u}_{i,\bar{t}+1}(t)$, as well as the individual estimation, $\boldsymbol{p}_{i,\bar{t}}(t)$, to its immediate neighbors at each iteration $\bar{t}$. Especially, any agent's individual estimation $\boldsymbol{p}_{i,\bar{t}}(t)$ is used by each of its local neighbors to evaluate the gradient of the Jensen-Shannon based function (14), and once the gradient $\nabla_{\boldsymbol{p}(t)} f_j(\boldsymbol{p}_{i,\bar{t}}(t))$ is obtained by a neighbor j, it should be fed back from the neighbor j to the agent i.

There are two time indexes, i.e., t and $\bar{t}$. The former denotes each discrete time interval, during which any agent i carries out multiple measurements (i.e., collecting V_i samples on the detection signal) to obtain the real-time observation information on the local subregions, and then performs the iterative algorithm presented by (35), (55) and (57). At the end of each time interval t, the agent calculates the Bayesian updating formula (5) with the individual estimation on the global region obtained from the iterative algorithm. As shown in Fig.1, $\bar{t}$ is used to index the iteration of the proposed algorithm. At the beginning of the iterative procedure of any agent i at $t = 0$, i.e., at $\bar{t} = 0$ and $t = 0$, we initialize the individual estimation $\boldsymbol{p}_{i,\bar{t}}(t)$ by following the uniform distribution. That is, the
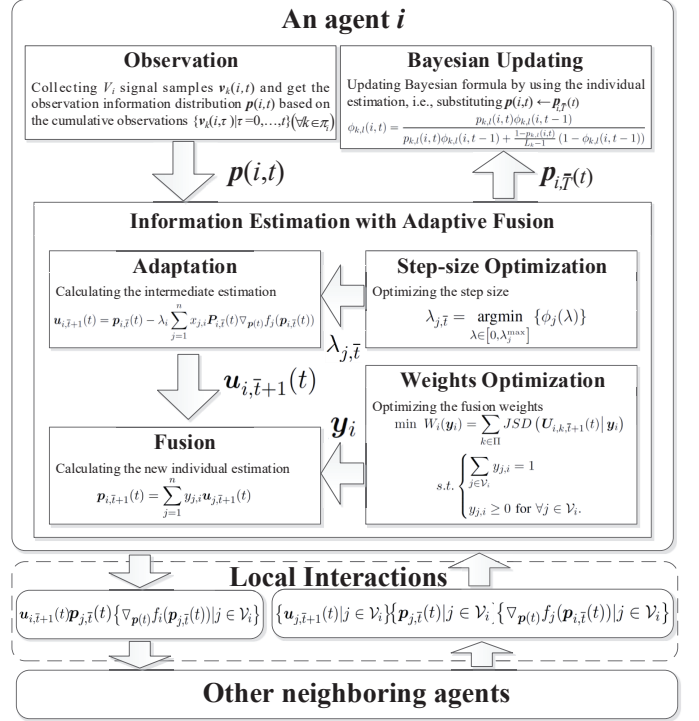
adaptive fusion algorithm for information estimation is initialized by setting $\{p_{i,k,l,0}(0) = 1/L_k | l = 1, \ldots, L_k; \forall k \in \Pi\}$ for all i, where $p_{i,k,l,0}(0)$ denotes the probability of the l-th information state of the subregion k estimated by the agent i at $\bar{t} = 0$. The individual prior distribution of information states of any subregion out of the detection range is also initialized by following the uniform distribution, i.e., $\{p_{k',l}(i,0) = 1/L_{k'} | i = 1, \ldots, L_{k'}; \forall k' \in \bar{\pi}_i\}$ for all i. In addition, for $t \geq 1$, at the beginning of the algorithm cycle, $\bar{t} = 0$, the individual estimation $\boldsymbol{p}_{i,0}(t)$ is initialized as $\boldsymbol{p}(i,t)$, i.e., $\boldsymbol{p}_{i,0}(t) \leftarrow \boldsymbol{p}(i,t)$. For the sake of practical computation, we pre-specify the total number of the algorithm iterations, denoted by $\overline{T}$, sufficiently large to guarantee the convergence of the algorithm. The detailed algorithm is shown in **Algorithm 1**.

The main steps of the adaptive fusion strategy are:

Step 1 **Observation**: During time interval t, each agent $i \in \mathcal{V}$ performs multiple measurements to collect V_i samples, $\{\boldsymbol{v}_k(i,t) | k \in \pi_i\}$, from the detection signals corresponding to the subregions in its detection range. Then, the agent can calculate the observation information distribution $\boldsymbol{p}_k(i,t)$ based on the cumulative observations $\{\boldsymbol{v}_k(i,\tau) | \tau = 0, \ldots, t;\ k \in \pi_i\}$. For the subregions out of the detection range, i.e., $\forall k' \in \bar{\pi}_i$, their prior distributions of information states are set to a zero vector. That is, let $\boldsymbol{p}_{k'}(i,t) \leftarrow \boldsymbol{0}$ for all $k' \in \bar{\pi}_i$.

Step 2 **Information estimation with adaptive fusion**: Initialize $\boldsymbol{p}_{i,\bar{t}}(t)$ for any i by setting $\boldsymbol{p}_{i,k,0}(t) = \boldsymbol{p}_k(i,t)$ for the subregions in i's detection range, i.e., for all $k \in \pi_i$, and setting $\boldsymbol{p}_{i,k',0}(t) = \boldsymbol{p}_{k'}(i,t-1)$ for those out of i's detection range, i.e., for all $k' \in \bar{\pi}_i$. Then,

---

Algorithm 1 DCIE-AF (Distributed Cooperative Information Estimation with Adaptive Fusion)

Input: $\boldsymbol{p}(i,t)$, $\boldsymbol{p}_{i,0}(t)$ and $\overline{T}$
Output: $\boldsymbol{p}_{i,\overline{T}}(t)$
  repeat
      Broadcast $\boldsymbol{p}_{i,\overline{t}}(t)$ to $j \in \mathcal{V}_i$ as well as collect $\{\boldsymbol{p}_{j,\overline{t}}(t)|j \in \mathcal{V}_i\}$.
      Evaluate $f_i(\boldsymbol{p}(t))$ at each $\boldsymbol{p}_{j,\overline{t}}(t)$ and broadcast $\{\nabla_{\boldsymbol{p}(t)}f_i(\boldsymbol{p}_{j,\overline{t}}(t))|j \in \mathcal{V}_i\}$.
      Collect $\{\nabla_{\boldsymbol{p}(t)}f_j(\boldsymbol{p}_{i,\overline{t}}(t))|j \in \mathcal{V}_i\}$.
      Construct $\boldsymbol{P}_{i,\overline{t}}(t)$ and $\boldsymbol{d}_{i,\overline{t}}(t)$ based on its own $\nabla_{\boldsymbol{p}(t)}f_i(\boldsymbol{p}_{i,\overline{t}}(t))$ and $\{\nabla_{\boldsymbol{p}(t)}f_j(\boldsymbol{p}_{i,\overline{t}}(t))|j \in \mathcal{V}_i\}$.
      Compute $\lambda_i^{\max}$ according to (36) and obtain an optimal step $\lambda_{i,\overline{t}}$ by solving (55).
      Obtain $\boldsymbol{u}_{i,\overline{t}+1}(t)$ based on $\boldsymbol{p}_{i,\overline{t}}(t)$, $\boldsymbol{d}_{i,\overline{t}}(t)$ and $\lambda_{i,\overline{t}}$ according to (35).
      Broadcast $\boldsymbol{u}_{i,\overline{t}+1}(t)$ to $j \in \mathcal{V}_i$ as well as collect $\{\boldsymbol{u}_{j,\overline{t}+1}(t)|j \in \mathcal{V}_i\}$.
      Optimize the fusion weights $\boldsymbol{y}_i$ with $\{\boldsymbol{u}_{j,\overline{t}+1}(t)|j \in \mathcal{V}_i\}$ by solving (57).
      Obtain new estimation $\boldsymbol{p}_{i,\overline{t}+1}(t)$ according to (35).
      $\overline{t} \leftarrow \overline{t}+1$.
  until $\overline{t} = \overline{T}$

---

      run the **Algorithm 1** until $\overline{t} = \overline{T}$.

Step 3 Bayesian Updating: Let $\boldsymbol{p}(i,t) \leftarrow \boldsymbol{p}_{i,\overline{T}}(t)$, do Bayesian updating based on (5) where $\boldsymbol{p}(i,t)$ is input parameter.

According to the iterative algorithm given above, we remark that since there may be some subregions that are located out of an agent's detection range, this agent's real-time observation distributions over these subregions can be simply set to zero, implying that this agent does not contribute to the information gain in estimation of these subregions. In fact, following the underlying idea of the model (35), the agents can achieve their individual estimations over the subregions out of their detection range by iteratively collecting and fusing the diffused estimations of others that can directly detect these subregions.

E. Analysis of Convergence

In a distributed sensor environment, each agent's computation is not only influenced by its own local observation data but also by the shared information of its local neighbors. To analyze the performance of the adaptive fusion strategy, we introduce an error vector:

$$\boldsymbol{e}_{i,\overline{t}}(t) \triangleq \boldsymbol{p}^*(t) - \boldsymbol{p}_{i,\overline{t}}(t) \tag{58}$$

Combining (24) and (34), we derive

$$\boldsymbol{p}_{i,\overline{t}+1}(t) = \sum_{j=1}^{n} y_{j,i}\boldsymbol{p}_{j,\overline{t}}(t) - \sum_{j=1}^{n} y_{j,i}\lambda_{j,\overline{t}}\boldsymbol{P}_{j,\overline{t}}(t)\nabla_{\boldsymbol{p}(t)}g_j(\boldsymbol{p}_{j,\overline{t}}(t)) \tag{59}$$

which can lead to (recalling $\sum_{j=1}^{n} y_{j,i} = 1$)

$$\boldsymbol{e}_{i,\overline{t}+1}(t) = \sum_{j=1}^{n} y_{j,i}\boldsymbol{e}_{j,\overline{t}}(t) + \sum_{j=1}^{n} y_{j,i}\lambda_{j,\overline{t}}\boldsymbol{P}_{j,\overline{t}}(t)\nabla_{\boldsymbol{p}(t)}g_j(\boldsymbol{p}_{j,\overline{t}}(t)) \tag{60}$$

Besides, the mean value theorem [35] shows that for any continuously differentiable function $R(x)$, the following integral equation is held:

$$R(x+h) - R(x) = \left(\int_0^1 R'(x+uh)\,\mathrm{d}u\right)h \tag{61}$$

where x denotes a certain variable, h is a real parameter, and u is an integral variable. In order to relate the gradient term $\nabla_{\boldsymbol{p}(t)}g_j(\boldsymbol{p}_{j,\overline{t}}(t))$ with the error quantity $\boldsymbol{e}_{j,\overline{t}}(t)$, we substitute $h = -\boldsymbol{e}_{j,\overline{t}}(t)$, $x = \boldsymbol{p}^*(t)$ and $R \triangleq \nabla_{\boldsymbol{p}(t)}g_j(\cdot)$ into (61) and rewrite $\nabla_{\boldsymbol{p}(t)}g_j(\boldsymbol{p}_{j,\overline{t}}(t))$ as

$$\begin{aligned} &\nabla_{\boldsymbol{p}(t)}g_j(\boldsymbol{p}_{j,\overline{t}}(t)) - \nabla_{\boldsymbol{p}(t)}g_j(\boldsymbol{p}^*(t)) \\ &= -\left(\int_0^1 \nabla^2_{\boldsymbol{p}(t)}g_j\left(\boldsymbol{p}^*(t) - u\boldsymbol{e}_{j,\overline{t}}(t)\right)\mathrm{d}u\right)\boldsymbol{e}_{j,\overline{t}}(t) \end{aligned} \tag{62}$$

Since $\boldsymbol{p}^*(t)$ represents an optimum of the objective function $g_j(\boldsymbol{p}(t))$, we can have $\nabla_{\boldsymbol{p}(t)}g_j(\boldsymbol{p}^*(t)) = \boldsymbol{0}$. Thus, (62) is reduced to

$$\nabla_{\boldsymbol{p}(t)}g_j(\boldsymbol{p}_{j,\overline{t}}(t)) = -\left(\int_0^1 \nabla^2_{\boldsymbol{p}(t)}g_j\left(\boldsymbol{p}^*(t) - u\boldsymbol{e}_{j,\overline{t}}(t)\right)\mathrm{d}u\right)\boldsymbol{e}_{j,\overline{t}}(t) \tag{63}$$

Substituting (63) into (60) yields

$$\boldsymbol{e}_{j,\overline{t}+1}(t) = \sum_{j=1}^{n} y_{j,i}\left(\boldsymbol{E} - \lambda_{j,\overline{t}}\boldsymbol{P}_{j,\overline{t}}(t)\boldsymbol{H}_{j,\overline{t}}(t)\right)\boldsymbol{e}_{j,\overline{t}}(t) \tag{64}$$

where $\boldsymbol{H}_{j,\overline{t}}(t)$ is defined by

$$\boldsymbol{H}_{j,\overline{t}}(t) \triangleq \int_0^1 \nabla^2_{\boldsymbol{p}(t)}g_j\left(\boldsymbol{p}^*(t) - u\boldsymbol{e}_{j,\overline{t}}(t)\right)\mathrm{d}u \tag{65}$$

Lemma 4 (Positive semi-definiteness of gradient projection matrix): Given that $\boldsymbol{P}_{j,\overline{t}}(t)$ obtained from (22) satisfies

$\boldsymbol{P}_{j,\bar{t}}(t)\nabla^2_{\boldsymbol{p}(t)}g_j\left(\boldsymbol{p}_{j,\bar{t}}(t)\right) \neq \boldsymbol{0}$, $\boldsymbol{P}_{j,\bar{t}}(t)$ is a non-zero symmetrical positive-semidefinite matrix and its eigenvalues only consists of 1 and 0.

Proof: According to (22), it is easy to validate $\left(\boldsymbol{P}_{j,\bar{t}}(t)\right)^{\mathrm{T}} = \boldsymbol{P}_{j,\bar{t}}(t)$ and $\left(\boldsymbol{P}_{j,\bar{t}}(t)\right)^2 = \boldsymbol{P}_{j,\bar{t}}(t)$. Thus, for any non-zero column vector $\boldsymbol{x} \neq \boldsymbol{0}$, we can get

$$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{P}_{j,\bar{t}}(t)\boldsymbol{x} = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{P}_{j,\bar{t}}(t)^{\mathrm{T}}\boldsymbol{P}_{j,\bar{t}}(t)\boldsymbol{x} = \left(\boldsymbol{P}_{j,\bar{t}}(t)\boldsymbol{x}\right)^{\mathrm{T}}\left(\boldsymbol{P}_{j,\bar{t}}(t)\boldsymbol{x}\right) \geq 0 \tag{66}$$

Thus, $\boldsymbol{P}_{j,\bar{t}}(t)$ is a positive semidefinite matrix.

Furthermore, since $\boldsymbol{P}_{j,\bar{t}}(t) - \boldsymbol{P}_{j,\bar{t}}(t)^2 = \boldsymbol{0}$, the characteristic polynomial equation of $\boldsymbol{P}_{j,\bar{t}}(t)$ can be expressed as $x - x^2 = (0 - x)(1 - x) = 0$. Therefore, the eigenvalue set of $\boldsymbol{P}_{j,\bar{t}}(t)$ is composed of 0 and 1. Additionally, it should be noted that since $\boldsymbol{P}_{j,\bar{t}}(t) \neq \boldsymbol{0}$ (otherwise, a contradiction would arise from $\boldsymbol{P}_{j,\bar{t}}(t) = \boldsymbol{0}$ when considering $\boldsymbol{P}_{j,\bar{t}}(t)\nabla_{\boldsymbol{p}(t)}g_j(\boldsymbol{p}_{j,\bar{t}}(t)) \neq \boldsymbol{0}$), all of the eigenvalues of $\boldsymbol{P}_{j,\bar{t}}(t)$ cannot be equal to 0 simultaneously. Thus, **Lemma 4** is proved. ∎

On the other hand, according to **Lemma 2**, the Jensen-Shannon based function $f_i(\boldsymbol{p}(t))$ is strongly convex. This implies that its Hessian matrix $\nabla^2_{\boldsymbol{p}(t)}f_i(\boldsymbol{p}(t))$ can be sufficiently bounded away from 0. To show this, we first derive the expression of $\nabla^2_{\boldsymbol{p}(t)}f_i(\boldsymbol{p}(t))$ as

$$\nabla^2_{\boldsymbol{p}(t)}f_i(\boldsymbol{p}(t)) = \mathrm{diag}\left\{\nabla^2_{\boldsymbol{p}_k(t)}JSD_k(\boldsymbol{p}_k(t),\boldsymbol{p}_k(i,t))|k \in \Pi\right\} \tag{67}$$

Recalling the definition of $JSD_k(\boldsymbol{p}_k(t),\boldsymbol{p}_k(i,t))$, we further derive

$$\nabla^2_{\boldsymbol{p}_k(t)}JSD_k(\boldsymbol{p}_k(t),\boldsymbol{p}_k(i,t))$$
$$= \nabla^2_{\boldsymbol{p}_k(t)}H(\omega_{k,1}\boldsymbol{p}_k(t) + \omega_{k,2}\boldsymbol{p}_k(i,t)) - \omega_{k,1}\nabla^2_{\boldsymbol{p}_k(t)}H(\boldsymbol{p}_k(t)) \tag{68}$$

Accordingly, $\nabla^2_{\boldsymbol{p}_k(t)}JSD_k(\boldsymbol{p}_k(t),\boldsymbol{p}_k(n,t))$ can be expressed as

$$\nabla^2_{\boldsymbol{p}_k(t)}JSD_k(\boldsymbol{p}_k(t),\boldsymbol{p}_k(i,t))$$
$$= \mathrm{diag}\left\{\frac{\omega_{k,1}\omega_{k,2}\left(p_{k,l}(i,t) + \epsilon\right)}{(\ln 2)\left(p_{k,l}(t) + \epsilon\right)\left(\omega_{k,1}p_{k,l}(t) + \omega_{k,2}p_{k,l}(i,t) + \epsilon\right)}\Big|\right\} \tag{69}$$

where $l = 1,\ldots,L_k$. From (67) and (69), it can be found that $\nabla^2_{\boldsymbol{p}(t)}f_i(\boldsymbol{p}(t))$ is indeed a diagonal matrix. Thus, we can easily get the spectral radius of $\nabla^2_{\boldsymbol{p}(t)}f_i(\boldsymbol{p}(t))$

$$\rho\left(\nabla^2_{\boldsymbol{p}(t)}f_i(\boldsymbol{p}(t))\right)$$
$$= \max_{\forall k,l}\left\{\frac{\omega_{k,1}\omega_{k,2}\left(p_{k,l}(i,t) + \epsilon\right)}{(\ln 2)\left(p_{k,l}(t) + \epsilon\right)\left(\omega_{k,1}p_{k,l}(t) + \omega_{k,2}p_{k,l}(i,t) + \epsilon\right)}\right\} \tag{70}$$

Since the agent j''s Hessian matrix evaluated at the agent j's individual estimation $\boldsymbol{p}_{j,\bar{t}}(t)$ can be represented by $\nabla^2_{\boldsymbol{p}(t)}f_{j'}(\boldsymbol{p}_{j,\bar{t}}(t))$, we can calculate the upper bound of the spectral radius of $\nabla^2_{\boldsymbol{p}(t)}f_{j'}(\boldsymbol{p}_{j,\bar{t}}(t))$

$$\rho\left(\nabla^2_{\boldsymbol{p}(t)}f_{j'}(\boldsymbol{p}_{j,\bar{t}}(t))\right)$$
$$= \max_{\forall k,l}\left\{\frac{\omega_{k,1}\omega_{k,2}\left(p_{k,l}(j',t) + \epsilon\right)}{(\ln 2)\left(p_{j,k,l,\bar{t}}(t) + \epsilon\right)\left(\omega_{k,1}p_{j,k,l,\bar{t}}(t) + \omega_{k,2}p_{k,l}(j',t) + \epsilon\right)}\right\}$$
$$\leq \max_{\forall k,l}\left\{\frac{\omega_{k,1}\omega_{k,2}\left(p_{k,l}(j',t) + \epsilon\right)}{\epsilon(\ln 2)\left(\omega_{k,2}p_{k,l}(j',t) + \epsilon\right)}\right\} = \beta_{j',j} \tag{71}$$

where $p_{j,k,l,\bar{t}}(t)$ corresponds to the probability of the l-th information state of the subregion k estimated by the agent j, and $\beta_{j',j}$ represents the upper bound. The **Theorem 2** is presented for stable-state performance.

Theorem 2 (The stable-state convergence): Given $0 < \lambda_j < \min\left\{\lambda_j^{\max}, \frac{2}{\sum_{j'=1}^n x_{j',j}\beta_{j',j}}\right\}$ for any $j \in \mathcal{V}$, the iterative procedure given in (35) asymptotically converges to a stable state for any initial feasible solutions $\{\boldsymbol{p}_{j,0}(t)|j \in \mathcal{V}\}$, and as $\bar{t} \to +\infty$,

$$\lim_{\bar{t}\to+\infty}\|\boldsymbol{e}_{j,\bar{t}}(t)\|_2 = 0 \tag{72}$$

is held for all $j \in \mathcal{V}$.

Proof: According to **Lemma 4**, the spectral radius of $\boldsymbol{P}_{j,\bar{t}}(t)$ cannot be larger than 1, i.e., $\rho(\boldsymbol{P}_{j,\bar{t}}(t)) \leq 1$. This indicates that $\rho\left(\lambda_j\boldsymbol{P}_{j,\bar{t}}(t)\boldsymbol{H}_{j,\bar{t}}(t)\right) \leq \lambda_j\rho\left(\boldsymbol{P}_{j,\bar{t}}(t)\right)\rho\left(\boldsymbol{H}_{j,\bar{t}}(t)\right) \leq \lambda_j\rho\left(\boldsymbol{H}_{j,\bar{t}}(t)\right)$. From $\boldsymbol{H}_{j,\bar{t}}(t)$ in (65), we can further get

$$\lambda_j\rho\left(\boldsymbol{H}_{j,\bar{t}}(t)\right) = \lambda_j\rho\left(\sum_{j'=1}^n x_{j',j}\int_0^1 \nabla^2_{\boldsymbol{p}(t)}f_{j'}\left(\boldsymbol{p}^*(t) - u\boldsymbol{e}_{j,\bar{t}}(t)\right)\mathrm{d}u\right) \tag{73}$$

Based on the mean value theorem and (71), there always exists a certain point $\boldsymbol{\xi}_{j,\bar{t}}(t) \in \left[\boldsymbol{p}^*(t) - \boldsymbol{e}_{j,\bar{t}}(t), \boldsymbol{p}^*(t)\right]$ that

$$\int_0^1 \nabla^2_{\boldsymbol{p}(t)}f_{j'}\left(\boldsymbol{p}^*(t) - u\boldsymbol{e}_{j,\bar{t}}(t)\right)\mathrm{d}u$$
$$= \nabla^2_{\boldsymbol{p}(t)}f_{j'}\left(\boldsymbol{\xi}_{j,\bar{t}}(t)\right)\left(\int_0^1 \mathrm{d}u\right)$$
$$\leq \rho\left(\nabla^2_{\boldsymbol{p}(t)}f_{j'}\left(\boldsymbol{\xi}_{j,\bar{t}}(t)\right)\right)\boldsymbol{E}$$
$$< \beta_{j',j}\boldsymbol{E} \tag{74}$$

Substituting (74) into (73), we can get

$$\lambda_j\rho\left(\boldsymbol{H}_{j,\bar{t}}(t)\right) < \lambda_j\sum_{j'=1}^n x_{j',j}\beta_{j',j} \tag{75}$$

Since $\lambda_j < \frac{2}{\sum_{j'=1}^n x_{j',j}\beta_{j',j}}$, we can get $\rho\left(\lambda_j\boldsymbol{P}_{j,\bar{t}}(t)\boldsymbol{H}_{j,\bar{t}}(t)\right) < \lambda_j\sum_{j'=1}^n x_{j',j}\beta_{j',j} \leq 2$. This result is equivalent to

$$\left|1 - \rho\left(\lambda_j\boldsymbol{P}_{j,\bar{t}}(t)\boldsymbol{H}_{j,\bar{t}}(t)\right)\right| < 1 \tag{76}$$

which implies that the matrix $\left(\boldsymbol{E} - \lambda_j\boldsymbol{P}_{j,\bar{t}}(t)\boldsymbol{H}_{j,\bar{t}}(t)\right)$ is stable, namely, its sub-multiplicative matrix norm $\|\boldsymbol{E} - \lambda_j\boldsymbol{P}_{j,\bar{t}}(t)\boldsymbol{H}_{j,\bar{t}}(t)\|_2$ satisfying

$$\|\boldsymbol{E} - \lambda_j\boldsymbol{P}_{j,\bar{t}}(t)\boldsymbol{H}_{j,\bar{t}}(t)\|_2 \leq \rho_{max} + \delta < 1 \tag{77}$$

where $\rho_{max} = \max_{j \in \mathcal{V}} \left\{ \rho \left(\boldsymbol{E} - \lambda_j \boldsymbol{P}_{j,\bar{t}}(t) \boldsymbol{H}_{j,\bar{t}}(t) \right) \right\}$, and δ is a positive number that is sufficiently small.

Now, we denote the global error vector by $\boldsymbol{Q}_{\bar{t}}(t)$ that collects the error quantities from all the agents

$$\boldsymbol{Q}_{\bar{t}}(t) \triangleq \text{col} \left\{ \|e_{j,\bar{t}}(t)\|_2 \,\middle|\, j \in \mathcal{V} \right\} \qquad (78)$$

Also, we denote a fusion weight matrix as $\boldsymbol{Y}' \triangleq [y_{j,i}]_{n \times n}$ and lump all the step sizes into a diagonal matrix $\boldsymbol{\Lambda}' \triangleq \text{diag}\{\lambda_j | j \in \mathcal{V}\}$. Subsequently, we derive two block matrices by using the Kronecker product operator $\otimes$:

$$\boldsymbol{Y} = \boldsymbol{Y}' \otimes \boldsymbol{E} \qquad (79)$$

$$\boldsymbol{\Lambda} = \boldsymbol{\Lambda}' \otimes \boldsymbol{E} \qquad (80)$$

where $\boldsymbol{Y}$ and $\boldsymbol{\Lambda}$ are of nq columns and nq rows. Using these notations above, we can rewrite (64) as

$$\boldsymbol{Q}_{\bar{t}+1}(t) = \boldsymbol{Y}^{\mathrm{T}} \left(\boldsymbol{I}_{nq} - \boldsymbol{\Lambda} \boldsymbol{Z}_{\bar{t}}(t) \right) \boldsymbol{Q}_{\bar{t}}(t) \qquad (81)$$

where $\boldsymbol{Z}_{\bar{t}}(t) = \text{diag}\left\{ \boldsymbol{P}_{j,\bar{t}}(t) \boldsymbol{H}_{j,\bar{t}}(t) | j \in \mathcal{V} \right\}$.

Since $\|\boldsymbol{Y}^T\|_2 = 1$, we can calculate the 2-norm of both sides of (81) as

$$\begin{aligned} \|\boldsymbol{Q}_{\bar{t}+1}(t)\|_2 &= \|\boldsymbol{Y}^{\mathrm{T}} \left(\boldsymbol{I}_{nq} - \boldsymbol{\Lambda} \boldsymbol{Z}_{\bar{t}}(t) \right) \boldsymbol{Q}_{\bar{t}}(t)\|_2 \\ &\leq \|\boldsymbol{Y}^{\mathrm{T}}\|_2 \cdot \|\boldsymbol{I}_{nq} - \boldsymbol{\Lambda} \boldsymbol{Z}_{\bar{t}}(t)\|_2 \cdot \|\boldsymbol{Q}_{\bar{t}}(t)\|_2 \\ &\leq (\rho_{max} + \delta) \|\boldsymbol{Q}_{\bar{t}}(t)\|_2 \\ &= (\rho_{max} + \delta)^{\bar{t}+1} \|\boldsymbol{Q}_0(t)\|_2 \end{aligned} \qquad (82)$$

Since $(\rho_{max} + \delta) < 1$, we can get $\lim_{\bar{t} \to +\infty} \|\boldsymbol{Q}_{\bar{t}+1}(t)\|_2 = 0$. The result is equivalent to (72). ∎

V. Numerical Examples

To evaluate the performance of the distributed information estimation method, we set up a scenario where the entire surveillance region is divided into $16 = 4 \times 4$ subregions and several agents are initially generated and distributed over this surveillance region. Each agent is considered to be able to detect partial subregions, and multiple agents are assumed to be locally connected when they are within each other's communication range. An instance of a network's communication topology with 6 agents is shown in Fig.3, where every grid denotes a subregion, and the whole grid plane of 2 dimensions represents the overall surveillance region.

We assume that each subregion is an information source which could generate a series of information of interest. And the information values follow some certain distributions. For example, the k-th subregion generates the information $v_k(t)$ at time t, and this subregion's actual information is assumed to be a random variable following a finite discrete distribution that could be represented with different states $\{S_{k,l} | l = 1, \ldots, L_k\}$ (L_k is the number of the whole states.). Each state denotes a certain range of the information variable $v_k(t)$ and is associated with a certain probability $p(v_k(t) \in S_{k,l})$. For testing our model, we randomly generate the distribution of information states of each subregion, as shown by the dark histogram in Fig.4. These distributions are treated as the actual information distributions which each agent would like to estimate. In

measurement interval, each agent could observe $sampleNum$ samples from each subregion in its detection range and could not detect those subregions out of its range. For instance, in Fig.3, the grids at the right side of the entire plane could not be detected by the agent located at the top left of the plane. We also assume that there are some certain white Gaussian noises existing in each agent's observations. The mean value of the noise corresponding to the agent i is set to 0, and the standard deviation σ_i. For simulations, we randomly generate σ_i for each agent by following a uniform distribution $U[0, \sigma_{\max}]$, namely, $\sigma_i \sim U[0, \sigma_{\max}]$. Additionally, we denote the total number of time intervals as T, i.e., $0 \leq t \leq T$, and the number of epochs for performing the proposed iterative scheme (**Algorithm 1**) as $\overline{T}$, i.e., $0 \leq \bar{t} \leq \overline{T}$. We firstly fix $\sigma_{\max} = 1.5$, $sampleNum = 100$, $T = 1000$ and $\overline{T} = 50$ respectively. In the Shannon entropy based function (12), the small parameter ϵ is set to 1×10^{-6}. The probability distribution corresponding to the information state of every subregion maintained by each individual agent is initialized by following the uniform distribution. Then, we set the agent number $n = 6$ and have performed the simulations with 50 independent replications. Thus, we could average the results over those 50 simulations and then show the whole estimation results. we randomly select an agent (whose detection subregions are $\{2, 3, 4, 6, 7, 8\}$) and illustrate the convergence of its individual estimation corresponding to the whole subregions in Fig.4 (the gray histogram in Fig.4). It can be found that even though those subregions (the subregions $\{1, 5, 9, 10, 11, 12, 13, 14, 15, 16\}$) are out of the detection range of this selected agent, its actual information state distribution could also be well approached by this agent's estimation based on our proposed method.

In order to quantify the performance, we define the evaluation metric, i.e., the averaged absolute error at t obtained by an agent i, which is denoted by $AAE(i,t)$ and calculated as

$$AAE(i,t) = \frac{\sum_{k=1}^{m} \sum_{l=1}^{L_k} \left| p_{k,l}(i,t) - p_{k,l,\overline{T}}(i,t) \right|}{q} \qquad (83)$$

Similarly, to evaluate the whole agent network, we could also calculate the absolute error by averaging over all the agents' results. We define the averaged absolute error relevant to the
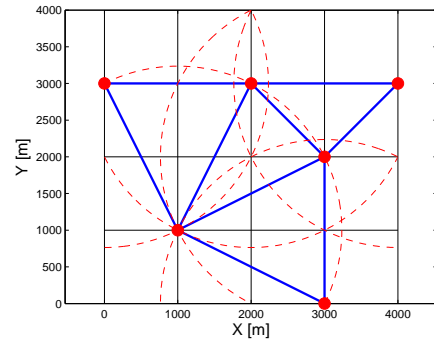

Fig. 3. A region and a network's communication topology with 6 agents. (Red dots: agents. Red dotted lines: agents' detection ranges. Blue lines: the network's communication topology).
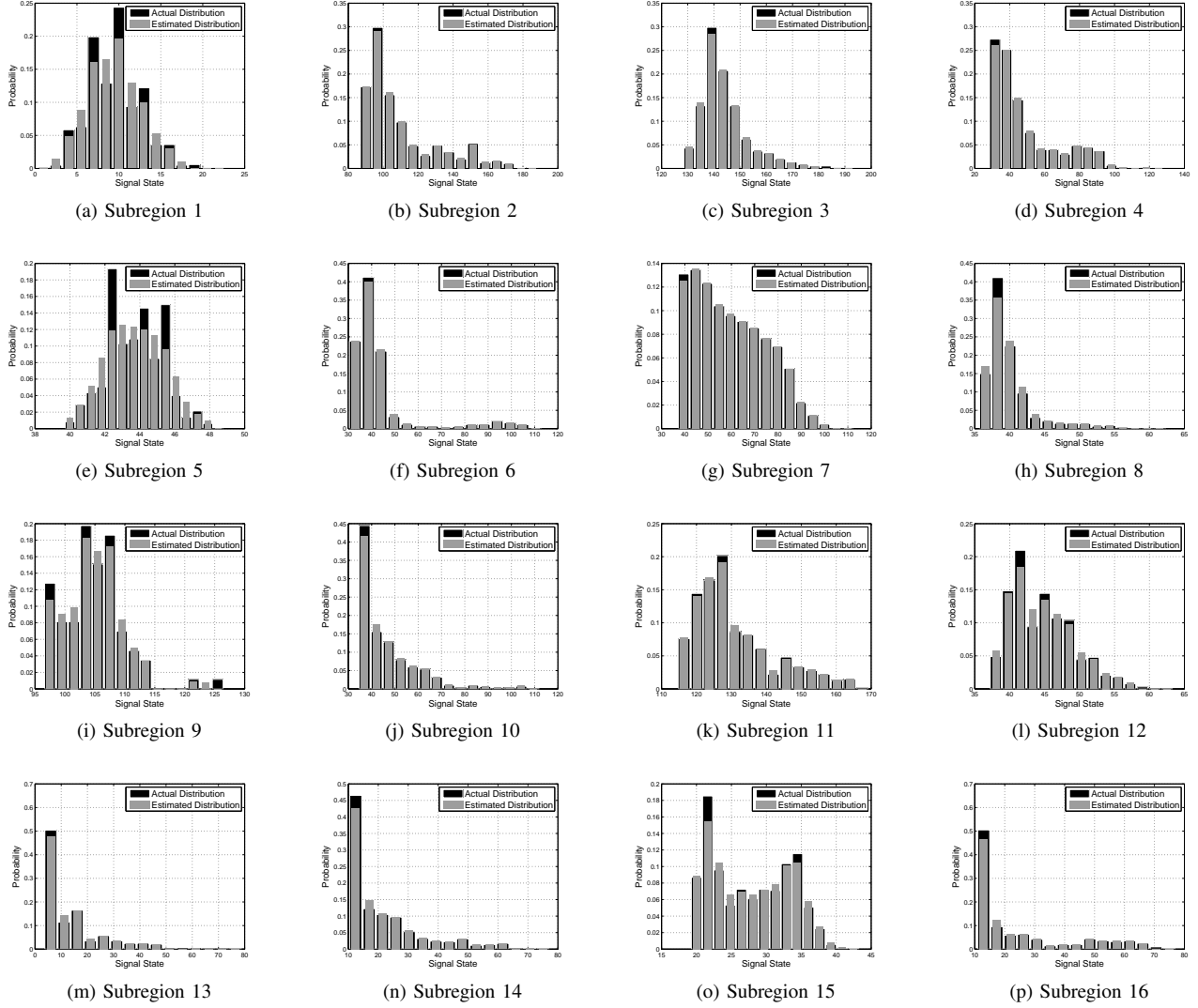
Fig. 4. Simulated distributions of information states of 16 subregions.

agent network as $AAE(t)$

$$AAE(t) = \frac{\sum_{i=1}^{n} AAE(i,t)}{n} \qquad (84)$$

Next, for evaluating the influence of different parameters on the proposed scheme, we set up several simulation situations. In the first situation, we use $\sigma_{\max} = 1.5$, $sampleNum = 100$, $T = 1000$ and $\overline{T} = 50$, and vary the number of agents from 6 to 14, namely, $n \in \{6, 8, 10, 12, 14\}$. In this situation, we also compare our method with the distributed information estimation strategy based on the traditional consensus approach [7], which adopts the same settings on $\sigma_{\max}$, $sampleNum$, T and $\overline{T}$. The compared method is simulated at $n = 6$. The results at each point n are also averaged over 50 independent simulations. These results are shown in Fig.5. In the second situation, we fix the number of agents as $n = 6$ and then set different $\sigma_{\max}$ to show the impact of the noise magnitude on the convergence. The consensus-based method uses the fixed setting $\sigma_{\max} = 0.1$ in this situation. The third situation simulates the proposed method at different settings on the sample number, $sampleNum \in \{10, 50, 200, 300\}$, while we

fix $sampleNum = 300$ for the compared method. In this situation we also keep $\sigma_{\max} = 1.5$, $T = 1000$, $\overline{T} = 50$ and $n = 6$.

From the results, it can be found that the agent number impacts the estimation performance of the overall network. Particularly, with the same number of agents (e.g., $n = 6$), our proposed method is shown to outperform the compared one. Besides, from Fig.5a, we find that more agents does not always guarantee a better estimation. This figure shows that the averaged absolute error of the individual estimation obtained by a network consisting of 6 agents is slightly better than those obtained by the other networks which contain more than 6 agents. The main reason for a slight drop in the performance is that more agents, indicating a larger-scale network (involving more decision variables in the optimization model), could slow down the convergence of the overall network. In order to ensure that the distributed algorithm could approach closer to an optimal point of a larger-scale optimization problem, they need more interactions and more numbers of learning (iterations) to diffuse and fuse local information over the

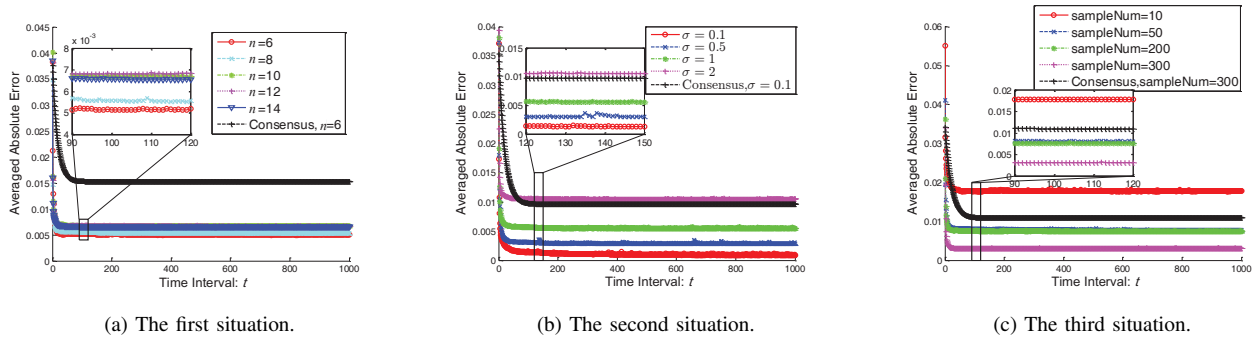(a) The first situation.   (b) The second situation.   (c) The third situation.

Fig. 5.   The results obtained from different situations.

network. Recall that we have set the same total number of iterations between successive observations, i.e., $\overline{T} = 50$, for all the networks. That is, the iterative algorithm dealing with a larger-scale optimization problem in the case of a larger-scale network is terminated in the same total step size as in the case of a smaller-scale network, which could result in a relatively conservative estimate for the optimal point. By contrast, the optimization algorithm could approach much closer to the optimal point within a finite number of iterations when it handles a smaller-scale optimization problem. Actually, it is clear that there is a trade-off between the choice of the amount of the agents deployed (i.e., the scale of the multi-agent network) and the potential computing cost arising from complexity in the distributed optimization. On one hand, a minimum number of agents should be needed so as to ensure that their overall detection range could cover the whole surveillance area. More agents added could improve the reliability of the distributed network. On the other hand, the network size is also increased by more agents, leading to more complexity of the network topology, which would require much more effort to handle the corresponding distributed optimization problem. Generally, a distributed algorithm needs more computing/communication resources (e.g., data buffer, bandwidth) to address a larger-scale model and more iterations to guarantee its convergence. Thus, there should be an optimal number of agents for deployment, which could depend on not only the algorithm cost and performance in distributed computation, but also the amount and distribution of surveillance regions, noise level in observations and other comprehensive factors involved in a specified application context. The issue is related to the optimal coverage control, which aims at a solution that could attain a specified optimal objective, at the meanwhile determining a minimum number of agents whose coverage could cover the target area. This issue is out of scope of this paper, it can be referred to many existing works such as [36]–[38].

In the second situation, see Fig.5b, the increased noise level indicated by $\sigma_{\max}$ could result in a larger averaged absolute error of the overall network. The noise in individual observations could also lead to the randomness of the gradient process in the equation (62), so that the actual estimation error of the network could not fully converge to zeros. This fact could also be validated by the results in the first situation

as shown in Fig.5a. That is, the larger amount of agents does not guarantee a lower gradient noise process of the overall network. Nevertheless, from the results in Fig.5b, it can be found that once the noises of individual observations under a certain level, a good global information estimation of the overall network could be guaranteed regardless of the limitation of each individual agent's detection and communication. The results from the third situation in Fig.5c shows that increasing the amount of measurements (denoted by the sample number $sampleNum$) at each individual observation could decrease the absolute error on average by the proposed method. It is because a larger number of samples observed by each individual agent could make the individual estimation more approximate to the actual distribution of a subregion. The result is in accordance with the law of large number [39]. Additionally, comparing the results obtained by our proposed method with those of the consensus-based method, the averaged absolute error converges to a lower level by our method. Specifically, even when our method is simulated at a higher noise level such as $\sigma_{\max} = 0.5, 1$ (see Fig.5b) or adopts a smaller sample number such as $sampleNum = 50, 200$ (see Fig.5c), it outperforms the consensus-based method as well.

Finally, to evaluate the influence of different observation amounts and iteration numbers on the performance of the multi-agent network, we simulate our method at different $T \in \{100, 400, 700\}$ and $\overline{T} \in \{50, 80, 100\}$. In addition, in this situation we fix $sampleNum = 100$, and the other parameters $\sigma_{\max}$ and n are set to be $\sigma_{\max} = 1.5$ and $n = 6$. From the results shown in Fig.6, it can be found that more observations or more iterations tend to reduce the averaged absolute error of the whole network. The main reason is that more observations, implying more measurements, could make each agent approach to the actual information state distributions of those subregions in its detection coverage, and increasing iteration number could make individual estimation much closer to the ideal solution of the optimization model (19). Appropriate settings on them could be selected or tuned based on a specified application scenario of interest.

VI. CONCLUSION

In this paper, we have proposed an adaptive distributed fusion strategy for estimating global information over a cooperative multi-agent network. We have modeled the information
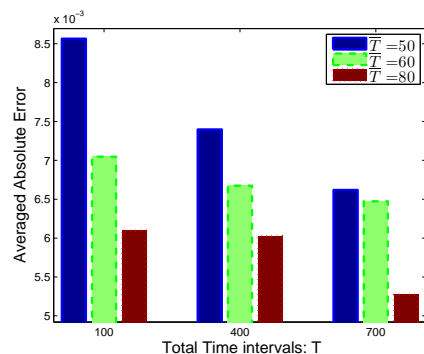
Fig. 6. Evaluation of the proposed method at different T and $\overline{T}$.

of interest associated with a global region being detected by the network from a probabilistic perspective, in which probability is related to the accuracy of observed and estimated information. We apply an information-theoretic measure, i.e., the Jensen-Shannon divergence, to formulate two objective functions of any agent subject to linear equality and inequality constraints, one of which is used in a localized estimation optimization model and the other in a fusion weight optimization model. The adaptive fusion strategy allows each agent to achieve its own optimal individual estimation on the global information of the entire region through minimizing the localized objective functions in a distributed and online manner. We have also analyzed the mean-square-error convergence behavior of the proposed algorithm. Finally, the experimental results have been provided to demonstrate the effectiveness of the proposed algorithm and its advantage over the conventional consensus protocol based algorithm. In our future research, we will extend the proposed method with consideration of optimal coverage control issue and potential failure of local communication links.

REFERENCES

[1] L. Bertuccelli and J. How, "Uav search for dynamic targets with uncertain motion models," in *Decision and Control, 2006 45th IEEE Conference on*, Dec 2006, pp. 5941–5946.

[2] P. T. Millet, D. W. Casbeer, T. Mercker, and J. Bishop, "Multi-agent decentralized search of a probability map with communication constraints," in *AIAA Guidance, Navigation and Control Conference*, 2010.

[3] J. Hu, L. Xie, K.-Y. Lum, and J. Xu, "Multiagent information fusion and cooperative control in target search," *Control Systems Technology, IEEE Transactions on*, vol. 21, no. 4, pp. 1223–1235, July 2013.

[4] D. P. Bertsekas, "A new class of incremental gradient methods for least squares problems," *SIAM J. Optim*, vol. 7, pp. 913–926, 1996.

[5] M. Rabbat and R. Nowak, "Quantized incremental algorithms for distributed optimization," *Selected Areas in Communications, IEEE Journal on*, vol. 23, no. 4, pp. 798–808, April 2005.

[6] C. Lopes and A. Sayed, "Incremental adaptive strategies over distributed networks," *Signal Processing, IEEE Transactions on*, vol. 55, no. 8, pp. 4064–4077, Aug 2007.

[7] X. Lin, "A scheme for robust distributed sensor fusion based on average consensus," in *Proceedings of the International Conference on Information Processing in Sensor Networks*, 2005, pp. 63–70.

[8] I. Schizas, G. Mateos, and G. Giannakis, "Distributed lms for consensus-based in-network adaptive processing," *Signal Processing, IEEE Transactions on*, vol. 57, no. 6, pp. 2365–2382, June 2009.

[9] S. Stankovic, M. Stankovic, and D. Stipanovic, "Decentralized parameter estimation by consensus based stochastic approximation," *Automatic Control, IEEE Transactions on*, vol. 56, no. 3, pp. 531–543, March 2011.

[10] C. Lopes and A. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *Signal Processing, IEEE Transactions on*, vol. 56, no. 7, pp. 3122–3136, July 2008.

[11] F. Cattivelli and A. Sayed, "Diffusion lms strategies for distributed estimation," *Signal Processing, IEEE Transactions on*, vol. 58, no. 3, pp. 1035–1048, March 2010.

[12] J. Chen and A. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *Signal Processing, IEEE Transactions on*, vol. 60, no. 8, pp. 4289–4305, Aug 2012.

[13] A. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. Towfic, "Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior," *Signal Processing Magazine, IEEE*, vol. 30, no. 3, pp. 155–171, May 2013.

[14] J. Chen, C. Richard, A. Hero, and A. Sayed, "Diffusion lms for multitask problems with overlapping hypothesis subspaces," in *Machine Learning for Signal Processing (MLSP), 2014 IEEE International Workshop on*, Sept 2014, pp. 1–6.

[15] J. Chen, S.-Y. Tu, and A. Sayed, "Distributed optimization via diffusion adaptation," in *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2011 4th IEEE International Workshop on*, Dec 2011, pp. 281–284.

[16] J. Chen and A. Sayed, "On the benefits of diffusion cooperation for distributed optimization and learning," in *Signal Processing Conference (EUSIPCO), 2013 Proceedings of the 21st European*, Sept 2013, pp. 1–5.

[17] Z. Towfic and A. Sayed, "Adaptive penalty-based distributed stochastic convex optimization," *Signal Processing, IEEE Transactions on*, vol. 62, no. 15, pp. 3924–3938, Aug 2014.

[18] A. Sayed, "Adaptive networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, April 2014.

[19] X. Zhao and A. Sayed, "Asynchronous adaptation and learning over networks — part i: Modeling and stability analysis," *Signal Processing, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2014.

[20] A. Nedic, A. Ozdaglar, and P. Parrilo, "Constrained consensus and optimization in multi-agent networks," *Automatic Control, IEEE Transactions on*, vol. 55, no. 4, pp. 922–938, April 2010.

[21] M. Zhu and S. Martinez, "On distributed convex optimization under inequality and equality constraints," *Automatic Control, IEEE Transactions on*, vol. 57, no. 1, pp. 151–164, Jan 2012.

[22] J. Lin, "Divergence measures based on the shannon entropy," *IEEE Transactions on Information theory*, vol. 37, pp. 145–151, 1991.

[23] S. Oh, L. Schenato, P. Chen, and S. Sastry, "Tracking and coordination of multiple agents using sensor networks: System design, algorithms and experiments," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 234–254, Jan 2007.

[24] P. Dames and V. Kumar, "Cooperative multi-target localization with noisy sensors," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 1877–1883.

[25] Schutze and Hinrich., *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[26] C. Shannon, "A mathematical theory of communication," *Bell System Technical Journal, The*, vol. 27, no. 3, pp. 379–423, July 1948.

[27] F. Nielsen and R. Nock, *Skew Jensen-Bregman Voronoi Diagrams*. Springer Berlin Heidelberg, 2011.

[28] H. W. Kuhn and A. W. Tucker, "Nonlinear programming," in *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and*

Probability, 1950. University of California Press, Berkeley and Los Angeles, 1951, pp. 481–492.

[29] M. Hestenes, "Multiplier and gradient methods," *Journal of Optimization Theory and Applications*, vol. 4, no. 5, pp. 303–320, 1969. [Online]. Available: http://dx.doi.org/10.1007/BF00927673

[30] M. J. D. Powell, "A method for nonlinear constraints in minimization problems," in *Optimization (Sympos., Univ. Keele, Keele, 1968).* Academic Press, London, 1969, pp. 283–298.

[31] M. Powell, "On search directions for minimization algorithms," *Mathematical Programming*, vol. 4, no. 1, pp. 193–201, 1973. [Online]. Available: http://dx.doi.org/10.1007/BF01584660

[32] J. E. Falk, "Lagrange multipliers and nonlinear programming," *J. Math. Anal. Appl.*, vol. 19, pp. 141–159, 1967.

[33] G. He, "Rosen's gradient projection with discrete steps," *Acta Mathematicae Applicatae Sinica*, vol. 6, no. 1, pp. 1–10, 1990. [Online]. Available: http://dx.doi.org/10.1007/BF02014710

[34] J. Rosen, "The gradient projection method for nonlinear programming. part i. linear constraints," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 1, pp. 181–217, 1960. [Online]. Available: http://dx.doi.org/10.1137/0108011

[35] W. Rudin, *Principles of mathematical analysis.* McGraw-Hill New York, 1964, vol. 3.

[36] C. Jorge, M. Sonia, K. Timur, and B. Francesco, "Coverage control for mobile sensing networks," *Robotics and Automation, IEEE Transactions on*, vol. 20, no. 2, pp. 243 – 255, 2004.

[37] A. T. Y. Stergiopoulos, "Spatially distributed area coverage optimisation in mobile robotic networks with arbitrary convex anisotropic patterns," *Automatica: A journal of IFAC the International Federation of Automatic Control*, vol. 49, no. 1, pp. pgs. 232–237, 2013.

[38] S. Cheng, G. Liu, L.and Feng, Y. Wang, and Q. Gao, "Persistent awareness coverage control for mobile sensor networks," *Automatica*, vol. 49, no. 6, p. 18671873, 2013.

[39] R. Durrett, "Probability: Theory and examples," *CAMBRIDGE U PRESS*, 2011.

Zhengguo Sheng is currently a Lecturer with the Department of Engineering and Design, University of Sussex, U.K. He has authored over 50 international conference and journal papers. His current research interests cover IoT/M2M, vehicular communications, and edge/cloud computing.

Daxin Tian [M'13, SM'16] is an associate professor in the School of Transportation Science and Engineering, Beihang University, Beijing, China. His current research interests include mobile computing, intelligent transportation systems, vehicular ad hoc networks, and swarm intelligent.

Jianshan Zhou received the B.Sc. and M.Sc. degrees in traffic information engineering and control in 2013 and 2016, respectively. He is currently working toward the Ph.D degree with the School of Transportation Science and Engineering, Beihang University, Beijing, China. His current research interests are focused on wireless communication, artificial intelligent system, and intelligent transportation systems.