

Parallelized Active Information Gathering Using Multisensor Network for Environment Monitoring

Bin Du¹, Kun Qian¹, Christian Claudel, and Dengfeng Sun¹, *Member, IEEE*

Abstract—Motivated by the application of environment monitoring, this article studies a novel algorithmic framework for solving, in a parallelized manner, the problem of multisensor active information gathering. Unlike the existing methods relying on the fully connected sensor network and sequential processing, our approach builds on the generic network topology and enables individual sensors to simultaneously make their decisions by communicating with the immediate neighbors. Leveraging the cooperation among the multisensor network, we show that the computational complexity of the proposed parallelized algorithm can be greatly reduced compared to the sequential updating schemes, and meanwhile, the suboptimality of obtained solutions is guaranteed. The interconnection between the algorithm complexity and solution quality is explicitly established with respect to the network topology. Based on such interconnection, we further provide the approach to design the optimal sensor network by the given time budget of the algorithm execution. Finally, numerical simulations on a methane emission monitoring scenario are presented to validate the effectiveness of our approach.

Index Terms—Distributed algorithms, environment monitoring, motion planning, multi-sensor network.

I. INTRODUCTION

RECENTLY, mobile sensors, due to their remarkable features, such as mobility, flexibility, and accessibility, have found various applications, including surveillance [1], [2], coverage control [3], [4], environment, and health monitoring [5]–[7]. In particular, it typically provides superior coverage if a team of mobile sensors, rather than a single one, is employed. For instance, as suggested in [8], considering that the cooperation among the multiple sensors enables the scaling up of sources of the observations in both time and space, more informative viewpoints might be obtained through the team of sensors. Such a coordinated team also has the potential to accomplish missions whose total computational workload goes

Manuscript received December 14, 2020; revised February 10, 2021; accepted March 23, 2021. Date of publication April 30, 2021; date of current version February 10, 2022. Manuscript received in final form March 24, 2021. This work was supported in part by the National Science Foundation (NSF) CPS under Grant 1739964, in part by NSF CIS under Grant 1636154, and in part by the Texas Department of Transportation (TxDOT) under Project 0-7007. Recommended by Associate Editor P. Mhaskar. (*Bin Du and Kun Qian contributed equally to this work.*) (*Corresponding author: Bin Du*)

Bin Du and Dengfeng Sun are with the School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN 47907 USA (e-mail: du185@purdue.edu; dsun@purdue.edu).

Kun Qian and Christian Claudel are with the Department of Civil, Architectural, and Environmental Engineering, The University of Texas at Austin, Austin, TX 78712 USA (e-mail: kunqian@utexas.edu; christian.claudel@utexas.edu).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TCST.2021.3069175>.

Digital Object Identifier 10.1109/TCST.2021.3069175

beyond the capability of every single one. On this account, the cooperation protocol among the multisensor team is supposed to be specifically devised to achieve the advantages.

In this article, we focus on the application of environment monitoring and are particularly interested in solving the multisensor active information gathering problem. As a motivating example, let us consider a real-world methane emission monitoring scenario, which is illustrated in Fig. 1 and will be further simulated in the numerical simulations (see Section VI). In order to actively monitor the unknown dynamical methane field (subject to process noise), we deploy a team of mobile sensors whose dynamics are assumed to be known. More specifically, our goal here is to determine the optimal control inputs for all involved mobile sensors such that the generated trajectories allow them to take measurements on the unknown environment, which provides as much information as possible. Consider that each mobile sensor can only measure the environment within a limited sensing range, depending on its real-time position and orientation. Therefore, in order to gather the most amount of information, the multiple sensors are expected to spatially approach the distinct leaking sources, as shown in the figure. To avoid the potential conflict, one will have to take into account the peer-to-peer communication within the sensor network. When it comes to the cooperation between different sensors, the considered problem becomes undoubtedly more complicated than the single sensor case due to the coupling of multiple sensors' trajectories and their measurements on the environment. On the other hand, however, such multiple sources of observations also provide more informative viewpoints for perceiving the unknown environment. Thus, given both the challenges and advantages, the multisensor active information gathering problem has attracted increasingly significant attention in the works of recent years.

A. Literature Review

The problem of active information gathering has been broadly studied in the literature [8]–[19]. For the sake of completeness, we start the literature review by first mentioning the centralized approaches [9], [10], which are proposed when only single sensor is involved. Building on the assumptions that the sensor's measurement is linear with respect to the environment states and sensing noises have the white Gaussian nature, the sensor trajectory optimization problem can be reformulated into a deterministic optimal control problem via

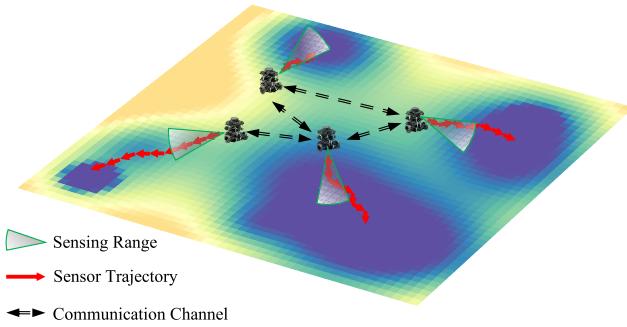


Fig. 1. Real-world methane emission monitoring scenario.

the technique of the Kalman filter. Subsequently, to solve the resulting deterministic problem, the well-known forward value iteration (FVI) method is applied in [9] by leveraging the fact that the sensor can only select its control input from a finite set at each time step. In this sense, the baseline of such an FVI-based method is to construct a search tree of all possible control sequences and then to find the sensor's optimal trajectory that gives the maximum/minimum objective function value. Considering that every possible control sequence needs to be evaluated to obtain the optimal trajectory, a critical concern of this approach is its exponential complexity in terms of the time horizon although it guarantees optimality of the obtained solution. Based on this, the idea of discarding unnecessary nodes in the search tree is adopted [10]. It is shown that the proposed reduced value iteration (RVI) algorithm has lower computational complexity, while its performance remains comparable to the FVI-based method.

Despite the fact that the RVI algorithm could decrease the computational amount (with respect to the time horizon) for the problem with multiple sensors as well, however, the size of the set of possible control inputs still scales exponentially with respect to the number of sensors. In this sense, such an RVI algorithm is not applicable for the problem with the sensor network, and thus, the decentralized variant of the RVI algorithm is studied in [11] by utilizing the idea of coordinate descent. While the coordinate descent scheme is capable of reducing the computational complexity from exponential to linear (with a suboptimal solution), we remark that such scheme inherently relies on the specific network topology and, meanwhile, requires the multiple sensors to sequentially make their decisions (see details in Section II-B). Starting from these two issues, Schlotfeldt *et al.* [12] attempt to fix the first one by using the technique of distributed Kalman filter [14]; however, the second one remains unsolved and could lead to critical concerns. Considering that, when one sensor is making its own decision, all others in the network need to be idled and wait for it, this is essentially a waste of resources in the perspective of parallel computing. Motivated by this, we here aim to develop a parallelized framework, in which multiple sensors perform the computation in parallel over the generic network. By doing so, the computational complexity of our parallelized algorithm is expected to be primarily reduced compared to the existing sequential updating schemes.

It is also worth noting that, while the coordinate descent scheme is highly related to the Gauss–Seidel iteration [20],

our algorithm is essentially based on the colored Gauss–Seidel method [21], which enables parallel computing at each iteration. As a counterpart of the Gauss–Seidel iteration, the Jacobi method can be naturally parallelized in shared-memory computing architecture and, thus, has been widely studied in the area of parallel/distributed processing [22], [23]. Nevertheless, we emphasize that the Jacobi method is not applicable for solving our problem here, since the sensor network is assumed to follow the generic topology and the optimization problem in our setup is essentially considered in the discrete domain.

Besides the above-reviewed approaches that are mainly focused on the problem of active information gathering, another noteworthy line of works has also been found in the literature [24]–[30] which solves the standard (multi)target tracking problem. Although these two types of problems are related in which the unknown states of the target(s) or environment are expected to be estimated by utilizing the sensors' measurements, we remark that they are significantly distinguishable in the following sense. While the target tracking problem typically assumes a set of static sensors and more focuses on the development of state estimation techniques, the problem of interest in this article couples the unknown state estimation with a sophisticated control or optimization procedure, which plans the optimal trajectory for each of the dynamical sensors. In this sense, one major challenge of our problem is to solve for the sensors' optimal control inputs that have the combinatorial nature as explained above.

B. Summary of Contributions

We summarize the contributions of this article as follows.

- 1) Building on the generic sensor network topology, a parallelized algorithm is developed for solving the active information gathering problem. As distinct from the existing methods [11], [12] that rely on sequential processing, the proposed algorithm enables simultaneous updates over the multisensor network.
- 2) While suboptimality guarantee might be compromised for the obtained solutions, the computational complexity of our parallelized algorithm is primarily reduced against the sequential updating schemes. Considering the trade-off between the complexity of the algorithm and the quality of solutions, their interconnection is explicitly established in terms of the sensor network topology.
- 3) Moreover, in order to achieve the best suboptimality guarantee of the obtained solution, the approach to design an optimal network topology is further provided by the given time budget of algorithm execution.

The rest of this article is organized as follows. Section II presents the formulation of the multisensor active information gathering problem and introduces the coordinate descent scheme. Our parallelized algorithm is developed in Section III. Section IV establishes the performance analysis of the proposed algorithm. Section IV provides the approach to design the optimal network topology. Section VI demonstrates the effectiveness of our algorithm via numerical simulations on a real-world methane emission monitoring scenario. The conclusion of this article is presented in Section VII.

II. PROBLEM STATEMENT

Let us consider a network of n mobile sensors and assume that each sensor $i \in \mathcal{I} := \{1, 2, \dots, n\}$ obeys the dynamics

$$\mathbf{x}_{i,t+1} = f_i(\mathbf{x}_{i,t}, \mathbf{u}_{i,t}). \quad (1)$$

Here, $\mathbf{x}_{i,t} \in \mathbb{R}^{d_x}$ represents the state of the i th sensor at the discrete time step t ; $\mathbf{u}_{i,t} \in \mathcal{U}_{i,t}$ denotes the control input; and $\mathcal{U}_{i,t} \subset \mathbb{R}^{d_u}$ is a finite set of the control inputs that the sensor i can possibly take.

Suppose that the goal of such a sensor network is to monitor the unknown dynamical environment. For instance, as shown in the motivating example (see Fig. 1), the environment corresponds to the unknown methane field in which multiple moving leaking sources might be present. Note that, here, we are interested in the model of the whole methane field, rather than the individual leaking sources. It is further assumed that the environment state follows the linear dynamics:

$$\mathbf{y}_{t+1} = A_t \mathbf{y}_t + \mathbf{w}_t, \quad \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, W_t) \quad (2)$$

where $A_t \in \mathbb{R}^{d_y \times d_y}$ is the system parameter and the process noise $\mathbf{w}_t \in \mathbb{R}^{d_y}$ follows the Gaussian distribution with covariance $W_t \in \mathbb{R}^{d_y \times d_y}$. To monitor the unknown environment state \mathbf{y}_t , we assume that each sensor has the following measurement model:

$$\mathbf{z}_{i,t} = H_{i,t}(\mathbf{x}_{i,t}) \mathbf{y}_t + \mathbf{v}_{i,t}(\mathbf{x}_{i,t}), \quad \mathbf{v}_{i,t} \sim \mathcal{N}(\mathbf{0}, V_{i,t}(\mathbf{x}_{i,t})). \quad (3)$$

Here, $\mathbf{z}_{i,t} \in \mathbb{R}^{d_z}$ denotes the data measured by the i th sensor; $H_{i,t}(\mathbf{x}_{i,t}) \in \mathbb{R}^{d_z \times d_y}$ is the sensing matrix; and $\mathbf{v}_{i,t}(\mathbf{x}_{i,t}) \in \mathbb{R}^{d_z}$ is the sensing noise following Gaussian distribution with zero mean and covariance $V_{i,t}(\mathbf{x}_{i,t}) \in \mathbb{R}^{d_z \times d_z}$. We note that the measurement model (3) is assumed to be linear with respect to the environment state \mathbf{y}_t , and such a requirement can be relaxed by linearizing the potential nonlinear model around the estimated state [12]. In addition, both the sensing matrix and noise are assumed to be dependent on the sensor's state $\mathbf{x}_{i,t}$, as the measurement might be related to the sensor's position and/or orientation. Nevertheless, for simplicity, shorthand notations $H_{i,t}$, $\mathbf{v}_{i,t}$ and $V_{i,t}$ are later used to denote the sensing matrix, noise, and covariance, respectively, where we take the dependence on $\mathbf{x}_{i,t}$ into consideration implicitly. Besides, we assume that the noise $\mathbf{v}_{i,t}$ is uncorrelated for different sensors.

A. Centralized Active Information Gathering

Let us start with a centralized perspective and consider that the local information tuple $(\mathbf{x}_{i,t}, \mathbf{z}_{i,t})$ can be collected and utilized globally. Thus, we denote the collected global information as $\mathbf{x}_t^{\mathcal{T}} \in \mathbb{R}^{nd_x}$ and $\mathbf{z}_t^{\mathcal{T}} \in \mathbb{R}^{nd_z}$, which stacks all local $\mathbf{x}_{i,t}$'s and $\mathbf{z}_{i,t}$'s, respectively. Note that the superscript \mathcal{T} , representing the entire sensor network, is used to denote the collected information. Given that the sensing noise is assumed to be uncorrelated for distinct sensors, thus, the measurement model for the network can be simply written as

$$\mathbf{z}_t^{\mathcal{T}} = H_t^{\mathcal{T}} \mathbf{y}_t + \mathbf{v}_t^{\mathcal{T}} \quad (4)$$

where the overall measurement matrix $H_t^{\mathcal{T}} \in \mathbb{R}^{nd_z \times d_y}$ stacks each $H_{i,t}$ and also the global sensing noise $\mathbf{v}_t^{\mathcal{T}} \in \mathbb{R}^{nd_z}$ follows the Gaussian distribution $\mathcal{N}(\mathbf{0}, V_t^{\mathcal{T}})$ with the covariance being:

$$V_t^{\mathcal{T}} := \text{diag}\{V_{1,t}, V_{2,t}, \dots, V_{n,t}\} \in \mathbb{R}^{nd_z \times nd_z}. \quad (5)$$

On this account, the centralized active information gathering problem can be formulated as the following optimization:

$$(\mathbf{P}_0^{\mathcal{T}}) : \max_{\{\mathbf{u}_{1:n,t}\}_{t \in \mathcal{T}}} \mathbb{I}(\mathbf{y}_{T+1}; \{\mathbf{z}_{1:n,t}\}_{t \in \mathcal{T}}) \quad (6a)$$

$$\text{s.t.} \quad \mathbf{x}_{i,t+1} = f_i(\mathbf{x}_{i,t}, \mathbf{u}_{i,t}) \quad (6b)$$

$$\mathbf{y}_{t+1} = A_t \mathbf{y}_t + \mathbf{w}_t \quad (6c)$$

$$\mathbf{z}_t^{\mathcal{T}} = H_t^{\mathcal{T}} \mathbf{y}_t + \mathbf{v}_t^{\mathcal{T}} \quad (6d)$$

$$\mathbf{u}_{i,t} \in \mathcal{U}_{i,t}, \quad i \in \mathcal{I}, \quad t \in \mathcal{T}. \quad (6e)$$

The objective function $\mathbb{I}(\mathbf{y}_{T+1}; \{\mathbf{z}_{1:n,t}\}_{t \in \mathcal{T}})$ denotes the mutual information, where $\{\mathbf{z}_{1:n,t}\}_{t \in \mathcal{T}}$ represents the set of measurements within the time horizon $\mathcal{T} := \{0, 1, \dots, T-1\}$. As distinct from the target tracking problem which primarily focuses on the estimation of the unknown environment state, we emphasize that $(\mathbf{P}_0^{\mathcal{T}})$ here aims to find the optimal control sequences $\{\mathbf{u}_{1:n,t}\}_{t \in \mathcal{T}}$ such that the mutual information between the environment state \mathbf{y}_{T+1} and all sensors' measurements $\{\mathbf{z}_{1:n,t}\}_{t \in \mathcal{T}}$ is maximized. For this purpose, it is natural that the multiple sensors, as shown in the motivating example (see Fig. 1), will intend to approach the distinct targets.

In fact, such (centralized) active information gathering problem has been widely studied in the literature, e.g., [10] and [9]. An early attempt [10] reduces the original problem with stochasticity into the following deterministic optimization:

$$\max_{\{\mathbf{u}_{1:n,t}\}_{t \in \mathcal{T}}} \log \det(\Sigma_T^{\mathcal{T}}) \quad (7a)$$

$$\text{s.t.} \quad \mathbf{x}_{i,t+1} = f_i(\mathbf{x}_{i,t}, \mathbf{u}_{i,t}) \quad (7b)$$

$$\Theta_{t+1}^{\mathcal{T}} = A_t(\Sigma_t^{\mathcal{T}})^{-1} A_t^{\top} + W_t \quad (7c)$$

$$\Sigma_{t+1}^{\mathcal{T}} = (H_{t+1}^{\mathcal{T}})^{\top} (V_{t+1}^{\mathcal{T}})^{-1} H_{t+1}^{\mathcal{T}} + (\Theta_{t+1}^{\mathcal{T}})^{-1} \quad (7d)$$

$$\mathbf{u}_{i,t} \in \mathcal{U}_{i,t}, \quad i \in \mathcal{I}, \quad t \in \mathcal{T}. \quad (7e)$$

Note that the constraints (7c) and (7d) are due to the information form of the Kalman filter. As a consequence of the above formulation (7), the FVI algorithm is developed in [9] by constructing a search tree of all possible control sequences and then finding the optimal one, which gives the maximum objective function value. Although this approach guarantees the global optimality of the obtained solution, however, since multiple sensors are involved in the problem setup, it is clear that the size of the search tree will grow exponentially as the number of sensors increasing. Thus, to further reduce the algorithm complexity in the multisensor setting, an idea of the coordinate descent, also known as the Gauss–Seidel iteration, is proposed in [11]. The parallelized algorithm proposed in this article is also inspired by such a coordinate descent scheme.

B. Coordinate Descent Scheme

Let us first simplify the formulation of optimization (7) by introducing the notion of reachable set. It is observed that the

objective function, as shown in (7d), is mainly determined by the key terms $(H_{t+1}^{\mathcal{I}})^{\top}(V_{t+1}^{\mathcal{I}})^{-1}H_{t+1}^{\mathcal{I}}$. Thus, we introduce a new variable $\Gamma_t^{\mathcal{I}} \in \mathbb{R}^{nd_z \times nd_z}$, called *information matrix*, to denote such terms, i.e.,

$$\Gamma_t^{\mathcal{I}} := (H_t^{\mathcal{I}})^{\top}(V_t^{\mathcal{I}})^{-1}H_t^{\mathcal{I}}, \quad t \in \mathcal{T}. \quad (8)$$

In addition, we refer to the set $\{\Gamma_t^{\mathcal{I}}\}_{t \in \mathcal{T}}$ as the *information trajectory*, which contains all information matrices within the time horizon. Note that this information trajectory is decided by the sensors' states $\{\mathbf{x}_t^{\mathcal{I}}\}_{t \in \mathcal{T}}$ or inherently their control inputs $\{\mathbf{u}_{1:n,t}\}_{t \in \mathcal{T}}$. Thus, we further define $\mathcal{R}_t^{\mathcal{I}} \subseteq \mathbb{R}^{nd_z \times nd_z}$ as the reachable set of the information matrix $\Gamma_t^{\mathcal{I}}$ at the time step t , given that each sensor's initial state is $\mathbf{x}_{i,0}$'s, i.e.,

$$\mathcal{R}_t^{\mathcal{I}} := \left\{ (H_t^{\mathcal{I}})^{\top}(V_t^{\mathcal{I}})^{-1}H_t^{\mathcal{I}} \mid \mathbf{x}_{i,\tau+1} = f_i(\mathbf{x}_{i,\tau}, \mathbf{u}_{i,\tau}), \mathbf{u}_{i,\tau} \in \mathcal{U}_{i,\tau}, \tau \in \{0, 1, \dots, t-1\}, i \in \mathcal{I} \right\}. \quad (9)$$

As a result, the deterministic optimization model (7) can be equivalently rewritten into

$$(\mathbf{P}^{\mathcal{I}}) : \max_{\{\mathbf{u}_{i:n,t}\}_{t \in \mathcal{T}}} \log \det(\Sigma_T^{\mathcal{I}}) \quad (10a)$$

$$\text{s.t.} \quad \Sigma_{t+1}^{\mathcal{I}} = \left(A_t (\Sigma_t^{\mathcal{I}})^{-1} A_t^{\top} + W_t \right)^{-1} + \Gamma_{t+1}^{\mathcal{I}} \quad (10b)$$

$$\Gamma_{t+1}^{\mathcal{I}} \in \mathcal{R}_{t+1}^{\mathcal{I}}, \quad t \in \mathcal{T}. \quad (10c)$$

Note that, with a slight abuse of the notation, we have absorbed the decision variables $\{\mathbf{u}_{i:n,t}\}_{t \in \mathcal{T}}$ into the reachable sets $\mathcal{R}_t^{\mathcal{I}}$, and it can be interpreted as follows. The entire sensor network aims at finding the optimal information trajectory $\{\Gamma_t^{\mathcal{I}}\}_{t \in \mathcal{T}}$ from the reachable sets such that the objective function is optimized; at the same time, the reachable sets are determined by the sensors' control sequences $\{\mathbf{u}_{i:n,t}\}_{t \in \mathcal{T}}$. It should be emphasized that we are still standing in the centralized perspective when considering the problem $(\mathbf{P}^{\mathcal{I}})$ since the reachable set $\mathcal{R}_t^{\mathcal{I}}$ combines all information from the entire network. Next, those couplings will be decomposed by leveraging the idea of coordinate descent.

Since the sensing covariance $V_t^{\mathcal{I}}$ is in the form of block diagonal, as shown in (5), by defining the local information matrix $\Gamma_{i,t} := H_{i,t}^{\top} V_{i,t}^{-1} H_{i,t} \in \mathbb{R}^{nd_z}$, then the global $\Gamma_t^{\mathcal{I}}$ admits the equation $\Gamma_t^{\mathcal{I}} = \sum_{i=1}^n \Gamma_{i,t}$. Keeping this in mind, the coordinate descent scheme performs the following sequential computing where each sensor only takes charge of its own decision. Precisely, let us assume, without loss of generality, that the multiple sensors make their decisions on the control sequences in the ascending order. In other words, when the i th sensor starts its own computation, the first $i-1$ sensors have decided their control sequences $\{\mathbf{u}_{j,t}^c\}_{j \in \mathcal{I}, t \in \mathcal{T}}$, $j = 1, 2, \dots, i-1$, as shown in Fig. 2. Suppose that these $i-1$ prior decisions are immediately available for the i th sensor; then, the coordinate descent scheme produces the i th sensor's control sequences

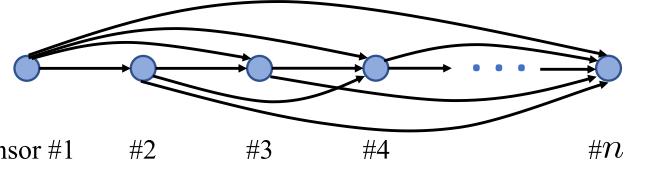


Fig. 2. Underlying topology of the coordinate descent scheme.

$\{\mathbf{u}_{i,t}^c\}_{t \in \mathcal{T}}$ by solving the following local optimization (\mathbf{P}_i) :

$$(\mathbf{P}_i) : \max_{\{\mathbf{u}_{i,t}\}_{t \in \mathcal{T}}} \log \det(\Sigma_{i,T}) \quad (11a)$$

$$\text{s.t.} \quad \Sigma_{i,t+1} = (A_t \Sigma_{i,t}^{-1} A_t^{\top} + W_t)^{-1} + \sum_{j=1}^{i-1} \Gamma_{j,t+1}^c + \Gamma_{i,t+1} \quad (11b)$$

$$\Gamma_{i,t+1} \in \mathcal{R}_{i,t+1}, \quad t \in \mathcal{T} \quad (11c)$$

where $\mathcal{R}_{i,t} \subseteq \mathbb{R}^{nd_z \times nd_z}$ is the reachable set of the local information matrix $\Gamma_{i,t}$, following the similar definition as (9):

$$\mathcal{R}_{i,t} := \left\{ \Gamma_{i,t} \mid \mathbf{x}_{i,\tau+1} = f_i(\mathbf{x}_{i,\tau}, \mathbf{u}_{i,\tau}), \mathbf{u}_{i,\tau} \in \mathcal{U}_{i,\tau}, \tau \in \{0, 1, \dots, t-1\} \right\}. \quad (12)$$

Remark 1: In fact, the procedure of the above coordinate descent scheme can be viewed from another perspective. That is, each sensor i considers the first $i-1$ sensors as the virtual obstacles when making its own decision and then creates an additional virtual obstacle for the next sensor. In this sense, it is clear to see that the first sensor has the most freedom, and the last one has the most restriction to plan their trajectories.

III. PARALLELIZED SOLUTION METHOD

Before proceeding to the development of our parallelized approach, let us first make a few remarks on the above coordinate descent scheme. It should be noted that the local problem (\mathbf{P}_i) can be still approached by the FVI-based method, and its complexity has been primarily reduced compared to the centralized problem $(\mathbf{P}^{\mathcal{I}})$ since (\mathbf{P}_i) is only involved with the single sensors decision. Precisely, the coordinate descent scheme reduces the complexity from exponential ($\mathcal{O}(|\prod_{i=1}^n \mathcal{U}_i|^T)$) to linear ($\mathcal{O}(\sum_{i=1}^n |\mathcal{U}_i|^T)$), by decomposing the interconnections among the multiple sensors. Furthermore, based on the analysis in [11], another remarkable result of the coordinate descent scheme is that the generated solution achieves 50%-suboptimality in the sense of maximizing the mutual information, i.e.,

$$\mathbb{I}(\mathbf{y}_{T+1}; \{\mathbf{z}_{1:n,t}^c\}_{t \in \mathcal{T}}) \geq \frac{1}{2} \cdot \mathbb{I}(\mathbf{y}_{T+1}; \{\mathbf{z}_{1:n,t}^{\text{opt}}\}_{t \in \mathcal{T}}). \quad (13)$$

Here, $\{\mathbf{z}_{1:n,t}^c\}_{t \in \mathcal{T}}$ represents the set of sensors' measurements resulted from the generated control sequences $\{\mathbf{u}_{1:n,t}^c\}_{t \in \mathcal{T}}$, and $\{\mathbf{z}_{1:n,t}^{\text{opt}}\}_{t \in \mathcal{T}}$ corresponds to the optimal solution obtained by solving the centralized problem (7) directly. Although the 50%-suboptimal solution is guaranteed, we remark that the scheme has critical drawbacks in the following two aspects.

- Given that the subproblems (\mathbf{P}_i) need to be solved sequentially, thus, the multisensor structure is not inherently utilized in the sense of parallel computing.

2) As shown in Fig. 2, a specific network topology (similar to a fully connected graph) is required to perform the sequential updating. However, as shown in the motivating example, the general connections among the sensors can hardly obey the specific topology.

In light of these two issues regarding the network topology and parallel computing, this article aims to develop a parallelized approach building on the generic sensor network. Next, we start with a general model of the network topology.

A. Sensor Network Topology

Let us consider the sensor network as a general fixed undirected graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$. Here, $\mathcal{I} = \{1, 2, \dots, n\}$ is the set of n nodes and $\mathcal{E} \subset \mathcal{I} \times \mathcal{I}$ represents the set of undirected edges. An edge connecting the node $i \in \mathcal{I}$ and node $j \in \mathcal{I}$, denoted by $(i, j) \in \mathcal{E}$, implies that there is a communication channel between the corresponding two sensors. Each node's self-loop is excluded from the set of edges; that is, $(i, i) \notin \mathcal{E} \forall i \in \mathcal{I}$. It is emphasized that here we do not assume the network to be fully connected but consider a generally connected graph, stated as the following assumption.

Assumption 1: The time-invariant undirected graph \mathcal{G} is assumed to be connected, i.e., there exists at least one path connecting every pair of two distinct nodes.

In addition, once the nodes of graph are indexed by their labels $\{1, 2, \dots, n\}$, we restrict, by convention, the information flow from node i to j to satisfy with $i < j$. In this sense, we have actually reduced the original undirected graph \mathcal{G} into a new directed one, denoted by $\mathcal{G}^+ = (\mathcal{I}, \mathcal{E}^+)$, where the set of directed edges is defined as $\mathcal{E}^+ := \{(i, j) | (i, j) \in \mathcal{E}, i < j\}$. Furthermore, we use the set $\mathcal{N}_i^+ := \{j | (j, i) \in \mathcal{E}^+\}$ to represent the neighbors of the i th node, meaning that the sensor i receives information from sensors in the set \mathcal{N}_i^+ . Finally, to establish the performance analysis for our parallelized approach, we will need the following important notions in graph theory. A subset of nodes in the undirected graph \mathcal{G} is said to be a *clique* if there exists an edge between every two distinct nodes in the subset. Also, a *clique cover* is a partition of the nodes in the graph \mathcal{G} if each set of the partition forms a clique. We use $\alpha(\mathcal{G}) \in \mathbb{Z}_+$, called *clique cover number* of the undirected graph \mathcal{G} , to denote the minimum number of sets that are needed to form a clique cover.

B. Parallelized Algorithm

With the topology of the generic sensor network specified, we are now in the position to present our parallelized solution method. Let us recall that the original centralized active information gathering problem has been simplified into the deterministic optimization $(\mathbf{P}^{\mathcal{I}})$; subsequently, the idea of coordinate descent inherently considers the first $i - 1$ sensors as the virtual obstacles when the sensor i makes its own decision (see Remark 1). Next, we leverage the similar idea and introduce a new notion called *partial active information gathering problem*. Instead of involving the first $i - 1$ sensors' decisions in the i th sensor's computation, we here take its neighbors \mathcal{N}_i^+ into consideration. In other words, the virtual obstacles are generated based on the neighbors' information

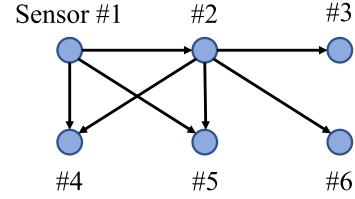


Fig. 3. Parallelized execution over a generic network.

Algorithm 1 Parallelized Active Information Gathering

Initialization:

- (I.1) Set the initial iteration index $k = 1$;
- (I.2) Each sensor i solves $(\mathbf{P}^{\mathcal{N}_i^+})$ with $\Gamma_{j,t}[0] = 0$, and obtains the local solution $\{\mathbf{u}_{i,t}^p[1]\}_{t \in \mathcal{T}}$;
- (I.3) Record the information trajectory $\{\Gamma_{i,t}[1]\}_{t \in \mathcal{T}}$ based on $\{\mathbf{u}_{i,t}^p\}_{t \in \mathcal{T}}$, and send it to neighbors;
- (I.4) Set the maximum iteration number K .

while $1 \leq k \leq K$ is satisfied **do**

 Each sensor i **simultaneously** does

- (S.1) Receive the required data $\{\Gamma_{j,t}[k]\}_{t \in \mathcal{T}}$ from its neighbors $j \in \mathcal{N}_i^+$;
- (S.2) Solve $(\mathbf{P}^{\mathcal{N}_i^+})$ based on the received data;
- (S.3) Update the solution $\{\mathbf{u}_{i,t}^p[k+1]\}_{t \in \mathcal{T}}$, record the information trajectory $\{\Gamma_{i,t}[k+1]\}_{t \in \mathcal{T}}$ and send it to neighbors;
- (S.4) Let $k \leftarrow k + 1$, and continue.

end

when the sensor i makes its decisions. As a result, the partial active information gathering problem for each sensor i is defined as the following optimization:

$$\left(\mathbf{P}^{\mathcal{N}_i^+} \right) : \max_{\{\mathbf{u}_{i,t}\}_{t \in \mathcal{T}}} \log \det(\Sigma_{i,T}) \quad (14a)$$

$$\text{s.t.} \quad \Sigma_{i,t+1} = (A_t \Sigma_{i,t}^{-1} A_t^\top + W_t)^{-1} + \sum_{j \in \mathcal{N}_i^+} \Gamma_{j,t+1}[k] + \Gamma_{i,t+1} \quad (14b)$$

$$\Gamma_{i,t+1} \in \mathcal{R}_{i,t+1}, \quad t \in \mathcal{T}. \quad (14c)$$

Compared to the previous local problem (\mathbf{P}_i) in the coordinate descent scheme, the primary difference here is that the recursion of variable $\Sigma_{i,t+1}$ in (14b) involves only the neighbors' information matrices $\Gamma_{j,t+1}[k]$, $j \in \mathcal{N}_i^+$, rather than the first $i - 1$ sensors' information matrices. Note that we use an additional index k to denote the iterations since, later on, we will upgrade our approach into an iterative scheme.

It is worth pointing out that such a change into the dependence on the neighbors' information makes our approach parallelizable since the information from neighbors can be immediately obtained given the sensor network topology. For instance, as shown in the motivating example (see Fig. 1), the four mobile sensors will be able to simultaneously make their decisions and then exchange the information through the existing communication channels. Such parallelism is also the motivation why we would like to upgrade our scheme to be iterative. As another illustrative example, Fig. 3 demonstrates, more precisely, the execution of our parallelized algorithm,

where a simple network of six sensors is considered. At the initialization stage of the algorithm, each of the six sensors in the network greedily solves the partial active information gathering problem and makes its own decision on the control sequence, without considering the neighbors' influence. After that, the local decisions (in the form of information trajectories) are sent to the neighbors. By doing so, at the next step, the six sensors can simultaneously update their decisions based on the received data from neighbors and then send out the updated decision again. Such a *receive-update-send process* will be repeated for a few iterations until the stability of sensors' decisions is achieved. Here, stability means that every sensor in the network will not change its decision if the iteration of the algorithm continues. Ideally, we expect that the number of iterations needed to achieve the stability is less than the number of sensors so that the proposed parallelized algorithm saves execution time compared to the coordinate descent scheme. In Section IV, we will rigorously establish the upper bound of the needed iterations and further show how good the stable decisions will be, compared to the optimal solution. At the end of this section, we outline the complete parallelized approach as Algorithm 1 and make a few remarks regarding the algorithm implementation.

Remark 2: It should be pointed out that there will be some fake updates in the execution of Algorithm 1 if we let every sensor solve the partial problem $(\mathbf{P}^{\mathcal{N}_i^+})$ at each iteration. For example, as shown in Fig. 3, sensor #1 does not receive any information from the neighbors during the whole process; thus, its local decision has reached stability immediately after the initialization. According to this fact, it will be a waste of computing resources if we still let the sensor solve and update its decision after it reaches stability. To fix this issue, the appropriate implementation of our parallelized algorithm is based on the event-trigger execution. That is, we let the sensor's computing process be triggered by the received data at each iteration. In this sense, it can be easily verified that the sensor #1 will be idled after the initialization since it will be never triggered by any input data. In addition, as another advantage of such event-trigger-based implementation, the execution of the algorithm can be realized asynchronously since there is no need to enforce every sensor to complete its computation simultaneously.

Remark 3: Algorithm 1 is actually based on the directed graph \mathcal{G}^+ since each sensor needs to identify the set of neighbors \mathcal{N}_i^+ by comparing its own label i and the other's j . Even though such directed graph \mathcal{G}^+ can be easily induced by the original undirected graph \mathcal{G} using some labeling strategies, different labeling of \mathcal{G} may lead to different complexities of the algorithm implementation. In Section IV, we show that the upper bound of the number of the needed parallel iterations is indeed dependent on the labeling strategy; however, the suboptimality guarantee of the obtained solution is actually determined by the original graph \mathcal{G} . We should also remark that, with a fully connected network, arbitrary labeling of the graph \mathcal{G} results in the identical directed \mathcal{G}^+ , as shown in Fig. 2. As a consequence, our parallelized algorithm will be reduced to the standard coordinate descent scheme, and thus, the 50%-suboptimality will be achieved (see details in Section IV).

IV. PERFORMANCE ANALYSIS

In this section, we theoretically analyze the performance of our parallelized approach over the generic sensor network. In particular, by characterizing the network topology, we explicitly establish the theory of: 1) the needed number of parallel iterations to reach the stability of decisions (see Theorem 1) and 2) the suboptimality guarantee achieved by the stable decisions (see Theorem 2).

Let us start with showing by the first theorem that the number of needed parallel iterations to achieve the stable decisions can be upper bounded in terms of the network topology.

Theorem 1: Suppose that Algorithm 1 is applied to solve the problem $(\mathbf{P}_0^{\mathcal{I}})$ over the labeled directed graph \mathcal{G}^+ . Let $\{\mathbf{u}_{i,t}^p[k]\}_{t \in \mathcal{T}}$ be the i th sensor's decision on the control sequence generated at iteration k ; then, these decisions will reach the stability on the sequence $\{\mathbf{u}_{i,t}^p\}_{t \in \mathcal{T}}$ after at most T^{\max} iterations, i.e., $\forall i \in \mathcal{I}, t \in \mathcal{T}$

$$\mathbf{u}_{i,t}^p[k] = \mathbf{u}_{i,t}^p, \quad k \geq T^{\max} \quad (15)$$

where T^{\max} can be expressed by

$$T^{\max} = \delta(\mathcal{G}^+) \quad (16)$$

with $\delta(\mathcal{G}^+)$ being the depth of the directed graph \mathcal{G}^+ .

Proof: See Appendix A. \blacksquare

Note that Theorem 1 proves that the minimum number of needed parallel iterations to achieve the stability of decisions can be upper bounded by the graph depth $\delta(\mathcal{G}^+)$. The directed graph \mathcal{G}^+ is induced by labeling the original undirected graph \mathcal{G} . Next, we further show that the quality of the stable decisions can be lower bounded by the original graph \mathcal{G} , rather than the induced \mathcal{G}^+ . It should be remarked that a similar bound is derived in the context of distributed submodular maximization by using the notion of fractional independence number (see [31, Th. 2]). As distinct from the result presented in [31], our bound is provided in terms of the graph clique cover number $\alpha(\mathcal{G})$ directly, and furthermore, we build the interconnection between $\alpha(\mathcal{G})$ and the graph depth $\delta(\mathcal{G}^+)$. It is unknown how $\delta(\mathcal{G}^+)$ is related to the fractional independence number. Before proceeding to the theorem, let us first characterize the quality of solutions by defining the notion called γ -suboptimality.

Recall that $\{\mathbf{u}_{1:n,t}^{\text{opt}}\}_{t \in \mathcal{T}}$ represents the optimal control sequence obtained by solving the original problem $(\mathbf{P}_0^{\mathcal{I}})$ directly; $\{\mathbf{z}_{1:n,t}^{\text{opt}}\}_{t \in \mathcal{T}}$ is the set of optimal sensor measurements derived by the solution $\{\mathbf{u}_{1:n,t}^{\text{opt}}\}_{t \in \mathcal{T}}$. Let $\{\mathbf{u}_{1:n,t}^p\}_{t \in \mathcal{T}}$ (correspondingly $\{\mathbf{z}_{1:n,t}^p\}_{t \in \mathcal{T}}$) be the solutions generated by our parallelized algorithm; then, the quality of the solution can be characterized by the following ratio:

$$\gamma = \frac{\mathbb{I}(\mathbf{y}_{T+1}; \{\mathbf{z}_{1:n,t}^p\}_{t \in \mathcal{T}})}{\mathbb{I}(\mathbf{y}_{T+1}; \{\mathbf{z}_{1:n,t}^{\text{opt}}\}_{t \in \mathcal{T}})}. \quad (17)$$

Since the maximization of mutual information $\mathbb{I}(\cdot)$ is considered in the problem $(\mathbf{P}_0^{\mathcal{I}})$ originally, it can be verified that the measure of suboptimality must have $\gamma \leq 1$, and the larger γ provides the better quality of the solution $\{\mathbf{z}_{1:n,t}^p\}_{t \in \mathcal{T}}$.

Theorem 2: Suppose that Algorithm 1 is applied to solve the problem $(\mathbf{P}_0^{\mathcal{I}})$ over the original undirected graph \mathcal{G} .

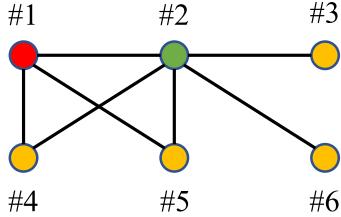


Fig. 4. Undirected graph with colors.

Let $\{\mathbf{u}_{1,n,t}^p\}_{t \in \mathcal{T}}$ (correspondingly $\{\mathbf{z}_{1,n,t}^p\}_{t \in \mathcal{T}}$) be the stable solutions generated after T^{\max} iterations; then, it holds that the generated solution satisfies the γ -suboptimality with

$$\gamma \geq \frac{1}{1 + \alpha(\mathcal{G})} \quad (18)$$

where $\alpha(\mathcal{G})$ is the clique cover number of graph \mathcal{G} . \blacksquare

Proof: See Appendix B. \blacksquare

It should be noted that the above analysis (see Theorems 1 and 2) is basically consistent with the result of the coordinate descent scheme for solving the active information gathering problem. As remarked in Section II-B, the procedure of coordinate descent inherently assumes a fully connected graph. In this sense, it takes $\delta(\mathcal{G}^+) = n$ steps to finish algorithm execution and, thus, achieves 50%-suboptimal solution as $\alpha(\mathcal{G}) = 1$. We emphasize that the more general results are here provided: building on the generic sensor network, the proposed algorithm saves the execution time by leveraging the parallel computing but with the suboptimality guarantee compromised. This can be interpreted as a tradeoff between the time complexity and solution accuracy.

In addition, we have also mentioned that, even though the suboptimal guarantee of the solutions is established with respect to the undirected graph \mathcal{G} , the algorithm implementation and, thus, its execution time is decided by the labeled directed graph \mathcal{G}^+ . Since the same graph \mathcal{G} may result in different directed \mathcal{G}^+ 's according to the labeling strategy, a natural question here is how to obtain an optimal \mathcal{G}^+ by given the original undirected graph. We next address this issue by designing a specific labeling strategy.

V. DESIGN OF THE OPTIMAL NETWORK TOPOLOGY

In this section, the following two questions will be answered: 1) what is the optimal graph \mathcal{G}^+ in the sense of minimum depth $\delta(\mathcal{G}^+)$, given the original graph \mathcal{G} and 2) conversely, provided the budget of parallel iterations $T^{\max} = \delta(\mathcal{G}^+)$, what is the optimal sensor network topology in the sense that solutions are obtained with best suboptimality guarantee, i.e., minimum clique cover number $\alpha(\mathcal{G})$.

To approach the first question, we build on the technique called *node coloring* in the graph theory. Given an undirected graph \mathcal{G} , the node coloring is the procedure of assigning colors to each node in \mathcal{G} such that no adjacent nodes are of the same color. Let us still take the previous example, as shown in Figs. 3 and 11. It is clear that the color of nodes can be assigned as illustrated in Fig. 4 with the requirement satisfied. As a result of the node coloring, an appropriate labeling strategy of nodes can be specified based on the assigned colors. Suppose that, totally, C colors are assigned in the graph \mathcal{G} ,

and with each color $c \in \{1, 2, \dots, C\}$, there are I_c nodes. Subsequently, we label the total $\sum_{c=1}^C I_c$ nodes in \mathcal{G} by the following procedure: 1) first, label the I_1 nodes that have the same color $c = 1$ with labels $\{1, 2, \dots, I_1\}$; 2) label the nodes in color $c = 2$ with $\{I_1+1, I_1+2, \dots, I_1+I_2\}$; and 3) continue the process until the last color. Fig. 4 also shows the result of such labeling procedure. By doing so, it is easy to see that the depth of the labeled directed graph \mathcal{G}^+ is equal to the number of colors. Thus, the problem of seeking minimum depth $\delta(\mathcal{G}^+)$ is equivalent to finding the minimum number of colors (termed the chromatic number $\chi(\mathcal{G})$) in the graph \mathcal{G} . Such a minimization problem is also known as the famous *graph coloring* problem. Note that, even though the graph coloring problem is proven to be NP-complete [32], a variety of algorithms and their applications have been extensively studied in the literature (see [33] and [34]).

In addition, considering that the suboptimality guarantee of our algorithm is provided in terms of $\alpha(\mathcal{G})$, we next give a simple connection between the chromatic number $\chi(\mathcal{G})$ and the clique cover number $\alpha(\mathcal{G})$.

Proposition 1: Consider the generic connected undirected graph \mathcal{G} consisting of n nodes; then, its chromatic number $\chi(\mathcal{G})$ and clique cover number $\alpha(\mathcal{G})$ satisfy

$$\chi(\mathcal{G}) \leq n - \alpha(\mathcal{G}) + 1. \quad (19)$$

Proof: See Appendix C. \blacksquare

According to the above analysis, we know that the minimum number of needed parallel iterations for our algorithm is $T^{\max} = \chi(\mathcal{G})$, and the chromatic number $\chi(\mathcal{G})$ can be obtained by solving the graph coloring problem. Furthermore, Proposition 1 tells that, for any undirected graph \mathcal{G} , there exists a labeling strategy such that $n - \alpha(\mathcal{G}) + 1$ parallel iterations are sufficient to achieve the $1/(1 + \alpha(\mathcal{G}))$ -suboptimal solution.

The next question to be answered here is how to design the optimal network topology in the sense of obtaining the best suboptimality guarantee for our algorithm when the budget of parallel iterations T^{\max} is given. The previous analysis has shown that the minimum number of needed parallel iterations is determined by the chromatic number $\chi(\mathcal{G})$. Thus, such a question is equivalent to asking how to design the network topology with minimum $\alpha(\mathcal{G})$ when $\chi(\mathcal{G})$ is provided.

We first show, in the following proposition, that $\alpha(\mathcal{G})$ can be lower bounded when the budget $T^{\max} = \chi(\mathcal{G})$ is given.

Proposition 2: Consider the generic connected undirected graph \mathcal{G} consisting of n nodes; then, its chromatic number $\chi(\mathcal{G})$ and clique cover number $\alpha(\mathcal{G})$ satisfy

$$\alpha(\mathcal{G}) \geq \lceil n/\chi(\mathcal{G}) \rceil \quad (20)$$

where $\lceil \cdot \rceil$ is the ceiling function. \blacksquare

Proof: See Appendix D. \blacksquare

Proposition 2 states that the clique cover number of graph $\alpha(\mathcal{G})$ has to be no less than $\lceil n/\chi(\mathcal{G}) \rceil$. Interestingly, when we are finishing this work, we noticed that a new paper [35] appears with the similar result presented. The quantity $\lceil n/\chi(\mathcal{G}) \rceil$ is interpreted as concurrence in [35], and by doing so, the optimal network is devised to obtain the best suboptimality guarantee. In the following, we start from a different perspective: the obtained network is guaranteed to be optimal

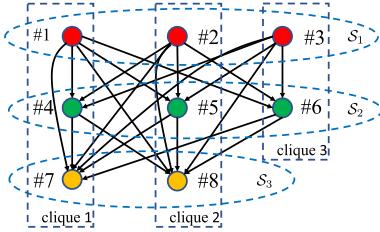


Fig. 5. Optimal network topology with fixed $\chi(\mathcal{G})$.

in the sense that the lower bound in Proposition 2 is exactly hit, i.e., $\alpha(\mathcal{G}) = \lceil n/\chi(\mathcal{G}) \rceil$. Next, we provide the approach to design such an optimal network topology.

Let us first partition the n nodes into $\chi(\mathcal{G})$ sets, indexed by $i \in \{1, 2, \dots, \chi(\mathcal{G})\}$, with each of the first $S - 1$ sets has $\lceil n/\chi(\mathcal{G}) \rceil$ nodes and the last one has $\text{mod}^+(n, \chi(\mathcal{G}))$ nodes. Note that here $\text{mod}^+(a, b)$ means the remainder of a divided by b , and in particular, $\text{mod}^+(a, b) = a/b$ if a is a integer multiple of b . Suppose that we denote each of the sets as S_i ; then, the partition of n nodes can be exactly represented as

$$S_i = \begin{cases} \{(i-1)\lceil n/\chi(\mathcal{G}) \rceil + j \mid j = 1, 2, \dots, \lceil n/\chi(\mathcal{G}) \rceil\}, & \text{if } i \leq \chi(\mathcal{G}) - 1 \\ \{(i-1)\lceil n/\chi(\mathcal{G}) \rceil + j \mid j = 1, 2, \dots, \text{mod}^+(n, \chi(\mathcal{G}))\}, & \text{if } i = \chi(\mathcal{G}). \end{cases} \quad (21)$$

Next, we connect the n nodes in the graph by following the subsequent two rules: 1) there is no edge connecting two nodes in the same individual set S_i and 2) each node in S_i connects every node in the prior sets $\cup_{j < i} S_j$. By doing so, all nodes in the same set S_i can be assigned with the identical color, and thus, the graph has $\chi(\mathcal{G})$ colors. In addition, the set of nodes $\mathcal{K}_i = \{j \mid \text{mod}^+(j, \lceil n/\chi(\mathcal{G}) \rceil) = i\}$ forms a clique of the graph, and thus, the total number of cliques is $\lceil n/\chi(\mathcal{G}) \rceil$. Fig. 5 provides an illustration of the optimal network topology with $n = 8$ and $\chi(\mathcal{G}) = 3$. It can be immediately verified that following a such design of network topology, the resulting graph has the chromatic number $\chi(\mathcal{G})$ and, meanwhile, hits the lower bound of clique cover number, i.e., $\alpha(\mathcal{G}) = \lceil n/\chi(\mathcal{G}) \rceil$.

VI. SIMULATION

In this section, we validate the proposed parallelized algorithm by simulating the methane emission monitoring problem, which has been previously introduced at the beginning of this article (see Fig. 1) and also been investigated in [10] and [36]. As distinct from simulations in [10] and [36], we here deploy a group of the so-called *Gasbots*, and suppose that each of them is equipped with a tunable laser absorption spectroscopy (TDLAS) sensor [36], which is capable of measuring the integral gas concentration over the path of the laser beam. Our ultimate goal is to plan trajectories of the gasbot team such that an accurate estimate of the methane field can be obtained.

A. Simulation Setup

Suppose that the target methane emission field is described by a $d_m \times d_m$ lattice, as shown in the background of Fig. 6,

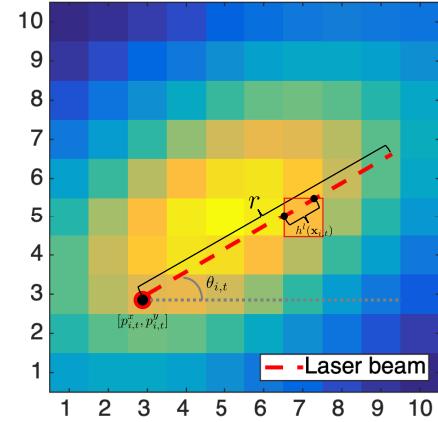


Fig. 6. TDLAS sensor measurement.

with each cell $l \in \{1, 2, \dots, d_m^2\}$ assigned a value of the gas concentration y_t^l in the unit of parts per million (ppm). Overall, the state of the methane field $\mathbf{y}_t = [y_t^1, y_t^2, \dots, y_t^{d_y}]^\top$ is assumed to have the linear dynamics $\mathbf{y}_{t+1} = A_t \mathbf{y}_t + \mathbf{w}_t$. In particular, we consider two different scenarios for this simulation: 1) a static and noise-free environment, i.e., $A_t \equiv I_{d_y \times d_y}$ and $\mathbf{w}_t \equiv \mathbf{0}$ and 2) a noisy dynamical environment governed by the diffusion equation (see details in Section VI-C). Note that the dimension of the environment state is equal to the number of cells in the methane field, i.e., $d_y = d_m^2$. In order to estimate the unknown environment state \mathbf{y}_t , each gasbot i measures a scalar $z_{i,t} \in \mathbb{R}$, representing the integral gas concentration, based on the sensor's state $\mathbf{x}_{i,t}$ at the time step t

$$z_{i,t} = H(\mathbf{x}_{i,t})^\top \mathbf{y}_t + v_{i,t}. \quad (22)$$

Here, the sensor's state $\mathbf{x}_{i,t} = [p_{i,t}^x, p_{i,t}^y, \theta_{i,t}]^\top \in \mathbb{R}^3$ is composed of two components: 1) the position of the gasbot $[p_{i,t}^x, p_{i,t}^y]^\top \in \mathbb{R}^2$ and 2) the orientation of the laser beam $\theta_{i,t}$. Moreover, the sensing parameter $H(\mathbf{x}_{i,t}) \in \mathbb{R}^{d_y}$ is a vector dependent on the sensor's state $\mathbf{x}_{i,t}$, where its l th element measures the distance $h^l(\mathbf{x}_{i,t})$ that is traveled by the laser beam in the l th cell. In addition, each laser beam is assumed to have a limited sensing range r . Fig. 6 exemplifies the specification of elements $h^l(\mathbf{x}_{i,t})$'s in the sensing vector $H(\mathbf{x}_{i,t})$. Consequently, the sensor's measurement model can be equivalently expressed as

$$z_{i,t} = \sum_{l=1}^{d_y} h^l(\mathbf{x}_{i,t}) y_t^l + v_{i,t} \quad (23)$$

where the sensing noise $v_{i,t}$ follows the Gaussian distribution $\mathcal{N}(\mathbf{0}, V)$ with zero mean and fixed variance $V \in \mathbb{R}$ independent on the sensor and time step.

In this simulation, we consider the methane emission monitoring problem in a 20×20 lattice. In other words, the state of environment has dimension $d_y = 400$. We generate the methane field via Gaussian distributions that are centered at four distinct leak sources, i.e., [3, 3.5], [3, 18], [17.5, 3.5], and [16, 16.5]. To explore the unknown methane field, a network of six gasbots is employed; each of them is capable of measuring the environment states via (23) and, meanwhile,

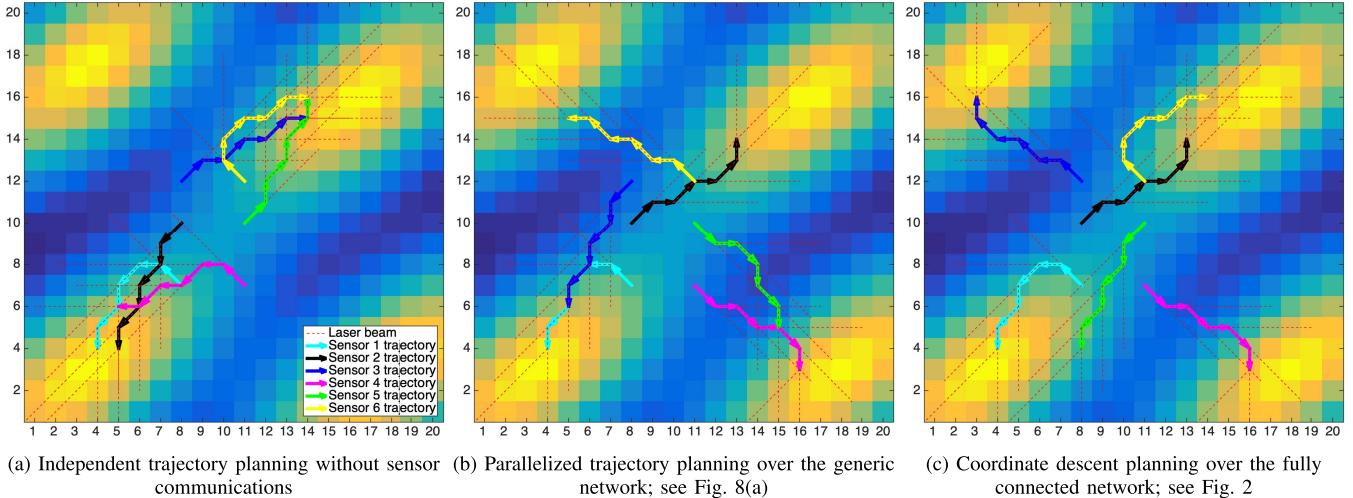


Fig. 7. Comparison of trajectory planning results using different schemes. (a) Independent trajectory planning without sensor communications. (b) Parallelized trajectory planning over the generic network [see Fig. 8(a)]. (c) Coordinate descent planning over the fully connected network [see Fig. 2].

communicating with its neighbors. Suppose that the TDLAS sensor in each gasbot measures the integral gas concentration $z_{i,t}$ within range $r = 5$ and the sensing noise $v_{i,t}$ is Gaussian distributed with variance $V = 0.01$. The team of gasbots starts from six distinct original positions, i.e., [8, 7], [8, 10], [8, 12], [11, 7], [11, 10], and [11, 12]. We next specify their dynamics as follows. At the first time step $t = 0$, each gasbot i determines its original orientation $\theta_{i,0}$ by choosing one angle from the set $\mathcal{U}_{i,0} = \{0, \pm\pi/4, \pm\pi/2, \pm3\pi/4, \pi\}$. Subsequently, at the next time steps, the gasbot moves one cell along with the prefixed angle $\theta_{i,t-1}$ and, meanwhile, changes its orientation by rotating $u_{i,t} \in \mathcal{U}_{i,t} = \{0, \pm\pi/4\}^\circ$, i.e.,

$$p_{i,t}^x = p_{i,t-1}^x + \text{sgn}(\sin(\theta_{i,t-1})) \quad (24a)$$

$$p_{i,t}^y = p_{i,t-1}^y + \text{sgn}(\cos(\theta_{i,t-1})) \quad (24b)$$

$$\theta_{i,t} = \theta_{i,t-1} + u_{i,t}, \quad u_{i,t} \in \mathcal{U}_{i,t}. \quad (24c)$$

In this sense, at each time step t , each gasbot i only needs to plan its orientation by determining the angle input $u_{i,t}$, and thus, the size of control input spaces has $|\mathcal{U}_{i,0}| = 8$ and $|\mathcal{U}_{i,t}| = 3 \forall t > 0$.

B. Performance of the Trajectory Planning

The planned trajectories of the team of six gasbots are demonstrated in Fig. 7 when considering the static and noise-free environment. We here compare three different trajectory planning scenarios: 1) each sensor independently decides its own trajectory without communicating with others; 2) the team of sensors cooperatively plans the trajectories based on our parallelized algorithm, over the network, as shown in Fig. 8(a); and 3) the coordinate descent scheme is adopted by assuming the fully connected network topology. In addition, we assign each sensor with the identical initialized matrix $\Sigma_{i,0} \in \mathbb{R}^{d_y \times d_y}$ for all three planning scenarios. As similar to the previous works that focus on the environment state estimation [37], [38], we initialize the covariance matrix such that its (m, n) th entry is dependent on the distance between cells m and n , as well as the scale of the initialized value

of y_t^m and y_t^n . The intuition behind this is that spatially close states tend to have stronger correlations, and meanwhile, the correlation between two states is more likely to be affected by noises when they are spatially far away from each other. It should be emphasized that, while, totally, six iterations are required for the execution of coordinate descent scheme, our parallelized algorithm only costs three parallel iterations due to the fact that the considered sensor network has depth $\delta(\mathcal{G}^+) = 3$. It can be observed from Fig. 7(a) that the independent trajectory planning result only explores two out of four leak sources since each sensor determines its own area of interest without considering others. On the contrary, Fig. 7(b) and (c) shows that all four leak sources are covered when the cooperation among sensors is utilized both in our algorithm and the coordinate descent scheme. It can be also seen that fewer overlaps are present in the cooperatively planned trajectories [see Fig. 7(b) and (c)] compared to the independent planning result [see Fig. 7(a)]. Furthermore, the trajectories generated by our parallelized algorithm are very close to the ones obtained by the coordinate descent scheme, despite the fact that only one-half of the computing iterations is demanded for our algorithm.

C. Performance of the Mutual Information Maximization and State Estimation

To further validate the effectiveness of the proposed parallelized algorithm, we next quantitatively evaluate the performance of mutual information maximization and state estimation by comparing three different schemes: 1) the myopic algorithm, also known as the greedy method, in which the six sensors cooperatively plan their trajectories for only one time step; 2) the parallelized nonmyopic scheme, where the trajectories for next five time steps are optimized as a whole, by using our parallelized algorithm; and 3) the coordinate descent nonmyopic scheme, i.e., the five-step trajectories are optimized based on the coordinate descent. Moreover, in order to evaluate the approach, we propose to design the optimal network

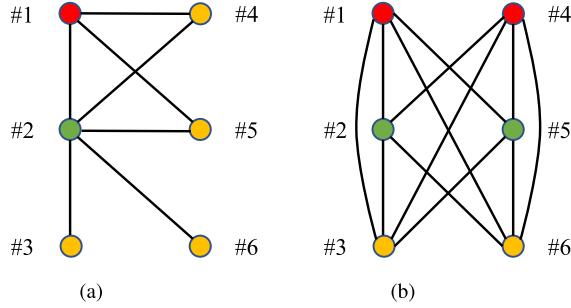


Fig. 8. Two adopted network topologies. (a) General topology. (b) Optimal topology.

topology (see Section IV), and the parallelized nonmyopic scheme is simulated based on two different graphs separately: 1) the general graph that is adopted previously [see Fig 8(a)] and 2) the designed optimal graph [see Fig. 8(b)]. We emphasize that both networks have the same depth $\delta(\mathcal{G}^+) = 3$, and thus, the number of needed parallel iterations will be identical. However, the latter one is expected to have the better performance since it has the smaller clique cover number $\alpha(\mathcal{G}) = 2$. The performance of these schemes is investigated in the following two aspects: 1) maximization of the mutual information and 2) estimation of the unknown environment states. Note that the mutual information is characterized by the log-determinant of the matrices $\Sigma_t^{\mathcal{I}}$'s, as shown in the objective function in optimization problem (7). In addition, according to the well-known Kalman filter, the state estimation $\hat{\mathbf{y}}_t \in \mathbb{R}^{d_y}$ can be obtained by the following iterations:

$$\hat{\mathbf{y}}_{t+1} = \hat{\mathbf{y}}_t + (\Sigma_t^{\mathcal{I}})^{-1} \left((H_t^{\mathcal{I}})^{\top} (V_t^{\mathcal{I}})^{-1} \mathbf{z}_t^{\mathcal{I}} - \Gamma_t^{\mathcal{I}} \hat{\mathbf{y}}_t \right) \quad (25a)$$

$$\Sigma_{t+1}^{\mathcal{I}} = \Sigma_t^{\mathcal{I}} + \Gamma_{t+1}^{\mathcal{I}}. \quad (25b)$$

Here, the initial guess of the environment state $\hat{\mathbf{y}}_0$ is generated by imposing random shifts on the positions of leak sources and also the covariance of the methane spatial distributions. We note that, such initialization is commonly adopted in the applications of the Kalman filter. In practice, this information can be obtained prior by tracking the environment for a short period of time.

As distinct from the previous simulation, we here consider both static and dynamical methane fields. In particular, it is assumed that the dynamical environment is governed by the well-known diffusion equation [39], [40], and it can be discretized by the finite difference discretization [41], which yields the following dynamics:

$$y_{t+1}^{i,j} = y_t^{i,j} + \rho \left(y_t^{i+1,j} + y_t^{i,j+1} + y_t^{i-1,j} + y_t^{i,j-1} - 4y_t^{i,j} \right). \quad (26)$$

Note that $y_t^{i,j} \in \mathbb{R}$ denotes the value of gas concentration on the (i, j) th cell at the time step t , and $\rho > 0$ represents the rate of diffusion. By assuming the zero boundary conditions, the above discretized diffusion equation can be expressed the compact linear form $\mathbf{y}_{t+1} = A_t \mathbf{y}_t + \mathbf{w}_t$, where we additionally consider the process noise \mathbf{w}_t following the Gaussian distribution $\mathcal{N}(\mathbf{0}, W_t)$. Specifically, we set the rate of diffusion as $\rho = 0.01$ and assume the covariance matrix as

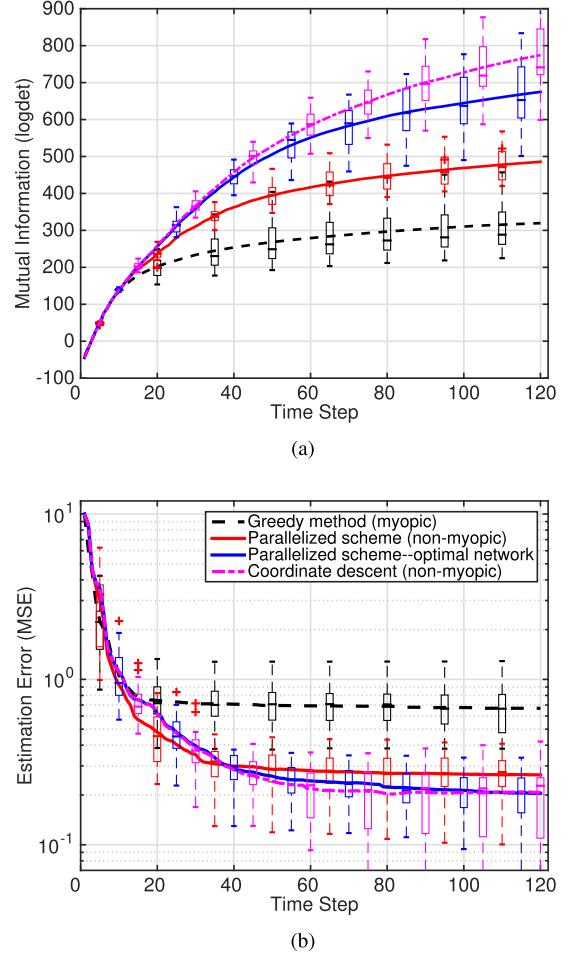


Fig. 9. Static case: comparison of different schemes. (a) Mutual information. (b) State estimation.

$W_t = 0.01 \cdot I_{d_y \times d_y}$. The initial environment state \mathbf{y}_0 is generated as the same in Section VI-B.

The obtained numerical results for both static and dynamical environments are demonstrated in Figs. 9 and 10, in which each of the three schemes is run for 120 time steps. Since each scheme is simulated for 20 Monte Carlo trials, both mean values and variance are reported in the figures. It can be seen from Figs. 9(a) and 10(a) that the team of sensors guided by the coordinate descent scheme collects the highest mutual information, while it also requires the most densely connected network topology. Although less mutual information is gathered by our parallelized algorithm compared to the coordinate descent, only sparse connections among the sensors are present, and fewer iterations are required to obtain the solution. In particular, with the optimally designed network, the mutual information collected by our algorithm is close to the coordinate descent scheme, and we emphasize that the computational complexity of our algorithm is only one-half of the coordinate descent. It is not surprising that both nonmyopic schemes outperform the greedy method in terms of the performance on maximizing mutual information. In addition, the algorithm performances of state estimation are illustrated in Figs. 9(b) and 10(b). It is observed that the performances on state estimation are close for the two nonmyopic schemes; however, the estimation error for the

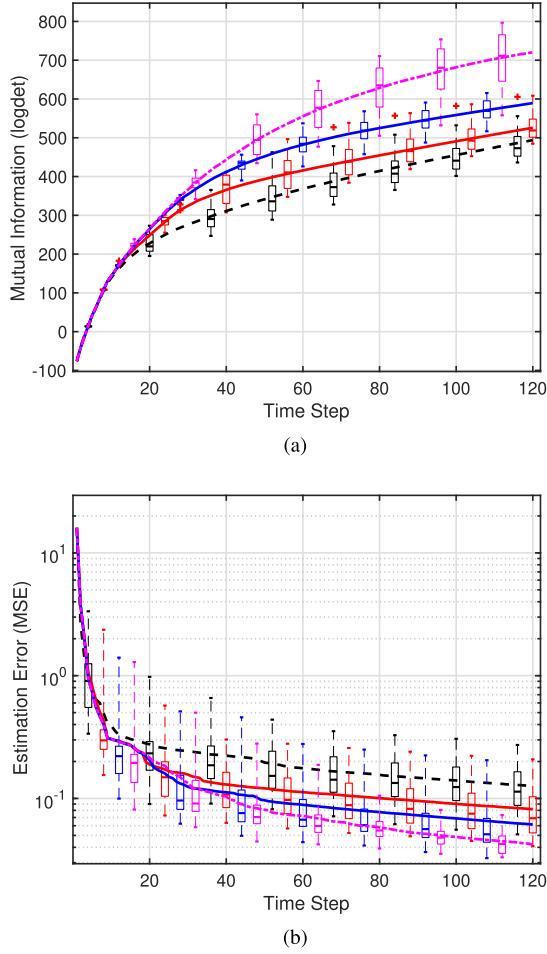


Fig. 10. Dynamical case: comparison of different schemes. (a) Mutual information. (b) State estimation.

myopic scheme is only reduced at the beginning, especially for the static case. This is because trajectories generated by the myopic scheme tend to be stuck in the areas that possess high uncertainties; the two nonmyopic schemes perform similarly as they can collect more viewpoints on the environment and will continue to explore more areas. One can also observe that the estimation error for the dynamical case is, in general, less than the static case; this is because the methane field is inherently diffusing; and thus, the overall magnitude of the gas concentration is decreasing. Finally, the comparison between parallelized and coordinate descent scheme suggests that they could achieve comparable estimation performance, and as expected, the parallelized scheme over the optimal network performs better than the one with a general graph.

VII. CONCLUSION

This article proposes a parallelized approach for solving the multisensor active information gathering problem. As distinct from the coordinate descent-based algorithms, our parallelized approach enables parallel computing over the sensor network and is applicable for any generic network topology. As a result, the computational complexity of our algorithm can be primarily reduced according to the depth of the sensor network. Moreover, in order to guarantee the best quality of obtained solutions, an approach to design the optimal network

topology is further provided by the given time budget of algorithm execution. Simulation results on the methane emission monitoring problem are present to validate the effectiveness of the proposed algorithm.

The directions of future work can be considered from the following two perspectives. First, as a counterpart of the mutual information, the well-known posterior Cramér–Rao lower bound (PCRLB) can be also adopted as a criterion to measure the tracking or monitoring performance. It has been shown that, with less computational cost, the PCRLB-based source seeking approach achieves a similar performance compared to the standard mutual information-based methods. Therefore, our next step is to explore the possibility of using PCRLB to guide the planning of the sensors’ trajectories. In addition, consider that the PCRLB criteria can deal with a more general measurement model, such as the case in which the sensors may fail to measure the unknown environment. Thus, we plan to incorporate the probability of detection into the sensor measurement model so that a more robust approach can be developed against the potential sensor failure or other random interruptions.

APPENDIX

A. Proof of Theorem 1

Let us first clarify the definition of the depth of a directed graph. The distance from the node i to node j , denoted as $d(i, j) \in \mathbb{Z}_+$, is defined as the maximum number of directed edges (longest path) that link the nodes i and j . Note that $d(i, j) = 0$ if there is no path from i to j . Subsequently, the depth of a directed graph \mathcal{G}^+ , denoted as $\delta(\mathcal{G}^+) \in \mathbb{Z}_+$, is defined as one plus the maximum distance between any two linked nodes, i.e., $\delta(\mathcal{G}^+) = 1 + \max_{i, j \in \mathcal{I}} d(i, j)$.

According to the procedure of Algorithm 1, we next introduce a new notation $\mathcal{I}^s(k) \subseteq \mathcal{I}$ to represent the set of sensors that have reached the stability of decisions after k iterations. In addition, we denote $\iota^s(k) = \mathcal{I}^s(k) \setminus \mathcal{I}^s(k-1)$ as the set of sensors that reach the stability at the k th iteration. On this account, it is easy to see that the set $\mathcal{I}^s(1) (= \iota^s(1))$ contains the sensors that do not receive any information from the neighbors; for example, sensor #1 is included in the sets, as shown in Fig. 3. Furthermore, we also know that $\iota^s(k)$ contains the sensors with two properties: 1) the neighbors of sensor $i \in \iota^s(k)$ must satisfy with $\mathcal{N}_i^+ \subseteq \mathcal{I}^s(k-1)$ and 2) the distance from sensor $i \in \iota^s(1)$ to sensor $j \in \iota^s(k)$ must have $d(i, j) = k-1$. As in the previous example, it holds that $\iota^s(2) = \{2\}$ and $\iota^s(3) = \{3, 4, 5, 6\}$. For the sake of illustration, the positions of sensors are rearranged, as shown in Fig. 11.

We are now ready to prove the theorem by contradiction. Suppose that Algorithm 1 does not reach the stability after $T^{\max} = \delta(\mathcal{G}^+)$ iterations. Without loss of generality, we can assume that the sensor j reaches the stability at $(T^{\max} + 1)$ th iteration, i.e., $j \in \iota^s(T^{\max} + 1)$. According to the aforementioned properties of $\iota^s(k)$, it holds that $d(i, j) = T^{\max} \forall i \in \iota^s(1)$, which contradicts the fact $T^{\max} = \delta(\mathcal{G}^+)$.

B. Proof of Theorem 2

We begin the proof by introducing the following fact, which is proven in [11, Lemma 1] and [13, Lemma 2.1].

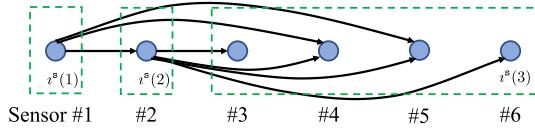


Fig. 11. Parallelism in the view of sequential executions.

Lemma 1: In the problem setup of active information gathering, the mutual information function $\mathbb{I}(\mathbf{y}_{T+1}; \{\mathbf{z}_{1:n,t}\}_{t \in \mathcal{T}})$ is monotonically nondecreasing and submodular. That is, if the sets \mathcal{A} and \mathcal{B} have $\mathcal{A} \subseteq \mathcal{B}$, then the mutual information is nondescreasing

$$\mathbb{I}(\mathbf{y}_{T+1}; \{\mathbf{z}_{\mathcal{A},t}\}_{t \in \mathcal{T}}) \leq \mathbb{I}(\mathbf{y}_{T+1}; \{\mathbf{z}_{\mathcal{B},t}\}_{t \in \mathcal{T}}) \quad (27)$$

submodular

$$\begin{aligned} & \mathbb{I}(\mathbf{y}_{T+1}; \{\mathbf{z}_{\mathcal{A} \cup \mathcal{C},t}\}_{t \in \mathcal{T}}) - \mathbb{I}(\mathbf{y}_{T+1}; \{\mathbf{z}_{\mathcal{A},t}\}_{t \in \mathcal{T}}) \\ & \geq \mathbb{I}(\mathbf{y}_{T+1}; \{\mathbf{z}_{\mathcal{B} \cup \mathcal{C},t}\}_{t \in \mathcal{T}}) - \mathbb{I}(\mathbf{y}_{T+1}; \{\mathbf{z}_{\mathcal{B},t}\}_{t \in \mathcal{T}}). \end{aligned} \quad (28)$$

Based on Lemma 1, we next define the marginal function of mutual information as

$$\begin{aligned} & \Delta(\mathbf{y}_{T+1}; \{\mathbf{z}_{i,t}\}_{t \in \mathcal{T}} | \{\mathbf{z}_{\mathcal{A},t}\}_{t \in \mathcal{T}}) \\ & = \mathbb{I}(\mathbf{y}_{T+1}; \{\mathbf{z}_{i,t} \cup \mathbf{z}_{\mathcal{A},t}\}_{t \in \mathcal{T}}) - \mathbb{I}(\mathbf{y}_{T+1}; \{\mathbf{z}_{\mathcal{A},t}\}_{t \in \mathcal{T}}). \end{aligned} \quad (29)$$

It can be verified that, if $\mathcal{A} \subseteq \mathcal{B}$, then

$$\begin{aligned} & \Delta(\mathbf{y}_{T+1}; \{\mathbf{z}_{i,t}\}_{t \in \mathcal{T}} | \{\mathbf{z}_{\mathcal{A},t}\}_{t \in \mathcal{T}}) \\ & \geq \Delta(\mathbf{y}_{T+1}; \{\mathbf{z}_{i,t}\}_{t \in \mathcal{T}} | \{\mathbf{z}_{\mathcal{B},t}\}_{t \in \mathcal{T}}). \end{aligned} \quad (30)$$

Now, we are in the position to prove the theorem. The following proof follows a similar path as in [31]. For the sake of notational simplicity, we omit the time index in the variables, i.e., simplifying the variables \mathbf{y}_{T+1} and $\{\mathbf{z}_{1:n,t}\}_{t \in \mathcal{T}}$ as \mathbf{y} and $\mathbf{z}_{1:n}$, respectively. Then, it holds that

$$\begin{aligned} & \mathbb{I}(\mathbf{y}, \mathbf{z}_{1:n}^{\text{opt}}) \\ & \stackrel{(31.a)}{\leq} \mathbb{I}(\mathbf{y}, \mathbf{z}_{1:n}^{\text{opt}} \cup \mathbf{z}_{1:n}^{\text{p}}) \\ & = \mathbb{I}(\mathbf{y}, \mathbf{z}_{1:n}^{\text{p}}) + \sum_{i=1}^n (\mathbb{I}(\mathbf{y}, \mathbf{z}_{1:i}^{\text{opt}} \cup \mathbf{z}_{1:n}^{\text{p}}) - \mathbb{I}(\mathbf{y}, \mathbf{z}_{1:i-1}^{\text{opt}} \cup \mathbf{z}_{1:n}^{\text{p}})) \\ & \stackrel{(31.b)}{=} \mathbb{I}(\mathbf{y}, \mathbf{z}_{1:n}^{\text{p}}) + \sum_{i=1}^n \Delta(\mathbf{y}, \mathbf{z}_i^{\text{opt}} | \mathbf{z}_{1:i-1}^{\text{opt}} \cup \mathbf{z}_{1:n}^{\text{p}}) \\ & \stackrel{(31.c)}{\leq} \mathbb{I}(\mathbf{y}, \mathbf{z}_{1:n}^{\text{p}}) + \sum_{i=1}^n \Delta(\mathbf{y}, \mathbf{z}_i^{\text{opt}} | \mathbf{z}_{\{j|j \in \mathcal{N}_i^+\}}^{\text{p}}) \end{aligned} \quad (31)$$

where (31. a) follows from the nondecreasing property of the mutual information; (31. b) is by the definition of marginal function; and (31. c) is due to the inequality (30). Recall that the variable $\mathbf{z}_{1:n}^{\text{p}}$ here corresponds to the stable solution generated by Algorithm 1. We have also shown in the previous proof that, if the sensor $i \in \iota^s(k)$, meaning that it reaches the stability at iteration k , then its neighbors must have reached the stability, i.e., $\mathcal{N}_i^+ \subseteq \mathcal{I}^s(k-1)$; see the first property of the defined stable sets $\iota^s(k)$ and $\mathcal{I}^s(k-1)$. This implies that, at the k th iteration, we have

$$\mathbb{I}(\mathbf{y}, \mathbf{z}_i^{\text{p}} \cup \mathbf{z}_{\{j|j \in \mathcal{N}_i^+\}}^{\text{p}}) \geq \mathbb{I}(\mathbf{y}, \mathbf{z}_i^{\text{opt}} \cup \mathbf{z}_{\{j|j \in \mathcal{N}_i^+\}}^{\text{p}}) \quad (32)$$

and thus

$$\Delta(\mathbf{y}, \mathbf{z}_i^{\text{p}} | \mathbf{z}_{\{j|j \in \mathcal{N}_i^+\}}^{\text{p}}) \geq \Delta(\mathbf{y}, \mathbf{z}_i^{\text{opt}} | \mathbf{z}_{\{j|j \in \mathcal{N}_i^+\}}^{\text{p}}). \quad (33)$$

Therefore, the previous inequality (31) can be continued as

$$\mathbb{I}(\mathbf{y}, \mathbf{z}_{1:n}^{\text{opt}}) \leq \mathbb{I}(\mathbf{y}, \mathbf{z}_{1:n}^{\text{p}}) + \sum_{i=1}^n \Delta(\mathbf{y}, \mathbf{z}_i^{\text{p}} | \mathbf{z}_{\{j|j \in \mathcal{N}_i^+\}}^{\text{p}}). \quad (34)$$

Next, we take the original undirected graph \mathcal{G} into account. Suppose that $\mathcal{C}(\mathcal{G})$ denotes the set of cliques, which covers the graph \mathcal{G} (with minimum number), and thus, we know that $|\mathcal{C}(\mathcal{G})| = \alpha(\mathcal{G})$. Subsequently, it can be shown that, for any clique $\mathcal{K} \in \mathcal{C}(\mathcal{G})$, we have

$$\sum_{i \in \mathcal{K}} \Delta(\mathbf{y}, \mathbf{z}_i^{\text{p}} | \mathbf{z}_{\{j|j \in \mathcal{N}_i^+ \cap \mathcal{K}\}}^{\text{p}}) = \mathbb{I}(\mathbf{y}, \mathbf{z}_{\mathcal{K}}^{\text{p}}). \quad (35)$$

To show this, we assume, without loss of any generality, that the sensors in the set \mathcal{K} are labeled by the numbers from 1 to I , i.e., $\mathcal{K} = \{1, 2, \dots, I\}$. Since \mathcal{K} is a clique, it is easy to see that $\mathcal{N}_i^+ = \{1, 2, \dots, i-1\} \forall i \in \mathcal{K}$. Thus, (35) can be verified by

$$\begin{aligned} \sum_{i \in \mathcal{K}} \Delta(\mathbf{y}, \mathbf{z}_i^{\text{p}} | \mathbf{z}_{\{j|j \in \mathcal{N}_i^+ \cap \mathcal{K}\}}^{\text{p}}) &= \sum_{i=1}^I \Delta(\mathbf{y}, \mathbf{z}_i^{\text{p}} | \mathbf{z}_{1:i-1}^{\text{p}}) \\ &= \sum_{i=1}^I (\mathbb{I}(\mathbf{y}, \mathbf{z}_{1:i}^{\text{p}}) - \mathbb{I}(\mathbf{y}, \mathbf{z}_{1:i-1}^{\text{p}})) \\ &= \mathbb{I}(\mathbf{y}, \mathbf{z}_{\mathcal{K}}^{\text{p}}). \end{aligned} \quad (36)$$

According to (35), the inequality (34) can be further derived as

$$\begin{aligned} \mathbb{I}(\mathbf{y}, \mathbf{z}_{1:n}^{\text{opt}}) &\stackrel{(37.a)}{\leq} \mathbb{I}(\mathbf{y}, \mathbf{z}_{1:n}^{\text{p}}) + \sum_{\mathcal{K} \in \mathcal{C}(\mathcal{G})} \sum_{i \in \mathcal{K}} \Delta(\mathbf{y}, \mathbf{z}_i^{\text{p}} | \mathbf{z}_{\{j|j \in \mathcal{N}_i^+\}}^{\text{p}}) \\ &\stackrel{(37.b)}{\leq} \mathbb{I}(\mathbf{y}, \mathbf{z}_{1:n}^{\text{p}}) + \sum_{\mathcal{K} \in \mathcal{C}(\mathcal{G})} \sum_{i \in \mathcal{K}} \Delta(\mathbf{y}, \mathbf{z}_i^{\text{p}} | \mathbf{z}_{\{j|j \in \mathcal{N}_i^+ \cap \mathcal{K}\}}^{\text{p}}) \\ &\stackrel{(37.c)}{=} \mathbb{I}(\mathbf{y}, \mathbf{z}_{1:n}^{\text{p}}) + \sum_{\mathcal{K} \in \mathcal{C}(\mathcal{G})} \mathbb{I}(\mathbf{y}, \mathbf{z}_{\mathcal{K}}^{\text{p}}) \\ &\stackrel{(37.d)}{\leq} \mathbb{I}(\mathbf{y}, \mathbf{z}_{1:n}^{\text{p}}) + \sum_{\mathcal{K} \in \mathcal{C}(\mathcal{G})} \mathbb{I}(\mathbf{y}, \mathbf{z}_{1:n}^{\text{p}}) \\ &\stackrel{(37.e)}{=} (1 + \alpha(\mathcal{G})) \cdot \mathbb{I}(\mathbf{y}, \mathbf{z}_{1:n}^{\text{p}}). \end{aligned} \quad (37)$$

Note that (37. a) is due to the fact that $\mathcal{C}(\mathcal{G})$ forms a clique cover of the graph; (37. b) and (37. d) follow from the submodularity and nondecreasing properties of the mutual information; (37. c) is due to equation (35); and (37. e) is because of the fact $|\mathcal{C}(\mathcal{G})| = \alpha(\mathcal{G})$. Therefore, the proof is completed.

C. Proof of Proposition 1

We prove this result by mathematical induction. It is clear that, when the graph has $n = 1$ node, the inequality can be immediately verified by the fact that $\chi(\mathcal{G}) = \alpha(\mathcal{G}) = 1$. Suppose that the statement is true when the graph has $n = k$ nodes; it will suffice to show that the inequality holds when $n = k + 1$. Let us denote \mathcal{G}^k as the graph having k nodes, and thus, we know $\chi(\mathcal{G}^k) + \alpha(\mathcal{G}^k) \leq k + 1$. In addition, we use

$\mathcal{G}^k \oplus v$ to denote a new graph obtained by arbitrarily adding a node v into the graph \mathcal{G}^k . Note that there are two cases for the additional node v : 1) it can be added into a clique of the original graph \mathcal{G}^k such that the clique cover number remains unchanged, i.e., $\alpha(\mathcal{G}^k) = \alpha(\mathcal{G}^k \oplus v)$ and 2) it cannot be added into the existing clique, and thus, $\alpha(\mathcal{G}^k \oplus v) = \alpha(\mathcal{G}^k) + 1$. For the first case, considering that some certain clique is extended by adding a new node, the chromatic number of graph has $\chi(\mathcal{G}^k \oplus v) \leq \chi(\mathcal{G}^k) + 1$. For the second case, there must exist a node in the certain clique that is connected with the node v since the considered graph is connected; meanwhile, there must exist another node in the same clique that is disconnected with the node v since v cannot be absorbed in the clique. On this account, the node v can be assigned with the same color as the disconnected node; thus, we have $\chi(\mathcal{G}^k \oplus v) \leq \chi(\mathcal{G}^k)$. Combining these two cases together, it is proven that

$$\chi(\mathcal{G}^k \oplus v) + \alpha(\mathcal{G}^k \oplus v) \leq \chi(\mathcal{G}^k) + \alpha(\mathcal{G}^k) + 1 \leq k + 2. \quad (38)$$

Therefore, the proof is completed.

D. Proof of Proposition 2

We prove the proposition by contradiction. Suppose that there exists a clique cover of the graph \mathcal{G} such that the number of cliques $\bar{\alpha}$ has $\bar{\alpha} \leq \lceil n/\chi(\mathcal{G}) \rceil - 1$. Since the set of cliques covers the graph, it is easy to see that the largest clique \mathcal{K} has the number of nodes $|\mathcal{K}| \geq \lceil n/\bar{\alpha} \rceil$. Meanwhile, considering that \mathcal{K} is a clique, thus, we must have $\chi(\mathcal{G}) \geq \chi(\mathcal{K}) = |\mathcal{K}|$. Consequently, it holds that

$$\chi(\mathcal{G}) \geq \lceil n/\bar{\alpha} \rceil \geq \left\lceil \frac{n}{\lceil n/\chi(\mathcal{G}) \rceil - 1} \right\rceil. \quad (39)$$

Next, let us denote $m = \lceil n/\chi(\mathcal{G}) \rceil$. To prove the contradiction, it will suffice to show that

$$\lceil n/(m-1) \rceil \geq \chi(\mathcal{G}) + 1. \quad (40)$$

We prove (40) by contradiction as well. Suppose that (40) is incorrect, i.e., $\lceil n/(m-1) \rceil \leq \chi(\mathcal{G})$. Then, it implies that $n/(m-1) \leq \chi(\mathcal{G})$, and thus, $n \leq \chi(\mathcal{G}) \cdot (m-1)$. As a result, we have

$$\lceil n/\chi(\mathcal{G}) \rceil \leq \lceil \chi(\mathcal{G}) \cdot (m-1)/\chi(\mathcal{G}) \rceil = m-1 \quad (41)$$

which contradicts the fact $m = \lceil n/\chi(\mathcal{G}) \rceil$. Therefore, the proof is completed.

REFERENCES

- [1] J. Haugen and L. Imsland, "Monitoring moving objects using aerial mobile sensors," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 2, pp. 475–486, Mar. 2016.
- [2] J. You and W. Wu, "Online passive identifier for spatially distributed systems using mobile sensor networks," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 6, pp. 2151–2159, Nov. 2017.
- [3] Y. Kantaros and M. M. Zavlanos, "Distributed communication-aware coverage control by mobile sensor networks," *Automatica*, vol. 63, pp. 209–220, Jan. 2016.
- [4] I. I. Hussein and D. M. Stipanović, "Effective coverage control for mobile sensor networks with guaranteed collision avoidance," *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 4, pp. 642–657, Jul. 2007.
- [5] M. Jalaliha and J. Choi, "Environmental monitoring using autonomous aquatic robots: Sampling algorithms and experiments," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 3, pp. 899–905, May 2013.
- [6] S. Martínez, "Distributed interpolation schemes for field estimation by mobile sensor networks," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 2, pp. 491–500, Mar. 2010.
- [7] K. Qian and C. G. Claudel, "Real-time mobile sensor management framework for city-scale environmental monitoring," 2020, *arXiv:2005.10378*. [Online]. Available: <http://arxiv.org/abs/2005.10378>
- [8] G. Best, O. M. Cliff, T. Patten, R. R. Mettu, and R. Fitch, "Dec-MCTS: Decentralized planning for multi-robot active perception," *Int. J. Robot. Res.*, vol. 38, nos. 2–3, pp. 316–337, 2019.
- [9] J. Le Ny and G. J. Pappas, "On trajectory optimization for active sensing in Gaussian process models," in *Proc. 48th IEEE Conf. Decis. Control (CDC) Held Jointly, 28th Chin. Control Conf.*, Dec. 2009, pp. 6286–6292.
- [10] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, "Information acquisition with sensing robots: Algorithms and error bounds," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 6447–6454.
- [11] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, "Decentralized active information acquisition: Theory and application to multi-robot SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 4775–4782.
- [12] B. Schlotfeldt, D. Thakur, N. Atanasov, V. Kumar, and G. J. Pappas, "Anytime planning for decentralized multirobot active information gathering," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 1025–1032, Apr. 2018.
- [13] J. L. Williams, "Information theoretic sensor management," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, USA, 2007.
- [14] N. Atanasov, R. Tron, V. M. Preciado, and G. J. Pappas, "Joint estimation and localization in sensor networks," in *Proc. 53rd IEEE Conf. Decis. Control*, Dec. 2014, pp. 6875–6882.
- [15] G. A. Hollinger and G. S. Sukhatme, "Sampling-based robotic information gathering algorithms," *Int. J. Robot. Res.*, vol. 33, no. 9, pp. 1271–1287, Aug. 2014.
- [16] Y. Kantaros, B. Schlotfeldt, N. Atanasov, and G. J. Pappas, "Asymptotically optimal planning for non-myopic multi-robot information gathering," in *Proc. Robot., Sci. Syst. (RSS)*, Freiburg, Germany, 2019, pp. 22–26.
- [17] G. A. Hollinger and G. S. Sukhatme, "Sampling-based motion planning for robotic information gathering," *Robot., Sci. Syst.*, vol. 3, no. 5, 2013.
- [18] L. Zhou, V. Tzoumas, G. J. Pappas, and P. Tokekar, "Resilient active target tracking with multiple robots," *IEEE Robot. Autom. Lett.*, vol. 4, no. 1, pp. 129–136, Jan. 2019.
- [19] L. Zhou, V. Tzoumas, G. J. Pappas, and P. Tokekar, "Distributed attack-robust submodular maximization for multi-robot planning," 2019, *arXiv:1910.01208*. [Online]. Available: <http://arxiv.org/abs/1910.01208>
- [20] S. J. Wright, "Coordinate descent algorithms," *Math. Program.*, vol. 151, no. 1, pp. 3–34, Jun. 2015.
- [21] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, vol. 23. Englewood Cliffs, NJ, USA: Prentice-Hall, 1989.
- [22] G. Banjac, K. Margellos, and P. J. Goulart, "On the convergence of a regularized Jacobi algorithm for convex optimization," *IEEE Trans. Autom. Control*, vol. 63, no. 4, pp. 1113–1119, Apr. 2018.
- [23] L. Deori, K. Margellos, and M. Prandini, "Regularized Jacobi iteration for decentralized convex quadratic optimization with separable constraints," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 4, pp. 1636–1644, Jul. 2019.
- [24] A. Sinha, T. Kirubarajan, and Y. Bar-Shalom, "Autonomous ground target tracking by multiple cooperative UAVs," in *Proc. IEEE Aerosp. Conf.*, Mar. 2005, pp. 1–9.
- [25] D. Akselrod, A. Sinha, and T. Kirubarajan, "Information flow control for collaborative distributed data fusion and multisensor multitarget tracking," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 4, pp. 501–517, Jul. 2012.
- [26] R. Tharmarasa, T. Kirubarajan, A. Sinha, and T. Lang, "Decentralized sensor selection for large-scale multisensor-multitarget tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 2, pp. 1307–1324, Apr. 2011.
- [27] X. Gu, X. Cao, Y. Xie, J. Chen, and X. Sun, "Cooperative trajectory planning for multi-UCAV using multiple traveling salesman problem," in *Proc. 35th Chin. Control Conf. (CCC)*, Jul. 2016, pp. 2722–2727.
- [28] R. Tharmarasa, T. Kirubarajan, M. Hernandez, and A. Sinha, "PCRLB-based multisensor array management for multitarget tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 2, pp. 539–555, Apr. 2007.

- [29] R. Tharmarasa, T. Kirubarajan, and M. Hernandez, "Large-scale optimal sensor array management for multitarget tracking," *IEEE Trans. Syst., Man Cybern. C, Appl. Rev.*, vol. 37, no. 5, pp. 803–814, Sep. 2007.
- [30] G. Soldi, F. Meyer, P. Braca, and F. Hlawatsch, "Self-tuning algorithms for multisensor-multitarget tracking using belief propagation," *IEEE Trans. Signal Process.*, vol. 67, no. 15, pp. 3922–3937, Aug. 2019.
- [31] D. Grimsman, M. S. Ali, J. P. Hespanha, and J. R. Marden, "The impact of information in distributed submodular maximization," *IEEE Trans. Control Netw. Syst.*, vol. 6, no. 4, pp. 1334–1343, Dec. 2019.
- [32] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*. Boston, MA, USA: Springer, 1972, pp. 85–103.
- [33] R. Lewis, *A Guide to Graph Colouring*, vol. 7. Berlin, Germany: Springer, 2015.
- [34] P. Formanowicz and K. Tanas, "A survey of graph coloring—its types, methods and applications," *Found. Comput. Decis. Sci.*, vol. 37, no. 3, pp. 223–238, Oct. 2012.
- [35] H. Sun, D. Grimsman, and J. R. Marden, "Distributed submodular maximization with parallel execution," 2020, *arXiv:2003.04364*. [Online]. Available: <http://arxiv.org/abs/2003.04364>
- [36] V. M. H. Bennetts, A. J. Lilienthal, A. A. Khalil, V. P. Sese, and M. Trincavelli, "Towards real-world gas distribution mapping and leak localization using a mobile robot with 3D and remote gas sensing capabilities," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 2335–2340.
- [37] T. M. Hamill, J. S. Whitaker, and C. Snyder, "Distance-dependent filtering of background error covariance estimates in an ensemble Kalman filter," *Monthly Weather Rev.*, vol. 129, no. 11, pp. 2776–2790, 2001.
- [38] K. Qian, A. Mohamed, and C. Claudel, "Physics informed data driven model for flood prediction: Application of deep learning in prediction of urban flood development," 2019, *arXiv:1908.10312*. [Online]. Available: <http://arxiv.org/abs/1908.10312>
- [39] T. Wiedemann, C. Manss, D. Shutin, A. J. Lilienthal, V. Karolj, and A. Viseras, "Probabilistic modeling of gas diffusion with partial differential equations for multi-robot exploration and gas source localization," in *Proc. Eur. Conf. Mobile Robots (ECMR)*, Sep. 2017, pp. 1–7.
- [40] J. Crank, *The Mathematics of Diffusion*. Oxford, U.K.: Oxford Univ. Press, 1979.
- [41] J. C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*. Philadelphia, PA, USA: SIAM, 2004.



Bin Du received the bachelor's degree in electrical engineering and automation from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2014, and the master's degree in control science and engineering from the University of Science and Technology of China, Hefei, China, in 2017. He is currently pursuing the Ph.D. degree with the School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN, USA.

His research interests include distributed control and optimization.



Kun Qian received the bachelor's degree in civil engineering from Tongji University, Shanghai, China, in 2017, and the master's degree in civil systems engineering from the University of California at Berkeley, Berkeley, CA, USA, in 2018. He is currently pursuing the Ph.D. degree with the Department of Civil, Architectural and Environmental Engineering, The University of Texas at Austin (UT-Austin), Austin, TX, USA.

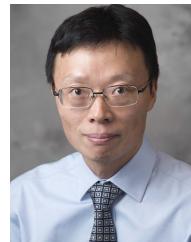
His research interests include multiagent systems, sequential decision processes, machine learning, and state estimation.



Christian Claudel received the master's degree in plasma physics from the École Normale Supérieure de Lyon, Lyon, France, in 2004, and the Ph.D. degree in electrical engineering from the University of California at Berkeley (UC-Berkeley), Berkeley, CA, USA, in 2010.

He is currently an Assistant Professor with the Department of Civil, Architectural and Environmental Engineering, The University of Texas at Austin (UT-Austin), Austin, TX, USA. His research interests include control and estimation of distributed parameter systems and wireless sensor networks.

Dr. Claudel received the Leon Chua Award from UC-Berkeley in 2010 for his work on the mobile millennium traffic monitoring system.



Dengfeng Sun (Member, IEEE) received the bachelor's degree in precision instruments and mechanology from Tsinghua University, Beijing, China, in 2000, the master's degree in industrial and systems engineering from The Ohio State University, Columbus, OH, USA, in 2002, and the Ph.D. degree in civil engineering from the University of California at Berkeley, Berkeley, CA, USA, in 2008.

He is currently an Associate Professor with the School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN, USA. His research interests include the control and optimization of large-scale systems and their applications.