Automatica 155 (2023) 111126

Contents lists available at ScienceDirect

Automatica

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

Maximization of nonsubmodular functions under multiple constraints with applications ☆

Lintao Ye [a,b,c], Zhi-Wei Liu [a,b,c,*], Ming Chi [a,b,c,**], Vijay Gupta [d]

[a] School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China
[b] Key Laboratory of Image Processing and Intelligent Control, Ministry of Education, Wuhan 430074, China
[c] Hubei Key Laboratory of Brain-inspired Intelligent Systems, Wuhan 430074, China
[d] The Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA

ARTICLE INFO

Article history:
Received 18 September 2022
Received in revised form 15 December 2022
Accepted 13 April 2023
Available online 14 June 2023

Keywords:
Combinatorial optimization
Approximation algorithms
Greedy algorithms
Submodularity

ABSTRACT

We consider the problem of maximizing a monotone nondecreasing set function under multiple constraints, where the constraints are also characterized by monotone nondecreasing set functions. We propose two greedy algorithms to solve the problem with provable approximation guarantees. The first algorithm exploits the structure of a special class of the general problem instance to obtain a better time complexity. The second algorithm is suitable for the general problem. We characterize the approximation guarantees of the two algorithms, leveraging the notions of submodularity ratio and curvature introduced for set functions. We then discuss particular applications of the general problem formulation to problems that have been considered in the literature. We validate our theoretical results using numerical examples.

1. Introduction

We study the problem of maximizing a set function over a ground set $\mathcal{S}$ in the presence of n constraints, where the constraints are also characterized by set functions. Specifically, given monotone nondecreasing set functions[1] $f : 2^{\mathcal{S}} \to \mathbb{R}_{\geq 0}$ and $h_i : 2^{\mathcal{S}} \to \mathbb{R}_{\geq 0}$ for all $i \in [n] \triangleq \{1, \dots, n\}$ with $n \in \mathbb{Z}_{\geq 1}$, we consider the following constrained optimization problem:

$$\begin{aligned} &\max_{\mathcal{A}\subseteq\mathcal{S}} f(\mathcal{A}) \\ &s.t.\ h_i(\mathcal{A}\cap\mathcal{S}_i) \leq H_i,\ \forall i \in [n], \end{aligned} \tag{P1}$$

where $\mathcal{S} = \cup_{i\in[n]}\mathcal{S}_i$, with $\mathcal{S}_i \subseteq \mathcal{S}$ for all $i \in [n]$, and $H_i \in \mathbb{R}_{\geq 0}$ for all $i \in [n]$. By simultaneously allowing nonsubmodular set functions[2] (in both the objective and the constraints) and multiple constraints (given by upper bounds on the set functions), (P1) generalizes a number of combinatorial optimization problems (e.g., Bian, Buhmann, Krause, and Tschiatschek (2017), Das and Kempe (2018), Iyer and Bilmes (2012), Khuller, Moss, and Naor (1999), Kulik, Shachnai, and Tamir (2009) and Leskovec et al. (2007)). Instances of (P1) arise in many important applications, including sensor (or measurement) selection for state (or parameter) estimation (e.g., Jawaid and Smith (2015), Ye, Paré and Sundaram (2021) and Zhang, Ayoub, and Sundaram (2017)), experimental design (e.g., Bian et al. (2017) and Krause, Singh, and Guestrin (2008)), and data subset (or client) selection for machine learning (e.g., Das and Kempe (2011), Durga, Iyer, Ramakrishnan, and De (2021) and Ye and Gupta (2021)). As an example, the problem of sensor selection for minimizing the error covariance of the Kalman filter (studied, e.g., in Jawaid and Smith (2015) and Ye, Woodford, Roy and Sundaram (2021)) can be viewed as a special case of (P1) where the objective function $f(\cdot)$ is defined on the ground set $\mathcal{S}$ that contains all candidate sensors, and is used to characterize the state estimation performance of the Kalman filter using measurements from an allowed set $\mathcal{A} \subseteq \mathcal{S}$ of selected sensors. The constraints corresponding to $h_i(\cdot)$ for all

☆ The work of the first three authors was supported in part by the National Natural Science Foundation of China under grants 62203179, 62222205, 61973133 and 61972170, and the Hubei Province National Natural Science Foundation under grants 2022CFB670, 2021CFB343 and 2022CFA052. The work of Vijay Gupta was supported in part by NSF under grant 2300355. The material in this paper was partially presented at the 60th IEEE Conference on Decision and Control, December 13-15, 2021, Austin, Texas, USA. This paper was recommended for publication in revised form by Associate Editor Kok Lay Teo under the direction of Editor Ian R Petersen.

* Corresponding author at: School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China.
** Corresponding author at: Hubei Key Laboratory of Brain-inspired Intelligent Systems, Wuhan 430074, China.
E-mail addresses: yelintao93@hust.edu.cn (L. Ye), zwliu@hust.edu.cn (Z.-W. Liu), chiming@hust.edu.cn (M. Chi), gupta869@purdue.edu (V. Gupta).

[1] A set function $f : 2^{\mathcal{S}} \to \mathbb{R}_{\geq 0}$ is monotone nondecreasing if $f(\mathcal{A}) \leq f(\mathcal{B})$ for all $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{S}$.

[2] A set function $f : 2^{\mathcal{S}} \to \mathbb{R}_{\geq 0}$ is submodular if and only if $f(\mathcal{A}\cup\{v\})-f(\mathcal{A}) \geq f(\mathcal{B}\cup\{v\})-f(\mathcal{B})$ for all $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{S}$ and for all $v \in \mathcal{S} \setminus \mathcal{B}$.

https://doi.org/10.1016/j.automatica.2023.111126
0005-1098/

$i \in [n]$ represent, e.g., budget, communication, spatial and energy constraints on the set of selected sensors (e.g., Mo, Ambrosino, and Sinopoli (2011), Prasad, Hudack, Mou, and Sundaram (2022) and Ye, Woodford et al. (2021)). We apply our results to two specific problems in Section 4.

In general, (P1) is NP-hard (e.g., Feige (1998)), i.e., obtaining an optimal solution to (P1) is computationally expensive. For instances of (P1) with a monotone nondecreasing submodular objective function $f(\cdot)$ and a single constraint, there is a long line of work for showing that greedy algorithms yield constant-factor approximation ratios for (P1) (e.g., Calinescu, Chekuri, Pal, and Vondrák (2011), Khuller et al. (1999) and Nemhauser, Wolsey, and Fisher (1978)). However, many important applications that can be captured by the general problem formulation in (P1) do not feature objective functions that are submodular (see, e.g., Das and Kempe (2011), Elenberg, Khanna, Dimakis, and Negahban (2018), Krause et al. (2008), Ye, Woodford et al. (2021) and Zhang et al. (2017)). For instances of (P1) with a nonsubmodular objective function $f(\cdot)$, it has been shown that the greedy algorithms yield approximation ratios that depend on the problem parameters (e.g., Bian et al. (2017), Das and Kempe (2011) and Tzoumas, Carlone, Pappas, and Jadbabaie (2020)). As an example, the approximation ratio of the greedy algorithm provided in Bian et al. (2017) depends on the submodularity ratio and the curvature of the objective function in (P1).

Moreover, most of the existing works consider instances of (P1) with a single constraint on the set of the selected elements, e.g., a cardinality, budget, or a matroid constraint. The objective of this paper is to relax this requirement of a single simple constraint being present on the set of selected elements. For instance, in the Kalman filtering based sensor scheduling (or selection) problem described above, a natural formulation is to impose a separate constraint on the set of sensors selected at different time steps, or to consider multiple constraints such as communication and budget constraints on how many sensors can work together simultaneously. Thus, we consider the problem formulation (P1), where the objective function is a monotone nondecreasing set function and the constraints are also characterized by monotone nondecreasing set functions. We do not assume that the objective function and the functions in the constraints are necessarily submodular. We propose approximation algorithms to solve (P1), and provide theoretical approximation guarantees for the proposed algorithms leveraging the notions of curvature and submodularity ratio.

Our main contributions are summarized as follows. First, we consider instances of (P1) with $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$ for all $i, j \in [n]$ $(i \neq j)$, and propose a parallel greedy algorithm with time complexity $O((\max_{i\in[n]} |\mathcal{S}_i|)^2)$ that runs for each $i \in [n]$ in parallel. We characterize the approximation guarantee of the parallel greedy algorithm, leveraging the submodularity and curvature of the set functions in the instances of (P1). Next, we consider general instances of (P1) without utilizing the assumption on mutually exclusive sets S_i. We propose a greedy algorithm with time complexity $O(n|\mathcal{S}|^2)$, and characterize its approximation guarantee. The approximation guarantee of this algorithm again depends on the submodularity ratio and curvature of the set functions and the solution returned by the algorithm. Third, we specialize these results to some example applications and evaluate these approximation guarantees by bounding the submodularity ratio and curvature of the set functions. Finally, we validate our theoretical results using numerical examples; the results show that the two greedy algorithms yield comparable performances that are reasonably good in practice. A preliminary version of this paper was presented in Ye and Gupta (2021), where only the parallel greedy algorithm was studied for a special instance of (P1).

Notation For a matrix $P \in \mathbb{R}^{n\times n}$, let $P^\top$, $\mathrm{Tr}(P)$, $\lambda_1(P)$ and $\lambda_n(P)$ be its transpose, trace, an eigenvalue with the largest magnitude, and an eigenvalue with the smallest magnitude, respectively. A positive definite matrix $P \in \mathbb{R}^{n\times n}$ is denoted by $P \succ 0$. Let I_n denote an $n \times n$ identity matrix. For a vector x, let x_i (or $(x)_i$) be the ith element of x, and define $\mathrm{supp}(x) = \{i : x_i \neq 0\}$. The Euclidean norm of x is denoted by $\|x\|$. Given two functions $\varphi_1 : \mathbb{R}_{\geq 0} \to \mathbb{R}$ and $\varphi_2 : \mathbb{R}_{\geq 0} \to \mathbb{R}$, $\varphi_1(n)$ is $O(\varphi_2(n))$ if there exist positive constants c and N such that $|\varphi_1(n)| \leq c|\varphi_2(n)|$ for all $n \geq N$.

2. Preliminaries

We begin with some definitions (see, e.g., Bian et al. (2017), Conforti and Cornuéjols (1984), Kuhnle, Smith, Crawford, and Thai (2018) and Nemhauser et al. (1978)).

Definition 1. The submodularity ratio of $h : 2^{\mathcal{S}} \to \mathbb{R}_{\geq 0}$ is the largest $\gamma \in \mathbb{R}$ such that

$$\sum_{v\in\mathcal{A}\setminus\mathcal{B}} \big(h(\{v\}\cup\mathcal{B}) - h(\mathcal{B})\big) \geq \gamma\big(h(\mathcal{A}\cup\mathcal{B}) - h(\mathcal{B})\big), \tag{1}$$

for all $A, B \subseteq \mathcal{S}$. The diminishing return (DR) ratio of $h(\cdot)$ is the largest $\kappa \in \mathbb{R}$ such that

$$h(\mathcal{A}\cup\{v\}) - h(\mathcal{A}) \geq \kappa\big(h(\mathcal{B}\cup\{v\}) - h(\mathcal{B})\big), \tag{2}$$

for all $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{S}$ and for all $v \in \mathcal{S}\setminus\mathcal{B}$.

Definition 2. The curvature of $h : 2^{\mathcal{S}} \to \mathbb{R}_{\geq 0}$ is the smallest $\alpha \in \mathbb{R}$ that satisfies

$$h(\mathcal{A}\cup\{v\}) - h(\mathcal{A}) \geq (1-\alpha)\big(h(\mathcal{B}\cup\{v\}) - h(\mathcal{B})\big), \tag{3}$$

for all $\mathcal{B} \subseteq \mathcal{A} \subseteq \mathcal{S}$ and for all $v \in \mathcal{S}\setminus\mathcal{A}$. The extended curvature of $h(\cdot)$ is the smallest $\tilde{\alpha} \in \mathbb{R}$ that satisfies

$$h(\mathcal{A}\cup\{v\}) - h(\mathcal{A}) \geq (1-\tilde{\alpha})\big(h(\mathcal{B}\cup\{v\}) - h(\mathcal{B})\big), \tag{4}$$

for all $\mathcal{A}, \mathcal{B} \subseteq \mathcal{S}$ and for all $v \in (\mathcal{S}\setminus\mathcal{A}) \cap (\mathcal{S}\setminus\mathcal{B})$.

For any monotone nondecreasing set function $h : 2^{\mathcal{S}} \to \mathbb{R}_{\geq 0}$, one can check that the submodularity ratio γ, the DR ratio κ, the curvature α and the extended curvature $\tilde{\alpha}$ of $h(\cdot)$ satisfy that $\gamma, \kappa, \alpha, \tilde{\alpha} \in [0, 1]$. Moreover, we see from Definition 2 that $\tilde{\alpha} \geq \alpha$, and it can also be shown that $\gamma \geq \kappa$ (e.g., Bian et al. (2017), Kuhnle et al. (2018) and Ye and Gupta (2021)). Further assuming that $h(\cdot)$ is submodular, one can show that $\gamma = \kappa = 1$ (e.g., Bian et al. (2017) and Kuhnle et al. (2018)). For a modular set function $h : 2^{\mathcal{S}} \to \mathbb{R}_{\geq 0}$,[3] we see from Definition 2 that the curvature and the extended curvature of $h(\cdot)$ satisfy that $\alpha = \tilde{\alpha} = 0$. Thus, the submodularity (resp., DR) ratio of a monotone nondecreasing set function $h(\cdot)$ characterizes the approximate submodularity (resp., approximate DR property) of $h(\cdot)$. The curvatures of $h(\cdot)$ characterize how far the function $h(\cdot)$ is from being modular. Before we proceed, we note that the set $\mathcal{S}_i$ in (P1) can potentially intersect with $\mathcal{S}_j$, for any $i, j \in \mathcal{S}$ with $i \neq j$. Moreover, one can show that a cardinality constraint, a (partitioned) matroid constraint, or multiple budget constraints on the set $\mathcal{A}$ of selected elements are special cases of the constraints in (P1). In particular, $h_i(\cdot)$ in (P1) reduces to a modular set function for any $i \in [n]$ when considering budget constraints.

[3] A set function $h : 2^{\mathcal{S}} \to \mathbb{R}_{\geq 0}$ is modular if and only if $h(\mathcal{A}) = \sum_{v\in\mathcal{A}} h(v)$ for all $\mathcal{A} \subseteq \mathcal{S}$.

Algorithm 1 Parallel greedy algorithm

Input: $\mathcal{S} = \cup_{i\in[n]}\mathcal{S}_i$, $f : 2^{\mathcal{S}} \to \mathbb{R}_{\geq 0}$ and $h_i : 2^{\mathcal{S}} \to \mathbb{R}_{\geq 0}$ $\forall i \in [n]$, $H_i \in \mathbb{R}_{\geq 0}$ $\forall i \in [n]$

1: **for** each $i \in [n]$ in parallel **do**
2: $\quad\mathcal{W}_i \leftarrow \mathcal{S}_i$, $\mathcal{A}_i^r \leftarrow \emptyset$
3: $\quad\mathcal{B}_i^r \leftarrow \arg\max_{v\in\mathcal{S}_i} f(v)$
4: $\quad$**while** $\mathcal{W}_i \neq \emptyset$ **do**
5: $\quad\quad v^\star \leftarrow \arg\max_{v\in\mathcal{W}_i} \frac{\delta_v(\mathcal{A}_i^r)}{\delta_v^i(\mathcal{A}_i^r)}$
6: $\quad\quad$**if** $h_i(\mathcal{A}_i^r \cup \{v^\star\}) \leq H_i$ **then**
7: $\quad\quad\quad\mathcal{A}_i^r \leftarrow \mathcal{A}_i^r \cup \{v^\star\}$
8: $\quad\quad\mathcal{W}_i \leftarrow \mathcal{W}_i \setminus \{v^\star\}$
9: $\quad\mathcal{A}_i^r \leftarrow \arg\max_{\mathcal{A}\in\{\mathcal{A}_i^r,\mathcal{B}_i^r\}} f(\mathcal{A})$
10: $\mathcal{A}^r \leftarrow \cup_{i\in[n]}\mathcal{A}_i^r$
11: **Return** $\mathcal{A}^r$

3. Approximation algorithms

We make the following standing assumption.

Assumption 3. The set functions $f(\cdot)$ and $h_i(\cdot)$ satisfy that $f(\emptyset) = 0$ and $h_i(\emptyset) = 0$ for all $i \in [n]$. Further, $h_i(v) > 0$ for all $i \in [n]$ and for all $v \in \mathcal{S}$.

Note that (P1) is NP-hard, and cannot be approximated within any constant factor independent of any problem parameter (if P≠NP), even when the constraints in (P1) reduce to a cardinality constraint $|\mathcal{A}| \leq H$ (Ye, Woodford et al., 2021). Thus, we aim to provide approximation algorithms for (P1) and characterize the corresponding approximation guarantees in terms of the problem parameters. To simplify the notation in the sequel, for any $\mathcal{A}, \mathcal{B} \subseteq \mathcal{S}$, we denote

$$\begin{aligned} &\delta_{\mathcal{B}}(\mathcal{A}) = f(\mathcal{A}\cup\mathcal{B}) - f(\mathcal{A}) \\ &\delta_{\mathcal{B}}^i(\mathcal{A}) = h_i(\mathcal{A}\cup\mathcal{B}\cap\mathcal{S}_i) - h_i(\mathcal{A}\cap\mathcal{S}_i). \end{aligned} \tag{5}$$

Thus, $\delta_{\mathcal{B}}(\mathcal{A})$ (resp., $\delta_{\mathcal{B}}^i(\mathcal{A})$) is the marginal return of $f(\cdot)$ (resp., $h_i(\cdot)$) when adding $\mathcal{B}$ to $\mathcal{A}$.

3.1. Parallel greedy algorithm for a special case

We rely on the following assumption and introduce a parallel greedy algorithm (Algorithm 1) for (P1).

Assumption 4. The ground set $\mathcal{S} = \cup_{i\in[n]}\mathcal{S}_i$ in (P1) satisfies that $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$ for all $i, j \in [n]$ with $i \neq j$.

For each $i \in [n]$ in parallel, Algorithm 1 first sets $\mathcal{S}_i$ to be the ground set $\mathcal{W}_i$ for the algorithm, and then iterates over the current elements in $\mathcal{W}_i$ in the while loop. In particular, the algorithm greedily chooses an element $v \in \mathcal{W}_i$ in line 5 that maximizes the ratio between the marginal returns $\delta_v(\mathcal{A}_i^r)$ and $\delta_v^i(\mathcal{A}_i^r)$ for all $v \in \mathcal{W}_i$. The overall greedy solution is given by $\mathcal{A}^r = \cup_{i\in[n]}\mathcal{A}_i^r$. Thus, one may view Algorithm 1 as solving the problem $\max_{\mathcal{A}\subseteq\mathcal{S}_i} f(\mathcal{A})$ s.t. $h_i(\mathcal{A}) \leq H_i$ for each $i \in [n]$ separately in parallel, and then merge the obtained solutions. Note that the overall time complexity of Algorithm 1 is $O((\max_{i\in[n]}|\mathcal{S}_i|)^2)$. To provide a guarantee on the quality of the approximation for the solution returned by Algorithm 1, we start with the following observation, which follows directly from the definition of the algorithm.

Observation 1. *For any $i \in [n]$ in Algorithm 1, denote $\mathcal{A}_i^r = \{q_1, \ldots, q_{|\mathcal{A}_i^r|}\}$ and $\mathcal{A}_{i,j}^r = \{q_1, \ldots, q_j\}$ for all $j \in [|\mathcal{A}_i^r|]$ with $\mathcal{A}_{i,0}^r = \emptyset$. Then, there exists $l_i \in [|\mathcal{A}_i^r|]$ such that (1) $q_k \in \arg\max_{v\in\mathcal{W}_i} \frac{\delta_v(\mathcal{A}_{i,k-1}^r)}{\delta_v^i(\mathcal{A}_{i,k-1}^r)}$ and $h_i(\mathcal{A}_{i,k}^r) \leq H_i$ for all $k \in [l_i]$; and (2) $h_i(\mathcal{A}_{i,l_i}^r \cup \{v_{l_i+1}^\star\}) > H_i$, where $v_{l_i+1}^\star \in \arg\max_{v\in\mathcal{W}_i} \frac{\delta_v(\mathcal{A}_{i,l_i}^r)}{\delta_v^i(\mathcal{A}_{i,l_i}^r)}$.*

Definition 5. The greedy submodularity ratio of $f : 2^{\mathcal{S}} \to \mathbb{R}_{\geq 0}$ in (P1) is the largest $\tilde{\gamma}_f \in \mathbb{R}$ that satisfies $f(\mathcal{B}_i^r) \geq \tilde{\gamma}_f\big(f(\mathcal{A}_{i,l_i}^r \cup \{v_{l_i+1}^\star\}) - f(\mathcal{A}_{i,l_i}^r)\big)$ for all $i \in [n]$, where $v_{l_i+1}^\star, \mathcal{A}_{i,l_i}^r$ are given in Observation 1, and $\mathcal{B}_i^r$ is given by line 3 of Algorithm 1.

For monotone nondecreasing $f(\cdot)$ in (P1), one can check that the greedy submodularity ratio of $f(\cdot)$ satisfies $\tilde{\gamma}_f \in \mathbb{R}_{\geq 0}$. Further assuming that $f(\cdot)$ is submodular, one can show that $\tilde{\gamma}_f \in \mathbb{R}_{\geq 1}$.

Theorem 6. *Suppose that Assumption 4 holds. Let $\mathcal{A}^r$ and $\mathcal{A}^\star$ be the solution to (P1) returned by Algorithm 1 and an optimal solution to (P1), respectively. Then,*

$$\begin{aligned} f(\mathcal{A}^r) \geq\ & \frac{(1-\alpha_f)\kappa_f \min\{1, \tilde{\gamma}_f\}}{2} \\ &\times \min_{i\in[n]}(1 - e^{-(1-\tilde{\alpha}_i)\gamma_f}) f(\mathcal{A}^\star), \end{aligned} \tag{6}$$

where $\alpha_f, \kappa_f, \gamma_f \in [0, 1]$ and $\tilde{\gamma}_f \in \mathbb{R}_{\geq 0}$ are the curvature, DR ratio, submodularity ratio and greedy submodularity ratio of $f(\cdot)$, respectively, and $\tilde{\alpha}_i \in [0, 1]$ is the extended curvature of $h_i(\cdot)$ for all $i \in [n]$.

We briefly explain the ideas for the proof of Theorem 6; a detailed proof is included in Appendix A. Supposing that $h_i(\cdot)$ is modular for any $i \in [n]$, the choice $v^\star$ in line 5 of the algorithm reduces to $v^\star \leftarrow \arg\max_{v\in\mathcal{W}_i} \frac{\delta_v(\mathcal{A}_i^r)}{h_i(v)}$, which is an element $v \in \mathcal{W}_i$ that maximizes the marginal return of the objective function $f(\cdot)$ per unit cost incurred by $h_i(\cdot)$ when adding v to the current greedy solution $\mathcal{A}_i^r$. This renders the greedy nature of the choice $v^\star$. To leverage this greedy choice property when $h_i(\cdot)$ is not modular, we use the (extended) curvature of $h_i(\cdot)$ (i.e., $\tilde{\alpha}_i$) to measure how close $h_i(\cdot)$ is to being modular. Moreover, we use $\gamma_f, \tilde{\gamma}_f$ to characterize the approximate submodularity of $f(\cdot)$. Note that the multiplicative factor $(1-\alpha_f)\kappa_f$ in (6) results from merging $\mathcal{A}_i^r$ for all $i \in [n]$ into $\mathcal{A}^r$ in line 10 of the algorithm.

Remark 7. Under the stronger assumption that the objective function $f(\cdot)$ in (P1) can be written as $f(\mathcal{A}) = \sum_{i\in[n]} f(\mathcal{A}\cap\mathcal{S}_i)$, similar arguments to those in the proof of Theorem 6 can be used to show that $f(\mathcal{A}^r) \geq \frac{\min\{1,\tilde{\gamma}_f\}}{2} \min_{i\in[n]}(1 - e^{-(1-\tilde{\alpha}_i)\gamma_f}) f(\mathcal{A}^\star)$.

3.2. A greedy algorithm for the general case

We now introduce a greedy algorithm (Algorithm 2) for general instances of (P1), where we define the feasible set associated with the constraints in (P1) as $\mathcal{F} = \big\{\mathcal{A} \subseteq \mathcal{S} : h_i(\mathcal{A}\cap\mathcal{S}_i) \leq H_i, \forall i \in [n]\big\}$. In the absence of Assumption 4, Algorithm 2 let $\mathcal{S} = \cup_{i\in[n]}\mathcal{S}_i$ be the ground set $\mathcal{W}$ in the algorithm. Algorithm 2 then iterates over the current elements in $\mathcal{W}$, and greedily chooses $v \in \mathcal{W}$ and $i \in [n]$ in line 3 such that the ratio between the marginal returns $\delta_v(\mathcal{A}^g)$ and $\delta_v^i(\mathcal{A}^g)$ are maximized for all $v \in \mathcal{W}$ and for all $i \in [n]$. The element $v^\star$ will be added to $\mathcal{A}^r$ if the constraint $h_i(\mathcal{A}^g \cup \{v^\star\}) \leq H_i$ is not violated for any $i \in [n]$. Note that different from line 5 in Algorithm 1, the maximization in line 3 of Algorithm 2 is also taken with respect to $i \in [n]$. This is because we do not consider the set $\mathcal{S}_i$ and the constraint associated with $h_i(\cdot)$ for each $i \in [n]$ separately in Algorithm 2. One can check that the time complexity of Algorithm 2 is $O(n|\mathcal{S}|^2)$. In order to characterize the approximation guarantee of Algorithm 2, we introduce the following definition.

Algorithm 2 Greedy algorithm for general instances of (P1)

Input: $\mathcal{S} = \cup_{i\in[n]}\mathcal{S}_i$, $f : 2^{\mathcal{S}} \to \mathbb{R}_{\geq 0}$ and $h_i : 2^{\mathcal{S}} \to \mathbb{R}_{\geq 0}$ $\forall i \in [n]$, $H_i \in \mathbb{R}_{\geq 0}$ $\forall i \in [n]$

1: $\mathcal{W} \leftarrow \mathcal{S}$, $\mathcal{A}^g \leftarrow \emptyset$
2: **while** $\mathcal{W} \neq \emptyset$ **do**
3: $\quad (v^\star, i^\star) \leftarrow \arg\max_{(v\in\mathcal{W}, i\in[n])} \frac{\delta_v(\mathcal{A}^g)}{\delta_v^i(\mathcal{A}^g)}$
4: $\quad$ **if** $(\mathcal{A}^g \cup \{v^\star\}) \in \mathcal{F}$ **then**
5: $\quad\quad \mathcal{A}^g \leftarrow \mathcal{A}^g \cup \{v^\star\}$
6: $\quad \mathcal{W} \leftarrow \mathcal{W} \setminus \{v^\star\}$
7: **Return** $\mathcal{A}^g$

Definition 8. Let $\mathcal{A}^g = \{q_1, \dots, q_{|\mathcal{A}^g|}\}$ and $\mathcal{A}^\star$ be the solution to (P1) returned by Algorithm 2 and an optimal solution to (P1), respectively. Denote $\mathcal{A}_j^g = \{q_1, \dots, q_j\}$ for all $j \in [|\mathcal{A}^g|]$ with $\mathcal{A}_0^g = \emptyset$. For any $j \in \{0, \dots, |\mathcal{A}^g|-1\}$, the jth greedy choice ratio of Algorithm 2 is the largest $\psi_j \in \mathbb{R}$ that satisfies $\frac{\delta_{q_{j+1}}(\mathcal{A}_j^g)}{\delta_{q_{j+1}}^{i_j}(\mathcal{A}_j^g)} \geq \psi_j \frac{\delta_v(\mathcal{A}_j^g)}{\delta_v^i(\mathcal{A}_j^g)}$ for all $v \in \mathcal{A}^\star \setminus \mathcal{A}_j^g$ and for all $i \in [n]$, where $i_j \in [n]$ is the index of the constraint in (P1) that corresponds to q_{j+1} given by line 3 of Algorithm 2.

Note that $v^\star$ chosen in line 3 of Algorithm 2 is not added to the greedy solution $\mathcal{A}_g$ if the constraint in line 4 is violated. Thus, the greedy choice ratio ψ_j given in Definition 8 is used to characterize the suboptimality of $q_{j+1} \in \mathcal{A}_g$ in terms of the maximization over $v \in \mathcal{W}$ and $i \in [n]$ in line 3 of the algorithm.[4] Since both $f(\cdot)$ and $h_i(\cdot)$ for all $i \in [n]$ are monotone nondecreasing functions, Definition 8 implies that $\psi_j \in \mathbb{R}_{\geq 0}$ for all $j \in \{0, \dots, |\mathcal{A}^g| - 1\}$. We also note that for any $j \in \{0, \dots, |\mathcal{A}^g| - 1\}$, a lower bound on ψ_j may be obtained by considering all $v \in \mathcal{S} \setminus \mathcal{A}_j^g$ (instead of $v \in \mathcal{A}^\star \setminus \mathcal{A}_j^g$) in Definition 8. Such lower bounds on ψ_j for all $j \in \{0, \dots, |\mathcal{A}^g|-1\}$ can be computed in $O(n|\mathcal{S}|^2)$ time and in parallel to Algorithm 2. The approximation guarantee for Algorithm 2 is provided in the following result proven in Appendix B.

Theorem 9. *Let $\mathcal{A}^g$ and $\mathcal{A}^\star$ be the solution to (P1) returned by Algorithm 2 and an optimal solution to (P1), respectively. Then,*

$$\begin{aligned} f(\mathcal{A}^g) &\geq \Big(1 - (1 - \frac{B}{|\mathcal{A}^g|})^{|\mathcal{A}^g|}\Big) f(\mathcal{A}^\star) \\ &\geq (1 - e^{-B}) f(\mathcal{A}^\star) \end{aligned} \tag{7}$$

with $B \triangleq \frac{(1-\alpha_h)\gamma_f}{\sum_{i\in[n]} H_i} \sum_{j=0}^{|\mathcal{A}^g|-1} \psi_j \delta_{q_{j+1}}^{i_j}(\mathcal{A}_j^g)$, where $\gamma_f \in [0, 1]$ is the submodularity ratio of $f(\cdot)$, $\alpha_h \triangleq \min_{i\in[n]} \tilde{\alpha}_i$ with $\tilde{\alpha}_i \in [0, 1]$ to be the extended curvature of $h_i(\cdot)$ for all $i \in [n]$, $\psi_j \in \mathbb{R}_{\geq 0}$ is the jth greedy choice ratio of Algorithm 2 for all $j \in \{0, \dots, |\mathcal{A}^g| - 1\}$, and $i_j \in [n]$ is the index of the constraint in (P1) that corresponds to q_{j+1} given by line 3 of Algorithm 2.

Similarly to Theorem 6, we leverage the greedy choice property corresponding to line 3 of Algorithm 2 and the properties of $f(\cdot)$ and $h_i(\cdot)$. However, since Algorithm 2 considers the constraints associated with $h_i(\cdot)$ for all $i \in [n]$ simultaneously, the proof of Theorem 9 requires more care, and the approximation guarantee in (7) does not contain the multiplicative factor $(1 - \alpha_f)\kappa_f$.

Remark 10. Similarly to Observation 1, there exists the maximum $l \in [|\mathcal{A}^g|]$ such that for any $q_j \in \mathcal{A}_l^g$, q_j does not violate the condition in line 4 of Algorithm 2 when adding to the greedy solution $\mathcal{A}^g$. One can then show via Definition 8 that $\psi_{j-1} \geq 1$ for all $j \in [|\mathcal{A}_l^g|]$. Further assuming that Assumption 4 holds, one can follow the arguments in the proof of Theorem 9 and show that

$$\begin{aligned} f(\mathcal{A}^g) &\geq \Big(1 - (1 - \frac{\tilde{B}}{|\mathcal{A}_l^g|})^{|\mathcal{A}_l^g|}\Big) f(\mathcal{A}^\star) \\ &\geq (1 - e^{-\tilde{B}}) f(\mathcal{A}^\star), \end{aligned} \tag{8}$$

where $\tilde{B} \triangleq \frac{(1-\alpha_h)\gamma_f}{\sum_{i\in[n]} H_i} \sum_{i\in[n]} h_i(\mathcal{A}_l^g \cap \mathcal{S}_i)$.

3.3. Comparisons to existing results

Theorems 6 and 9 generalize several existing results in the literature. First, consider instances of (P1) with a single budget constraint, i.e., $h(\mathcal{A}) = \sum_{v\in\mathcal{A}} h(v) \leq H$. We see from Definition 2 that the extended curvature of $h(\cdot)$ is 0. It follows from Remark 7 that the approximation guarantee of Algorithm 1 provided in Theorem 6 reduces to $f(\mathcal{A}^r) \geq \frac{\min\{1,\tilde{\gamma}_f\}}{2}(1 - e^{-\gamma_f}) f(\mathcal{A}^\star)$, which matches with the approximation guarantee of the greedy algorithm provided in Ye, Paré et al. (2021). Further assuming that the objective function $f(\cdot)$ in (P1) is submodular, we have from Definitions 1 and 5 that $\gamma_f = 1$ and $\tilde{\gamma}_f \geq 1$, and the approximation guarantee of Algorithm 1 further reduces to $f(\mathcal{A}^r) \geq \frac{1}{2}(1 - e^{-1}) f(\mathcal{A}^\star)$, which matches with the results in Khuller et al. (1999) and Leskovec et al. (2007).

Second, consider instances of (P1) with a single cardinality constraint $h(\mathcal{A}) = |\mathcal{A}| \leq H$. From Definition 2, we obtain that the curvature of $h(\cdot)$ is 0. It follows from Remark 10 that the approximation guarantee of Algorithm 2 provided in Theorem 9 reduces to $f(\mathcal{A}^g) \geq (1 - e^{-\gamma_f}) f(\mathcal{A}^\star)$, which matches with the approximation guarantee of the greedy algorithm provided in Das and Kempe (2018). Further assuming that the objective function $f(\cdot)$ in (P1) is submodular, we see from Definition 1 that the approximation guarantee of Algorithm 2 reduces to $f(\mathcal{A}^g) \geq (1 - e^{-1}) f(\mathcal{A}^\star)$, which matches with the result in Nemhauser et al. (1978).

Third, consider instances of (P1) with a partitioned matroid constraint, i.e., $h_i(\mathcal{A} \cap \mathcal{S}_i) = |\mathcal{A} \cap \mathcal{S}_i| \leq H_i$ $\forall i \in [n]$ and Assumption 4 holds. Definition 2 shows that the curvature of $h_i(\cdot)$ is 0 for all $i \in [n]$. Using similar arguments to those in Remark 10 and the proof of Theorem 9, one can show that Algorithm 2 yields the following approximation guarantee:

$$f(\mathcal{A}^g) \geq \Big(1 - (1 - \frac{\gamma_f}{\sum_{i\in[n]} H_i})^{|\mathcal{A}_l^g|}\Big) f(\mathcal{A}^\star). \tag{9}$$

Further assuming that $f(\cdot)$ in (P1) is submodular, i.e., $\gamma_f = 1$ in (9), one can check that the approximation guarantee in (9) matches with the result in Fisher, Nemhauser, and Wolsey (1978).

4. Specific application settings

We now discuss some specific applications that can be captured by the general problem formulation in (P1). For these applications, we bound the parameters given by Definitions 1–2 and evaluate the resulting approximation guarantees provided in Theorems 6 and 9.

4.1. Sensor selection

Sensor selection problems arise in many different applications, e.g., Chepuri and Leus (2014), Joshi and Boyd (2008), Krause et al. (2008) and Ye, Woodford et al. (2021). A typical scenario is that only a subset of all candidate sensors can be used to estimate the state of a target environment or system. The goal is to select this subset to optimize an estimation performance metric. If the target

[4] The proof of Theorem 9 shows that we only need to consider the case when ψ_j is well-defined.

system is a dynamical system whose state evolves over time, this problem is sometimes called sensor scheduling, in which different sets of sensors can be selected at different time steps (e.g., Jawaid and Smith (2015)) with possibly different constraints on the set of sensors selected at different time steps.

As an example, we can consider the Kalman filtering sensor scheduling (or selection) problem (e.g., Chamon, Pappas, and Ribeiro (2017), Jawaid and Smith (2015), Tzoumas, Jadbabaie, and Pappas (2016) and Ye, Woodford et al. (2021)) for a linear time-varying system

$$\begin{aligned} x_{k+1} &= A_k x_k + w_k \\ y_k &= C_k x_k + v_k, \end{aligned} \tag{10}$$

where $A_k \in \mathbb{R}^{n\times n}$, $C_k \in \mathbb{R}^{m\times n}$, $x_0 \sim \mathcal{N}(0, \Pi_0)$ with $\Pi_0 \succ 0$, w_k, v_k are zero-mean white Gaussian noise processes with $\mathbb{E}[w_k w_k^\top] = W \succ 0$, $\mathbb{E}[v_k v_k^\top] = \text{diag}(\sigma_1^2 \cdots \sigma_m^2)$, for all $k \in \mathbb{Z}_{\geq 0}$ with $\sigma_i > 0$ for all $i \in [m]$, and x_0 is independent of w_k, v_k for all $k \in \mathbb{Z}_{\geq 0}$. If there are multiple sensors present, we can let each row in C_k correspond to a candidate sensor at time step k. Given a target time step $\ell \in \mathbb{Z}_{\geq 0}$, we let $\mathcal{S} = \{(k, i) : i \in [m], k \in \{0, \dots, \ell\}\}$ be the ground set that contains all the candidate sensors at different time steps. Thus, we can write $\mathcal{S} = \cup_{k\in\{0,\dots,\ell\}}\mathcal{S}_k$ with $\mathcal{S}_k = \{(k, i) : i \in [m]\}$, where $\mathcal{S}_k$ (with $|\mathcal{S}_k| = m$) is the set of sensors available at time step k. Now, for any $\mathcal{A}_k \subseteq \mathcal{S}_k$, let $C_{\mathcal{A}_k} \in \mathbb{R}^{|\mathcal{A}_k|\times n}$ be the measurement matrix corresponding to the sensors in $\mathcal{A}_k$, i.e., $C_{\mathcal{A}_k}$ contains rows from C_k that correspond to the sensors in $\mathcal{A}_k$. We then consider the following set function $g : 2^{\mathcal{S}} \to \mathbb{R}_{\geq 0}$:

$$g(\mathcal{A}) = \text{Tr}\Big(\big(P_{\ell,\mathcal{A}}^{-1} + \sum_{v\in\mathcal{A}_\ell} \sigma_v^{-2} C_v^\top C_v\big)^{-1}\Big), \tag{11}$$

where $\mathcal{A} = \cup_{k\in\{0,\dots,\ell\}}\mathcal{A}_k \subseteq \mathcal{S}$ with $\mathcal{A}_k \subseteq \mathcal{S}_k$, and $P_{\ell,\mathcal{A}}$ is given recursively via

$$P_{k+1,\mathcal{A}} = W + A_k\big(P_{k,\mathcal{A}}^{-1} + \sum_{v\in\mathcal{A}_k} \sigma_v^{-2} C_v^\top C_v\big)^{-1} A_k^\top, \tag{12}$$

for $k = \{0, \dots, \ell - 1\}$ with $P_{0,\mathcal{A}} = \Pi_0$. For any $\mathcal{A} \subseteq \mathcal{S}$, $g(\mathcal{A})$ is the mean square estimation error of the Kalman filter for estimating the system state x_ℓ based on the measurements (up until time step ℓ) from the sensors in $\mathcal{A}$ (e.g., Anderson and Moore (1979)). Thus, the sensor selection problem can be cast in the framework of (P1) as:

$$\begin{aligned} &\max_{\mathcal{A}\subseteq\mathcal{S}} \Big\{ f_s(\mathcal{A}) \triangleq g(\emptyset) - g(\mathcal{A}) \Big\} \\ &\text{s.t. } h_k(\mathcal{A} \cap \mathcal{S}_k) \leq H_k, \ \forall k \in \{0, \dots, \ell\}, \end{aligned} \tag{13}$$

where $H_k \in \mathbb{R}_{\geq 0}$ and $h_k(\cdot)$ specify a constraint on the set of sensors scheduled for any time step $k \in \{0, \dots, \ell\}$. By construction, Assumption 4 holds for problem (13).

For the objective function, we have the following result for $f_s(\cdot)$; the proof can be adapted from Huber (2011), Kohara, Okano, Hirata, and Nakamura (2020) and Zhang et al. (2017) and is omitted for conciseness.

Proposition 11. *The set function $f_s : 2^{\mathcal{S}} \to \mathbb{R}_{\geq 0}$ in (13) is monotone nondecreasing with $f_s(\emptyset) = 0$. Moreover, both the submodularity ratio and DR ratio of $f_s(\cdot)$ given in Definition 1 are lower bounded by $\underline{\gamma}$, and both the curvature and extended curvature of $f_s(\cdot)$ given in Definition 2 are upper bounded by $\overline{\alpha}$, where $\underline{\gamma} = \frac{\lambda_n(P_{\ell,\emptyset}^{-1})}{\lambda_1(P_{\ell,\mathcal{S}}^{-1} + \sum_{v\in\mathcal{S}_\ell} \sigma_v^{-2} C_v^\top C_v)} > 0$ and $\overline{\alpha} = 1 - \underline{\gamma}^2 < 1$ with $P_{\ell,\emptyset}$ and $P_{\ell,\mathcal{S}}$ given by Eq. (12).*

Remark 12. Apart from the Kalman filtering sensor selection problem described above, the objective functions in many other formulations such as sensor selection for Gaussian processes (Krause et al., 2008), and sensor selection for hypothesis testing (Ye & Sundaram, 2019) have been shown to be submodular or to have a positive submodularity ratio.

For the constraints in the sensor selection problems (modeled by $h_k(\cdot)$ in (13)), popular choices include a cardinality constraint (Krause et al., 2008) or a budget constraint (Mo et al., 2011; Tzoumas et al., 2020) on the set of selected sensors. Our framework can consider such choices individually or simultaneously. More importantly, our framework is general enough to include other relevant constraints. As an example, suppose that the sensors transmit their local information to a fusion center via a (shared) communication channel. Since the fusion center needs to receive the sensor information before the system propagates to the next time step, there are constraints on the communication latency associated with the selected sensors. To ease our presentation, let us consider a specific time step $k \in \{0, \dots, \ell\}$ for the system given by (10). Let $[m]$ and $\mathcal{A} \subseteq [m]$ be the set of all the candidate sensors at time step k and the set of sensors selected for time step k, respectively. Assume that the sensors in $\mathcal{A}$ transmit the local information to the fusion center using the communication channel in a sequential manner; such an assumption is not restrictive as argued in, e.g., Dinh et al. (2020). For any $v \in [m]$, we let $t_v \in \mathbb{R}_{\geq 0}$ be the transmission latency corresponding to sensor v when using the communication channel, and let $c_v \in \mathbb{R}_{\geq 0}$ be the sensing and computation latency corresponding to sensor v. We assume that t_v, c_v are given at the beginning of time step k (e.g., Shi, Zhou, Niu, Jiang, and Geng (2020)). The following assumption says that the sensing and computation latency cannot dominate the transmission latency.

Assumption 13. For any $v \in [m]$, there exists $r_v \in \mathbb{R}_{>0}$ such that $c_v + t_v - c_u \geq r_v$ for all $u \in [m]$ with $c_u \geq c_v$.

Note that given a set $\mathcal{A} \subseteq [m]$, the total latency (i.e., the computation and transmission latency) depends on the order in which the sensors in $\mathcal{A}$ transmit. Denote an ordering of the elements in $\mathcal{A}$ as $\hat{\mathcal{A}} = \langle a_1, \dots, a_{|\mathcal{A}|}\rangle$. Define $h_c^s : \hat{\mathcal{S}} \to \mathbb{R}_{\geq 0}$ to be a function that maps a sequence of sensors to the corresponding total latency, where $\hat{\mathcal{S}}$ is the set that contains all possible sequences of sensors chosen from the set $[m]$. We know from Ye and Gupta (2021) that the total latency corresponding to $\hat{\mathcal{A}}$ can computed as

$$h_c^s(\hat{\mathcal{A}}_j) = \begin{cases} h_c^s(\hat{\mathcal{A}}_{j-1}) + t_{a_j} & \text{if } c_{a_j} < h_c^s(\hat{\mathcal{A}}_{j-1}), \\ c_{a_j} + t_{a_j} & \text{if } c_{a_j} \geq h_c^s(\hat{\mathcal{A}}_{j-1}), \end{cases} \tag{14}$$

where $\hat{\mathcal{A}}_j = \langle a_1, \dots, a_j\rangle$ for all $j \in [|\mathcal{A}|]$, with $\hat{\mathcal{A}}_0 = \emptyset$ and $h_c^s(\emptyset) = 0$.

We further define a set function $h_c : 2^{[m]} \to \mathbb{R}_{\geq 0}$ such that for any $\mathcal{A} \subseteq [m]$,

$$h_c(\mathcal{A}) = h_c^s(\langle a_1, \dots, a_{|\mathcal{A}|}\rangle), \tag{15}$$

where $\langle a_1, \dots, a_{|\mathcal{A}|}\rangle$ orders the elements in $\mathcal{A}$ such that $c_{a_1} \leq \cdots \leq c_{a_{|\mathcal{A}|}}$. We may now enforce $h_c(\mathcal{A}) \leq H$, where $H \in \mathbb{R}_{\geq 0}$. Thus, for any $\mathcal{A} \subseteq [m]$, we let the sensors in $\mathcal{A}$ transmit the local information in the order given by (15) and require the corresponding total latency to be no greater than H.[5] The following result justifies the way $h_c(\cdot)$ orders the selected sensors, and characterizes the curvature of $h_c(\cdot)$.

Proposition 14. *Consider any $\mathcal{A} \subseteq [m]$ and let $\hat{\mathcal{A}}$ be an arbitrary ordering of the elements in $\mathcal{A}$. Then, $h_c(\mathcal{A}) \leq h_c^s(\hat{\mathcal{A}})$, where $h(\cdot)$ and $h_c^s(\cdot)$ are defined in (14)* and *(15), respectively. Under Assumption 13, it holds that $h_c(\cdot)$ is monotone nondecreasing and that $\tilde{\alpha}_{h_c} \leq \tilde{\alpha}'_{h_c} \triangleq 1 - \min_{v\in[m]} \frac{r_v}{t_v}$, where $\tilde{\alpha}_{h_c} \in [0, 1]$ is the extended curvature of $h_c(\cdot)$, $r_v \in \mathbb{R}_{>0}$ is given in Assumption 13 and $t_v \in R_{\geq 0}$ is the transmission latency corresponding to sensor v.*

[5] A similar constraint can be enforced for each time step $k \in \{0, \dots, \ell\}$.

Proof. First, for any $\mathcal{A} \subseteq [m]$, denote an arbitrary ordering of the elements in $\mathcal{A}$ as $\hat{\mathcal{A}} = \langle a_1, \dots, a_{|\mathcal{A}|}\rangle$. Then, there exists $\tau \in \mathbb{Z}_{\geq 1}$ such that $c_{a_1} \leq \cdots \leq c_{a_\tau}$ and $c_{a_{\tau+1}} \geq c_{a_\tau}$, where $c_v \in \mathbb{R}_{\geq 0}$ is the computation latency corresponding to sensor $v \in [m]$. Moreover, using the definition of $h_c^s(\cdot)$ in (14), one can show that $h_c^s(\cdot)$ satisfies the expression $h_c^s(\hat{\mathcal{A}}) = V_c(\hat{\mathcal{A}}) + \sum_{j=1}^{|\mathcal{A}|} t_j$, where $V_c(\hat{\mathcal{A}}) \in \mathbb{R}_{\geq 0}$ is a function of $\hat{\mathcal{A}}$ that characterizes the time during which the communication channel (shared by all the sensors in $\mathcal{A}$) is idle. Switching the order of a_τ and $a_{\tau+1}$, one can further show that $V_c(\langle a_1, \dots, a_{\tau+1}, a_\tau\rangle) \leq V_c(\langle a_1, \dots, a_\tau, a_{\tau+1}\rangle)$, which implies via the above expression of $h_c^s(\cdot)$ that $h_c^s(\langle a_1, \dots, a_{\tau+1}, a_\tau\rangle) \leq h_c^s(\langle a_1, \dots, a_\tau, a_{\tau+1}\rangle)$. It then follows from (14) that $h_c^s(\langle a_1, \dots, a_{\tau-1}, a_{\tau+1}, a_\tau, \dots, a_{|\mathcal{A}|}\rangle) \leq h_c^s(\hat{\mathcal{A}})$. Repeating the above arguments yields $h_c(\mathcal{A}) \leq h_c^s(\hat{\mathcal{A}})$ for any $\mathcal{A} \subseteq [m]$ and any ordering $\hat{\mathcal{A}}$ of the elements in $\mathcal{A}$.

Next, suppose that Assumption 13 holds. We will show that $h_c(\cdot)$ is monotone nondecreasing and characterize the curvature of $h_c(\cdot)$. To this end, we leverage the expression of $h_c^s(\cdot)$ given above. Specifically, consider any $\mathcal{A} \subseteq [m]$ and let the elements in $\mathcal{A}$ be ordered such that $\hat{\mathcal{A}} = \langle a_1, \dots, a_{|\mathcal{A}|}\rangle$ with $c_{a_1} \leq \cdots \leq c_{a_{|\mathcal{A}|}}$. One can first show that $V_c(\hat{\mathcal{A}}) = 0$. Further considering any $v \in [m] \setminus \mathcal{A}$, one can then show that

$$h_c(\mathcal{A} \cup \{v\}) = \begin{cases} h_c(\mathcal{A}) + t_v & \text{if } c_v \geq c_{a_1}, \\ h_c(\mathcal{A}) - c_{a_1} + c_v + t_v & \text{if } c_v < c_{a_1}. \end{cases}$$

It follows from Assumption 13 that $h_c(\cdot)$ is monotone nondecreasing, and that $h_c(\mathcal{A}) + r_v \leq h_c(\mathcal{A} \cup \{v\}) \leq h_c(\mathcal{A}) + t_v$, for any $\mathcal{A} \subseteq [m]$ and any $v \in [m] \setminus \mathcal{A}$. Recalling Definition 2 completes the proof of the proposition. ■

Recalling (6) (resp., (8)), substituting γ_f, κ_f with $\underline{\gamma}$, substituting α_f with $\overline{\alpha}$ from Proposition 11, and substituting $\tilde{\alpha}_i$ with $\tilde{\alpha}'_{h_c}$ from Proposition 14, one can obtain the approximation guarantee of Algorithm 1 (resp., Algorithm 2) when applied to solve (13).

Remark 15. Many other types of constraints in the sensor selection problem can be captured by (P1). One example is if the selected sensors satisfy certain spatial constraints, e.g., two selected sensors may need to be within a certain distance (e.g., Gupta, Chung, Hassibi, and Murray (2006)); or if a mobile robot collects the measurements from the selected sensors (e.g., Prasad et al. (2022)), the length of the tour of the mobile robot is constrained. Another example is that the budget constraint, where the total cost of the selected sensors is not the sum of the costs of the sensors due to the cost of a sensor can (inversely) depend on the total number of selected sensors (e.g., Iyer and Bilmes (2012)). Such constraints lead to set function constraints on the selected sensors.

4.2. Client selection for distributed optimization

In a typical distributed optimization framework such as Federated Learning (FL), there is a (central) aggregator and a number of edge devices (i.e., clients) (e.g., Li, Huang, Yang, Wang, and Zhang (2020)). Specifically, let $[m]$ be the set of all the candidate clients. For any $v \in [m]$, we assume that the local objective function of client v is given by $F_v(\mathbf{w}) \triangleq \frac{1}{|\mathcal{D}_v|}\sum_{j=1}^{|\mathcal{D}_v|} \ell_j(\mathbf{w}; x_{v,j}, y_{v,j})$, where $\mathcal{D}_v = \{(x_{v,j}, y_{v,j}) : j \in [|\mathcal{D}_v|]\}$ is the local dataset at client v, $\ell_j(\cdot)$ is a loss function, and $\mathbf{w}$ is a model parameter. Here, we let $x_{v,j} \in \mathbb{R}^n$ and $y_{v,j} \in \mathbb{R}$ for all $j \in \mathcal{D}_v$, $\mathbf{w} \in \mathbb{R}^n$, and $F_v(\mathbf{w}) \in \mathbb{R}_{\geq 0}$ for all $\mathbf{w} \in \mathbb{R}^n$. The goal is to solve the following global optimization in a distributed manner:

$$\min_{\mathbf{w}} \Big\{ F(\mathbf{w}) \triangleq \sum_{v \in [m]} \frac{|\mathcal{D}_v|}{D} F_v(\mathbf{w}) \Big\}, \tag{16}$$

where $D = \sum_{v\in[m]} |\mathcal{D}_v|$. In general, the FL setup contains multiple rounds of communication between the clients and the aggregator, and solves (16) using an iterative method (e.g., Li et al. (2020)). Specifically, in each round of FL, the aggregator first broadcasts the current global model parameter to the clients. Each client then performs local computations in parallel, in order to update the model parameter using its local dataset via some gradient-based method. Finally, the clients transmit their updated model parameters to the aggregator for global update (see, e.g. Li et al. (2020), for more details).

Similarly to our discussions in Section 4.1, one has to consider constraints (e.g., communication constraints) in FL, which leads to partial participation of the clients (e.g., Reisizadeh, Mokhtari, Hassani, Jadbabaie, and Pedarsani (2020)). Specifically, given a set $\mathcal{A} \subseteq [m]$ of clients that participate in the FL task, it has been shown (e.g., Li et al. (2020)) that under certain assumptions on $F(\cdot)$ defined in Eq. (16), the FL algorithm (based on the clients in $\mathcal{A}$) converges to an optimal solution, denoted as $\mathbf{w}^\star_{\mathcal{A}}$, to $\min_{\mathbf{w}} \Big\{ F_{\mathcal{A}}(\mathbf{w}) \triangleq \sum_{v\in\mathcal{A}} \frac{|\mathcal{D}_v|}{D} F_v(\mathbf{w}) \Big\}$. We then consider the following client selection problem:

$$\max_{\mathcal{A}\subseteq\mathcal{S}} \Big\{ f_c(\mathcal{A}) \triangleq F(\mathbf{w}_\emptyset) - F(\mathbf{w}^\star_{\mathcal{A}}) \Big\} \qquad s.t. \quad h_F(\mathcal{A}) \leq T, \tag{17}$$

where $\mathbf{w}_\emptyset$ is the initialization of the model parameter.[6] Similarly to our discussions in Section 4.1, we use $h_F : 2^{\mathcal{S}} \to \mathbb{R}_{\geq 0}$ in (17) to characterize the computation latency (for the clients to perform local updates) and the communication latency (for the clients to transmit their local information to the aggregator) in a single round of the FL algorithm.[7] Moreover, we consider the scenario where the clients communicate with the aggregator via a shared channel in a sequential manner. In particular, we may define $h_F(\cdot)$ similarly to $h_c(\cdot)$ given by Eq. (15), and thus the results in Proposition 14 shown for $h_c(\cdot)$ also hold for $h_F(\cdot)$. The constraint in (17) then ensures that the FL algorithm completes within a certain time limit, when the number of total communication rounds is fixed. Hence, problem (17) can now be viewed as an instance of problem (P1). We also prove the following result for the objective function $f_c(\cdot)$ in problem (17).

Proposition 16. *Suppose that for any $\mathcal{A} \subseteq \mathcal{B} \subseteq [m]$, it holds that (1) $F(\mathbf{w}^\star_{\mathcal{A}}) \leq F(\mathbf{w})$ for all $\mathbf{w} \in \mathbb{R}^n$ with supp$(\mathbf{w}) \subseteq$ supp$(\mathbf{w}^\star_{\mathcal{A}})$; (2) supp$(\mathbf{w}^\star_{\mathcal{A}}) \subseteq$ supp$(\mathbf{w}^\star_{\mathcal{B}})$; and (3) $F(\mathbf{w}^\star_{\mathcal{A}}) \geq F(\mathbf{w}^\star_{\mathcal{B}})$. Moreover, suppose that for any $v \in [m]$, the local objective function $F_v(\cdot)$ of client v is strongly convex and smooth with parameters $\mu \in \mathbb{R}_{>0}$ and $\rho \in \mathbb{R}_{>0}$, respectively. Then, $f_c(\cdot)$ in problem (17) is monotone nondecreasing, and both the DR ratio and submodularity ratio of $f_c(\cdot)$ given by Definition 1 are lower bounded by μ/ρ.*

Proof. First, since $F(\mathbf{w}^\star_{\mathcal{A}}) \geq F(\mathbf{w}^\star_{\mathcal{B}})$ for all $\mathcal{A} \subseteq \mathcal{B} \subseteq [m]$, we see from (17) that $f_c(\cdot)$ is monotone nondecreasing. Next, one can show that the global objective function $F(\cdot)$ defined in (16) is also strongly convex and smooth with parameters μ and ρ, respectively. That is, for any $\mathbf{w}_1$, $\mathbf{w}_2$ in the domain of $F(\cdot)$, $\frac{\mu}{2}\|\mathbf{w}_2 - \mathbf{w}_1\|^2 \leq -F(\mathbf{w}_2) + F(\mathbf{w}_1) + \nabla F(\mathbf{w}_1)^\top(\mathbf{w}_2 - \mathbf{w}_1) \leq \frac{\rho}{2}\|\mathbf{w}_2 - \mathbf{w}_1\|^2$. One can now adapt the arguments in the proof of Elenberg et al.

[6] We assume that the FL algorithm converges exactly to $\mathbf{w}^\star_{\mathcal{A}}$ for any $\mathcal{A} \subseteq [m]$. However, the FL algorithm only finds a solution $\tilde{\mathbf{w}}^\star_{\mathcal{A}}$ such that $|F_{\mathcal{A}}(\tilde{\mathbf{w}}^\star_{\mathcal{A}}) - F_{\mathcal{A}}(\mathbf{w}^\star_{\mathcal{A}})| = O(1/T_c)$, where T_c is the number of communication rounds between the aggregator and the clients (Li et al., 2020). Nonetheless, one can use the techniques in Ye, Paré et al. (2021) and extend the results for the greedy algorithms provided in this paper to the setting when there are errors in evaluating the objective function $f(\cdot)$ in (P1).

[7] We ignore the latency corresponding to the aggregator, since it is typically more powerful than the clients (e.g., Shi et al. (2020)).

(2018, Theorem 1) and show that the bounds on the DR ratio and submodularity ratio of $f_c(\cdot)$ hold. Details of the adaption are omitted here in the interest of space. ■

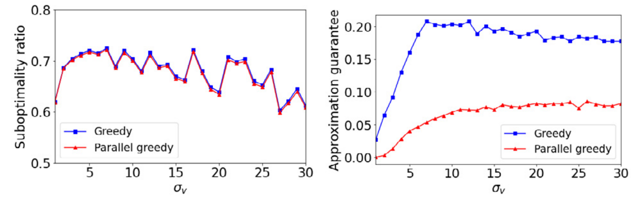
Recalling (8), substituting γ_f with μ/ρ from Proposition 16, and substituting $\tilde{\alpha}_i$ with $\tilde{\alpha}'_{h_c}$ from Proposition 14, one can obtain the approximation guarantee of Algorithm 2 when applied to solve (17).

One can check that a sufficient condition for assumptions (1)–(3) made in Proposition 16 to hold is that the datasets from different clients in $[m]$ are non-i.i.d. in the sense that different datasets contain data points with different features, i.e., $\text{supp}(x_{u,i}) \cap \text{supp}(x_{v,j}) = \emptyset$ for any $u, v \in [m]$ (with $u \neq v$), and for any $i \in \mathcal{D}_u$ and any $j \in \mathcal{D}_v$, where $x_{u,i}, x_{v,j} \in \mathbb{R}^n$. In this case, an element $\mathbf{w}_i$ in the model parameter $\mathbf{w} \in \mathbb{R}^n$ corresponds to one feature $(x_{u,j})_i$ of the data point $x_{u,j} \in \mathbb{R}^n$, and $\text{supp}(\mathbf{w}^\star_{\mathcal{A}}) = \cup_{u\in\mathcal{A}, i\in\mathcal{D}_u} \text{supp}(x_{u,i})$ for any $\mathcal{A} \subseteq [m]$ (Elenberg et al., 2018). Note that the datasets from different clients are typically assumed to be non-i.i.d. in FL, since the clients may obtain the local datasets from different data sources (Li et al., 2020; McMahan, Moore, Ramage, Hampson, & y Arcas, 2017; Shi et al., 2020). Moreover, strong convexity and smoothness hold for the loss functions in, e.g., (regularized) linear regression and logistic regression (Li et al., 2020; Shi et al., 2020). If assumptions (1)–(3) in Proposition 16 do not hold, one may use a surrogate for $f_c(\cdot)$ in (17) (see Ye and Gupta (2021) for more details).

The FL client selection problem has been studied under various different scenarios (e.g., Balakrishnan et al. (2021), Nishio and Yonetani (2019) and Shi et al. (2020)). In Nishio and Yonetani (2019), the authors studied a similar client selection problem to the one in this paper, but the objective function considered in Nishio and Yonetani (2019) is simply the sum of sizes of the datasets at the selected clients. In Shi et al. (2020), the authors studied a joint optimization problem of bandwidth allocation and client selection, under the training time constraints. However, these works do not provide theoretical performance guarantees for the proposed algorithms. In Balakrishnan et al. (2021), the authors considered a client selection problem with a cardinality constraint on the set of selected clients.

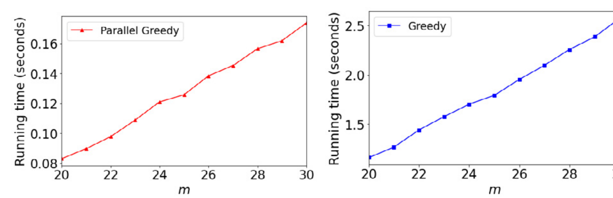
5. Numerical results

We consider the sensor scheduling problem introduced in (13) in Section 4.1, where $h_k(\cdot)$ corresponds to a communication constraint on the set of sensors scheduled for time step k and is defined in Eq. (15), for all $k \in \{0, \dots, \ell\}$. Let the target time step be $\ell = 2$, and generate the system matrices $A_k \in \mathbb{R}^{3\times 3}$ and $C_k \in \mathbb{R}^{3\times 3}$ in a random manner, for all $k \in \{0, \dots, \ell\}$. Each row in C_k corresponds to a candidate sensor at time step k. We set the input noise covariance as $W = 2I_3$, the measurement noise covariance as $\sigma_v^2 I_3$ with $\sigma_v \in \{1, \dots, 30\}$, and the covariance of x_0 as $\Pi_0 = I_3$. The ground set $\mathcal{S} = \{s_{i,k} : i \in [m], k \in \{0, \dots, \ell\}\}$ that contains all the candidate sensors satisfies $|\mathcal{S}| = 3 \times 3 = 9$. For any $k \in \{0, \dots, \ell\}$ and any $i \in [m]$, we generate the computation latency and transmission latency of sensor $s_{i,k}$, denoted as $c_{i,k}$ and $t_{i,k}$, respectively, by sampling exponential distributions with parameters 0.5 and 0.2, respectively. Finally, for any $k \in \{0, \dots, \ell\}$, we set the communication constraint to be $H_k = h_k(\mathcal{S}_k)/2$. We apply Algorithms 1 and 2 to solve the instances of problem (13) constructed above. In Fig. 1(a), we plot the actual performances of Algorithms 1–2 for $\sigma_v \in \{1, \dots, 30\}$, where the actual performance of Algorithm 1 (resp., Algorithm 2) is given by $f_s(\mathcal{A}^r)/f_s(\mathcal{A}^\star)$ (resp., $f_s(\mathcal{A}^g)/f_s(\mathcal{A}^\star)$), where $f_s(\cdot)$ is given in (13), $\mathcal{A}^r$ (resp., $\mathcal{A}^g$) is the solution to (13) returned by Algorithm 1 (resp., Algorithm 2), and $\mathcal{A}^\star$ is an optimal solution to (13) (obtained by brute force). In Fig. 1(b), we plot the approximation guarantees of Algorithms 1 and 2 given by Theorems 6 and 9,



(a) (b)

Fig. 1. Actual performances and approximation guarantees of Algorithms 1–2.



(a) (b)

Fig. 2. Running times of Algorithms 1–2.

respectively. For any $\sigma_v \in \{1, \dots, 30\}$, the results in Fig. 1(a)–(b) are averaged over 50 random instances of problem (13) constructed above. From Fig. 1(a), we see that the actual performance of Algorithm 2 is slightly better than that of Algorithm 1. We also see that as σ_v increases from 1 to 30, the actual performances of Algorithms 1 and 2 first tends to be better and then tends to be worse. Compared to Fig. 1(a), (b) shows that the approximation guarantees of Algorithms 1 and 2 provided by Theorems 6 and 9, respectively, are conservative. However, a tighter approximation guarantee potentially yields a better actual performance of the algorithm.

In Fig. 2, we plot the running times of Algorithms 1 and 2 when applied to solve similar random instances of problem (13) to those described above but with $A_k \in \mathbb{R}^{10\times 10}$ and $C_k \in \mathbb{R}^{m\times 10}$ for $m \in \{20, \dots 30\}$. Note that the simulations are conducted on a Mac with 8-core CPU, and for any $m \in \{20, \dots, 30\}$ the results in Fig. 2(a)–(b) are averaged over 5 random instances of problem (13). Fig. 2 shows that Algorithm 1 runs faster than Algorithm 2 matching with our discussions in Section 3.

6. Conclusion

We studied the problem of maximizing a monotone nondecreasing set function under multiple constraints, where the constraints are upper bound constraints characterized by monotone nondecreasing set functions. We proposed two greedy algorithms to solve the problem, and analyzed the approximation guarantees of the algorithms, leveraging the notions of submodularity ratio and curvature of set functions. We discussed several important real-world applications of the general problem, and provided bounds on the submodularity ratio and curvature of the set functions in the corresponding instances of the problem. Numerical results show that the two greedy algorithms yield comparable performances that are reasonably good in practice.

Appendix A. Proof of Theorem 6

Lemma 17. *Under the setting of Theorem 6, we have*

$$f(\mathcal{A}_i^r) \geq \frac{\min\{1, \tilde{\gamma}_f\}}{2}(1 - e^{-(1-\tilde{\alpha}_i)\gamma_f})f(\mathcal{A}_i^\star), \tag{A.1}$$

for all $i \in [n]$.

Proof. Under Assumption 4, we see from the definitions of (P1) and Algorithm 1 that $A^r = \cup_{i\in[n]}\mathcal{A}_i^r$ and $\mathcal{A}^\star = \cup_{i\in[n]}\mathcal{A}_i^\star$, where $\mathcal{A}_i^r, \mathcal{A}_i^\star \subseteq \mathcal{S}_i$ for all $i \in [n]$, and $\mathcal{A}_i^r \cap \mathcal{A}_j^r = \emptyset$, $\mathcal{A}_i^\star \cap \mathcal{A}_j^\star = \emptyset$ for all $i, j \subseteq [n]$ with $i \neq j$. Now, considering any $i \in [n]$, we note that (A.1) trivially holds if $\tilde{\gamma}_f = 0$, $\gamma_f = 0$ or $\tilde{\alpha}_i = 1$. Thus, in the remaining of this proof, we let $\tilde{\gamma}_f \in \mathbb{R}_{>0}$, $\gamma_f \in (0, 1]$ and $\tilde{\alpha}_i \in [0, 1)$. Recalling that we have assumed that $h_i(v) > 0$ for all $v \in \mathcal{S}$, we then have from Definition 2 that $\delta_v^i(\mathcal{A}) > 0$ for all $\mathcal{A} \subseteq \mathcal{S}$ and for all $v \in \mathcal{S} \setminus \mathcal{A}$, which implies that line 5 of Algorithm 1 is well-defined. Recall from Observation 1 that $\mathcal{A}_{i,j}^r = \{q_1, \dots, q_j\}$ for all $j \in [|\mathcal{A}_i^r|]$ with $\mathcal{A}_{i,0}^r = \emptyset$. Moreover, denote $\tilde{\mathcal{A}}_{i,j} = \{\tilde{q}_1, \dots, \tilde{q}_j\}$ for all $j \in [l_i + 1]$ with $\tilde{\mathcal{A}}_{i,0} = \emptyset$, where $\tilde{q}_i = q_i$ for all $i \in [l_i]$ and $\tilde{q}_{j+1} = v_{l_i+1}^\star$, where l_i and $v_{l_i+1}^\star$ are given in Observation 1. Now, considering any $j \in \{0, 1, \dots, l_i\}$ and denoting $\tilde{\mathcal{A}}_i^\star = \mathcal{A}_i^\star \setminus \tilde{\mathcal{A}}_{i,j}^r = \{p_1, \dots, p_{|\tilde{\mathcal{A}}_i^\star|}\}$, we have

$$
\begin{aligned}
&f(\mathcal{A}_i^\star \cup \tilde{\mathcal{A}}_{i,j}^r) - f(\tilde{\mathcal{A}}_{i,j}^r) \le \frac{1}{\gamma_f}\sum_{k=1}^{|\tilde{\mathcal{A}}_i^\star|}\delta_{p_k}(\tilde{\mathcal{A}}_{i,j}^r)\\
&= \frac{1}{\gamma_f}\sum_{k=1}^{|\tilde{\mathcal{A}}_i^\star|}\frac{\delta_{p_k}(\tilde{\mathcal{A}}_{i,j})}{\delta_{p_k}^i(\tilde{\mathcal{A}}_{i,j}^r)}\delta_{p_k}^i(\tilde{\mathcal{A}}_{i,j}^r) \le \frac{\delta_{\tilde{q}_{j+1}}(\tilde{\mathcal{A}}_{i,j}^r)}{\gamma_f\delta_{\tilde{q}_{j+1}}^i(\tilde{\mathcal{A}}_{i,j}^r)}\sum_{k=1}^{|\tilde{\mathcal{A}}_i^\star|}\delta_{p_k}^i(\tilde{\mathcal{A}}_{i,j}^r)\\
&\le \frac{\delta_{\tilde{q}_{j+1}}(\tilde{\mathcal{A}}_{i,j}^r)}{\delta_{\tilde{q}_{j+1}}^i(\tilde{\mathcal{A}}_{i,j}^r)\gamma_f(1-\tilde{\alpha}_i)}\sum_{k=1}^{|\tilde{\mathcal{A}}_i^\star|}\delta_{p_k}^i(\{p_1, \dots, p_{k-1}\})\\
&= \frac{\delta_{\tilde{q}_{j+1}}(\tilde{\mathcal{A}}_{i,j}^r)}{(h_i(\tilde{\mathcal{A}}_{i,j+1}^r) - h_i(\tilde{\mathcal{A}}_{i,j}^r))\gamma_f(1-\tilde{\alpha}_i)}h_i(\tilde{\mathcal{A}}_i^\star),
\end{aligned} \tag{A.2}
$$

where the first inequality follows from Definition 1, the second inequality follows from the greedy choice in line 5 of Algorithm 1, and the third inequality follows from Definition 2. To proceed, denoting $\Delta_j \triangleq f(\mathcal{A}_i^\star) - f(\tilde{\mathcal{A}}_{i,j}^r)$ for all $j \in \{0, 1, \dots, l_i + 1\}$, and noting that $h_i(\tilde{\mathcal{A}}_i^\star) \le h_i(\mathcal{A}_i^\star) \le H_i$, we have from (A.2) the following:

$$
\begin{aligned}
&\Delta_{j+1} \le \Big(1 - \frac{\gamma_f(1-\tilde{\alpha}_i)(h_i(\tilde{\mathcal{A}}_{i,j+1}^r) - h_i(\tilde{\mathcal{A}}_{i,j}^r))}{H_i}\Big)\Delta_j\\
&\Rightarrow \Delta_{l_i+1} \le \Delta_0\prod_{j=0}^{l_i}\Big(1 - \frac{\gamma_f(1-\tilde{\alpha}_i)(h_i(\tilde{\mathcal{A}}_{i,j+1}^r) - h_i(\tilde{\mathcal{A}}_{i,j}^r))}{H_i}\Big)\\
&\qquad \le \Delta_0\prod_{j=0}^{l_i}\Big(1 - \frac{\gamma_f(1-\tilde{\alpha}_i)(h_i(\tilde{\mathcal{A}}_{i,j+1}^r) - h_i(\tilde{\mathcal{A}}_{i,j}^r))}{h_i(\tilde{\mathcal{A}}_{i,l_i+1}^r)}\Big),
\end{aligned}
$$

where the third inequality follows from $h_i(\tilde{\mathcal{A}}_{i,l_i+1}^r) > H_i$ as we argued above. Note the fact that if $a_1, \dots, a_n \in \mathbb{R}_{>0}$ such that $\sum_{i=1}^n a_i = \alpha G$, where $G \in \mathbb{R}_{>0}$ and $\alpha \in (0, 1]$, then the function $\Pi_{i=1}^n(1 - \frac{a_i}{G})$ achieves its maximum at $a_1 = \dots = a_n = \frac{\alpha G}{n}$ (Kulik et al., 2009). Since $\sum_{j=0}^{l_i}(h_i(\tilde{\mathcal{A}}_{i,j+1}^r) - h_i(\tilde{\mathcal{A}}_{i,j}^r)) = h_i(\tilde{\mathcal{A}}_{i,l_i+1}^r)$ and $\gamma_f(1 - \tilde{\alpha}_i) < 1$, we have

$$
\begin{aligned}
&\prod_{j=0}^{l_i}\Big(1 - \frac{\gamma_f(1-\tilde{\alpha}_i)(h_i(\tilde{\mathcal{A}}_{i,j+1}^r) - h_i(\tilde{\mathcal{A}}_{i,j}^r))}{h_i(\tilde{\mathcal{A}}_{i,l_i+1}^r)}\Big)\\
&\le \prod_{j=0}^{l_i}\Big(1 - \frac{\gamma_f(1-\tilde{\alpha}_i)\frac{h_i(\tilde{\mathcal{A}}_{i,l_i+1}^r)}{l_i+1}}{h_i(\tilde{\mathcal{A}}_{i,l_i+1}^r)}\Big) = \Big(1 - \frac{\gamma_f(1-\tilde{\alpha}_i)}{l_i+1}\Big)^{l_i+1}.
\end{aligned}
$$

It follows that

$$
\begin{aligned}
&\Delta_{l_i+1} \le \Big(1 - \frac{\gamma_f(1-\tilde{\alpha}_i)}{l_i+1}\Big)^{l_i+1}\Delta_0 \le e^{-\gamma_f(1-\tilde{\alpha}_i)}\Delta_0\\
&\Rightarrow f(\tilde{\mathcal{A}}_{i,l_i+1}^r) \ge (1 - e^{-\gamma_f(1-\tilde{\alpha}_i)})f(\mathcal{A}_i^\star).
\end{aligned} \tag{A.3}
$$

Recalling from Definition 5 that $f(\tilde{\mathcal{A}}_{i,l_i+1}^r) - f(\tilde{\mathcal{A}}_{i,l_i}^r) \le \frac{1}{\tilde{\gamma}_f}f(\mathcal{B}_i^r)$, we obtain from (A.3) that $f(\tilde{\mathcal{A}}_{i,l_i}^r) + \frac{1}{\tilde{\gamma}_f}f(\mathcal{B}_i^r) \ge (1 - e^{-\gamma_f(1-\tilde{\alpha}_i)})f(\mathcal{A}_i^\star)$. Since $f(\cdot)$ is monotone nondecreasing, it follows that $f(\mathcal{A}_i^r) + \frac{1}{\tilde{\gamma}_f}f(\mathcal{B}_i^r) \ge (1 - e^{-\gamma_f(1-\tilde{\alpha}_i)})f(\mathcal{A}_i^\star)$, which implies that at least one of $f(\mathcal{A}_i^r)$ and $\frac{1}{\tilde{\gamma}_f}f(\mathcal{B}_i^r)$ is greater than or equal to $\frac{1}{2}(1 - e^{-\gamma_f(1-\tilde{\alpha}_i)})f(\mathcal{A}_i^\star)$. Thus, we see from line 9 in Algorithm 1 that (A.1) holds. ■

Proof of Theorem 6. Since (6) naturally holds if $\kappa_f = 0$, we let $\kappa_f \in (0, 1]$ in this proof. Considering $\mathcal{A}^r = \cup_{i\in[n]}\mathcal{A}_i^r$ returned by Algorithm 1, and denoting $\mathcal{A}_1 = \cup_{i\in[n-1]}\mathcal{A}_i^r$ and $\mathcal{A}_2 = \mathcal{A}_n^r = \{q_1, \dots, q_{|\mathcal{A}_n^r|}\}$, we have

$$
\begin{aligned}
f(\mathcal{A}_1 \cup \mathcal{A}_2) &= f(\mathcal{A}_1) + \sum_{j=1}^{|\mathcal{A}_n^r|}\delta_{q_j}(\mathcal{A}_1 \cup \{q_1, \dots, q_{j-1}\})\\
&\ge f(\mathcal{A}_1) + (1 - \alpha_f)\sum_{j=1}^{|\mathcal{A}_n^r|}\delta_{q_j}(\{q_1, \dots, q_{j-1}\})\\
&= f(\mathcal{A}_1) + (1 - \alpha_f)f(\mathcal{A}_2).
\end{aligned} \tag{A.4}
$$

where the inequality uses the fact that $q_j \notin A_1$ (from Assumption 4) and Definition 2. Repeating the above arguments for (A.4), one can show that $f(\cup_{i\in[n]}\mathcal{A}_i^r) \ge (1 - \alpha_f)\sum_{i=1}^n f(\mathcal{A}_i^r)$, which implies via (A.1) that

$$
\begin{aligned}
&f(\mathcal{A}^r) \ge \frac{(1-\alpha_f)\min\{1, \tilde{\gamma}_f\}}{2}\\
&\times \min_{i\in[n]}(1 - e^{-(1-\tilde{\alpha}_i)\gamma_i})\sum_{i=1}^n f(\mathcal{A}_i^\star).
\end{aligned} \tag{A.5}
$$

Now, consider the optimal solution $\mathcal{A}^\star = \cup_{i\in[n]}\mathcal{A}_i^\star$. Using similar arguments to those above, and recalling the definition of the DR ratio κ_f of $f(\cdot)$ given in Definition 1, one can show that $f(\cup_{i\in[n]}\mathcal{A}_i^\star) \le \frac{1}{\kappa_f}\sum_{i=1}^n f(\mathcal{A}_i^\star)$, which together with (A.5) complete the proof of (6). ■

Appendix B. Proof of Theorem 9

First, note that (7) trivially holds if $\gamma_f = 0$, $\alpha_h = 1$, or $f(\mathcal{A}^\star) = 0$. Thus, we let $\gamma_f \in (0, 1]$, $\tilde{\alpha}_i \in [0, 1)$, and $f(\mathcal{A}^\star) > 0$ in this proof. For our analysis in this proof, we assume without loss of generality that $f(\cdot)$ is normalized such that $f(\mathcal{A}^\star) = 1$. Also recalling that we have assumed that $h_i(v) > 0$ for all $v \in \mathcal{S}$, we know from Definition 2 that $\delta_v^i(\mathcal{A}) = h_i(\mathcal{A} \cup \{v\}) - h_i(\mathcal{A}) > 0$ for all $\mathcal{A} \subseteq \mathcal{S}$ and for all $v \in \mathcal{S} \setminus \mathcal{A}$, which implies that ψ_j in Definition 8 and line 3 of Algorithm 1 are well-defined. Denote $\mathcal{A}^g = \{q_1, \dots, q_{|\mathcal{A}^g|}\}$, and $\mathcal{A}_j^g = \{q_1, \dots, q_j\}$ for all $j \in [|\mathcal{A}^g|]$ with $\mathcal{A}_0^g = \emptyset$. We claim that in the remaining of this proof, we can further assume without loss of generality that $\psi_j > 0$ for all $j \in \{0, \dots, |\mathcal{A}^g| - 1\}$. To prove this claim, we first assume (for contradiction) that there exists $j \in \{0, \dots, |\mathcal{A}^g| - 1\}$ such that $\delta_{q_{j+1}}(\mathcal{A}_j^g) = 0$. Since $\gamma_f \in (0, 1]$ as we argued above, we see from Definition 1 and the greedy choice that $\delta_{\mathcal{W}}(\mathcal{A}_j^g) = 0$, where $\mathcal{W}$ is defined and iteratively updated in Algorithm 2. Recalling that $f(\cdot)$ is monotone nondecreasing as we assumed before, we then have that $f(\mathcal{A}_j^g \cup \mathcal{W}_1) = f(\mathcal{A}_j^g \cup \mathcal{W}_2)$ for all $\mathcal{W}_1, \mathcal{W}_2 \subseteq \mathcal{W}$. It follows that $\delta_{q_{j'+1}}(\mathcal{A}_{j'}^g) = 0$ for all $j' \in \{j, \dots, |\mathcal{A}^g| - 1\}$. In other words, Algorithm 2 is vacuous after adding q_j to the greedy solution $\mathcal{A}^g$. Thus, we can assume without loss of generality that $\delta_{q_{j+1}}(\mathcal{A}_j^g) > 0$ for all $j \in \{0, \dots, |\mathcal{A}^g| - 1\}$, which implies via Definition 8 that $\psi_j > 0$ for all $j \in \{0, \dots, |\mathcal{A}^g| - 1\}$.

To proceed, let us consider any $j \in \{0, \ldots, |\mathcal{A}^g|-1\}$. Denoting $\tilde{\mathcal{A}}_j^\star = \mathcal{A}^\star \setminus \mathcal{A}_j^g$, we have

$$\begin{aligned} f(\mathcal{A}^\star) &\leq f(\mathcal{A}_j^g) + \frac{1}{\gamma_f} \sum_{v \in \tilde{\mathcal{A}}_j^\star} \delta_v(\mathcal{A}_j^g) \\ &\leq f(\mathcal{A}_j^g) + \frac{1}{\gamma_f} \sum_{i \in [n]} \sum_{v \in \tilde{\mathcal{A}}_j^\star \cap \mathcal{S}_i} \frac{\delta_v(\mathcal{A}_j^g)}{\delta_v^i(\mathcal{A}_j^g)} \delta_v^i(\mathcal{A}_j^g) \\ &\leq f(\mathcal{A}_j^g) + \frac{\delta_{q_{j+1}}(\mathcal{A}_j^g)}{\gamma_f \psi_j \delta_{q_{j+1}}^{i_j}(\mathcal{A}_j^g)} \sum_{i \in [n]} \sum_{v \in \tilde{\mathcal{A}}_j^\star \cap \mathcal{S}_i} \delta_v^i(\mathcal{A}_j^g), \end{aligned} \tag{B.1}$$

where the first inequality follows from Definition 1 and the monotonicity of $f(\cdot)$, and the third inequality follows from Definition 8. Denoting $\tilde{\mathcal{A}}_j^\star \cap \mathcal{S}_i = \{p_1^i, \ldots, p_{|\tilde{\mathcal{A}}_j^\star \cap \mathcal{S}_i|}^i\}$ for all $i \in [n]$, we further obtain from Definition 2 that

$$\begin{aligned} &\sum_{i \in [n]} \sum_{v \in \tilde{\mathcal{A}}_j^\star \cap \mathcal{S}_i} \delta_v^i(\mathcal{A}_j^g) \\ &\leq \frac{1}{1-\alpha_h} \sum_{i \in [n]} \sum_{k=1}^{|\tilde{\mathcal{A}}_j^\star \cap \mathcal{S}_i|} \delta_{p_k^i}^i(\{p_1^i, \ldots, p_{k-1}^i\}) \\ &= \frac{1}{1-\alpha_h} \sum_{i \in [n]} h_i(\tilde{\mathcal{A}}_j^\star \cap \mathcal{S}_i) \leq \frac{\sum_{i \in [n]} H_i}{1-\alpha_h}. \end{aligned} \tag{B.2}$$

Denoting

$$M_j = \frac{\sum_{i \in [n]} H_i}{(1-\alpha_h)\gamma_f \psi_j \delta_{q_{j+1}}^{i_j}(\mathcal{A}_j^g)}, \tag{B.3}$$

and $\delta_j = \delta_{q_{j+1}}(\mathcal{A}_j^g)$ for all $j \in \{0, \ldots, |\mathcal{A}^g|-1\}$, we can combine (B.1)–(B.2) and obtain that

$$M_j \delta_j + \sum_{k=0}^{j-1} \delta_k \geq 1, \ \forall j \in \{0, \ldots, |\mathcal{A}^g|-1\}, \tag{B.4}$$

where we use the facts that $f(\mathcal{A}_j^g) = \sum_{k=0}^{j-1} \delta_k$ and $f(\mathcal{A}^\star) = 1$ as we argued before.

In order to prove the approximation guarantee of Algorithm 2 given by (7), we aim to provide a lower bound on $f(\mathcal{A}^g)/f(\mathcal{A}^\star)$ and we achieve this by first minimizing $f(\mathcal{A}^g)/f(\mathcal{A}^\star) = \sum_{j=0}^{|\mathcal{A}^g|-1} \delta_j$ subject to the constraints given in (B.4). In other words, we consider the following linear program and its dual:

$$\begin{aligned} &\min_{\delta_j} \sum_{j=0}^{|\mathcal{A}^g|-1} \delta_j \\ &s.t.\ M_j \delta_j + \sum_{k=0}^{j-1} \delta_k \geq 1, \ \forall j \in \{0, \ldots, |\mathcal{A}^g|-1\}, \end{aligned} \tag{B.5}$$

$$\begin{aligned} &\max_{\mu_k} \sum_{k=0}^{|\mathcal{A}^g|-1} \mu_k \\ &s.t.\ \mu_k \geq 0, \ \forall k \in \{0, \ldots, |\mathcal{A}^g|-1\} \\ &\quad M_j \mu_j + \sum_{k=j+1}^{|\mathcal{A}^g|-1} \mu_k = 1, \ \forall j \in \{0, \ldots, |\mathcal{A}^g|-1\}. \end{aligned} \tag{B.6}$$

We will then show that the optimal cost of (B.5) and (B.6) satisfies (7). Noting from Eq. (B.3) that $M_j > 0$ for all $j \in \{0, \ldots, |\mathcal{A}^g|-1\}$, we can obtain the optimal solution to (B.6) by solving the equations in u_k given by the equality constraints in (B.6), which yields $\mu_k^\star = \frac{1}{M_k} \prod_{j=k+1}^{|\mathcal{A}^g|-1} (1 - \frac{1}{M_j}) \ \forall k \in \{0, \ldots, |\mathcal{A}^g|-1\}$. Noting from Eq. (B.3) that $\sum_{k=0}^{|\mathcal{A}^g|-1} \frac{1}{M_k} = B$, we can further lower bound $\sum_{k=0}^{|\mathcal{A}^g|-1} \mu_k^\star$ by solving

$$\min_{\nu_k} \sum_{k=0}^{|\mathcal{A}^g|-1} \nu_k \prod_{j=k+1}^{|\mathcal{A}^g|-1} (1-\nu_j) \quad s.t. \sum_{k=0}^{|\mathcal{A}^g|-1} \nu_k = B. \tag{B.7}$$

One can obtain the optimal solution to (B.7) by considering a Lagrangian multiplier corresponding to the equality constraint in (B.7), which yields the optimal solution $\nu_k = \frac{B}{|\mathcal{A}^g|} \ \forall k \in \{0, \ldots, |\mathcal{A}^g|-1\}$. It then follows from our arguments above that

$$\begin{aligned} \sum_{k=0}^{|\mathcal{A}^g|-1} \mu_k^\star &\geq \sum_{k=0}^{|\mathcal{A}^g|-1} \frac{B}{|\mathcal{A}^g|} \prod_{j=k+1}^{|\mathcal{A}^g|-1} (1 - \frac{B}{|\mathcal{A}^g|}) \\ &= 1 - (1 - \frac{B}{|\mathcal{A}^g|})^{|\mathcal{A}^g|} \geq 1 - e^{-B}. \quad \blacksquare \end{aligned}$$

References

Anderson, Brian D. O., & Moore, John B. (1979). *Optimal filtering*. Dover Books.

Balakrishnan, Ravikumar, Li, Tian, Zhou, Tianyi, Himayat, Nageen, Smith, Virginia, & Bilmes, Jeff (2021). Diverse client selection for federated learning via submodular maximization. In *Proc. International Conference on Learning Representations*.

Bian, Andrew An, Buhmann, Joachim M., Krause, Andreas, & Tschiatschek, Sebastian (2017). Guarantees for greedy maximization of non-submodular functions with applications. In *Proc. International Conference on Machine Learning* (pp. 498–507).

Calinescu, Gruia, Chekuri, Chandra, Pal, Martin, & Vondrák, Jan (2011). Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, *40*(6), 1740–1766.

Chamon, Luiz F. O., Pappas, George J., & Ribeiro, Alejandro (2017). The mean square error in Kalman filtering sensor selection is approximately supermodular. In *Proc. IEEE Conference on Decision and Control* (pp. 343–350).

Chepuri, Sundeep Prabhakar, & Leus, Geert (2014). Sparsity-promoting sensor selection for non-linear measurement models. *IEEE Transactions on Signal Processing*, *63*(3), 684–698.

Conforti, Michele, & Cornuéjols, Gérard (1984). Submodular set functions, matroids and the greedy algorithm: tight worst-case bounds and some generalizations of the Rado-Edmonds theorem. *Discrete Applied Mathematics*, *7*(3), 251–274.

Das, Abhimanyu, & Kempe, David (2011). Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. In *Proc. International Conference on Machine Learning* (pp. 1057–1064).

Das, Abhimanyu, & Kempe, David (2018). Approximate submodularity and its applications: Subset selection, sparse approximation and dictionary selection. *Journal of Machine Learning Research*, *19*(1), 74–107.

Dinh, Canh T., Tran, Nguyen H., Nguyen, Minh N. H., Hong, Choong Seon, Bao, Wei, Zomaya, Albert Y., et al. (2020). Federated learning over wireless networks: Convergence analysis and resource allocation. *IEEE/ACM Transactions on Networking*, *29*(1), 398–409.

Durga, S., Iyer, Rishabh, Ramakrishnan, Ganesh, & De, Abir (2021). Training data subset selection for regression with controlled generalization error. In *Proc. International Conference on Machine Learning* (pp. 9202–9212).

Elenberg, Ethan R., Khanna, Rajiv, Dimakis, Alexandros G., & Negahban, Sahand (2018). Restricted strong convexity implies weak submodularity. *The Annals of Statistics*, *46*(6B), 3539–3568.

Feige, Uriel (1998). A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, *45*(4), 634–652.

Fisher, Marshall L., Nemhauser, George L., & Wolsey, Laurence A. (1978). An analysis of approximations for maximizing submodular set functions—II. In *Polyhedral Combinatorics* (pp. 73–87). Springer.

Gupta, Vijay, Chung, Timothy H., Hassibi, Babak, & Murray, Richard M. (2006). On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage. *Automatica*, *42*(2), 251–260.

Huber, Marco F. (2011). Optimal pruning for multi-step sensor scheduling. *IEEE Transactions on Automatic Control*, *57*(5), 1338–1343.

Iyer, Rishabh, & Bilmes, Jeff (2012). Algorithms for approximate minimization of the difference between submodular functions, with applications. In *Proc. Conference on Uncertainty in Artificial Intelligence* (pp. 407–417).

Jawaid, Syed Talha, & Smith, Stephen L. (2015). Submodularity and greedy algorithms in sensor scheduling for linear dynamical systems. *Automatica*, *61*, 282–288.

Joshi, Siddharth, & Boyd, Stephen (2008). Sensor selection via convex optimization. *IEEE Transactions on Signal Processing*, *57*(2), 451–462.

Khuller, Samir, Moss, Anna, & Naor, Joseph Seffi (1999). The budgeted maximum coverage problem. *Information Processing Letters*, *70*(1), 39–45.

Kohara, Akira, Okano, Kunihisa, Hirata, Kentaro, & Nakamura, Yukinori (2020). Sensor placement minimizing the state estimation mean square error: Performance guarantees of greedy solutions. In *Proc. IEEE Conference on Decision and Control* (pp. 1706–1711).

Krause, Andreas, Singh, Ajit, & Guestrin, Carlos (2008). Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, *9*(Feb), 235–284.

Kuhnle, Alan, Smith, J. David, Crawford, Victoria, & Thai, My (2018). Fast maximization of non-submodular, monotonic functions on the integer lattice. In *Proc. International Conference on Machine Learning* (pp. 2786–2795).

Kulik, Ariel, Shachnai, Hadas, & Tamir, Tami (2009). Maximizing submodular set functions subject to multiple linear constraints. In *Proc. Annual ACM-SIAM Symposium on Discrete Algorithms* (pp. 545–554).

Leskovec, Jure, Krause, Andreas, Guestrin, Carlos, Faloutsos, Christos, VanBriesen, Jeanne, & Glance, Natalie (2007). Cost-effective outbreak detection in networks. In *Proc. International Conference on Knowledge Discovery and Data Mining* (pp. 420–429).

Li, Xiang, Huang, Kaixuan, Yang, Wenhao, Wang, Shusen, & Zhang, Zhihua (2020). On the convergence of FedAvg on non-IID data. In *Proc. International Conference on Learning Representations*.

McMahan, Brendan, Moore, Eider, Ramage, Daniel, Hampson, Seth, & y Arcas, Blaise Aguera (2017). Communication-efficient learning of deep networks from decentralized data. In *Proc. Artificial Intelligence and Statistics* (pp. 1273–1282).

Mo, Yilin, Ambrosino, Roberto, & Sinopoli, Bruno (2011). Sensor selection strategies for state estimation in energy constrained wireless sensor networks. *Automatica*, *47*(7), 1330–1338.

Nemhauser, George L., Wolsey, Laurence A., & Fisher, Marshall L. (1978). An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, *14*(1), 265–294.

Nishio, Takayuki, & Yonetani, Ryo (2019). Client selection for federated learning with heterogeneous resources in mobile edge. In *Proc. IEEE International Conference on Communications* (pp. 1–7).

Prasad, Amritha, Hudack, Jeffrey, Mou, Shaoshuai, & Sundaram, Shreyas (2022). Policies for risk-aware sensor data collection by mobile agents. *Automatica*, *142*, Article 110391.

Reisizadeh, Amirhossein, Mokhtari, Aryan, Hassani, Hamed, Jadbabaie, Ali, & Pedarsani, Ramtin (2020). Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In *Proc. International Conference on Artificial Intelligence and Statistics* (pp. 2021–2031).

Shi, Wenqi, Zhou, Sheng, Niu, Zhisheng, Jiang, Miao, & Geng, Lu (2020). Joint device scheduling and resource allocation for latency constrained wireless federated learning. *IEEE Transactions on Wireless Communication*, *20*(1), 453–467.

Tzoumas, Vasileios, Carlone, Luca, Pappas, George J., & Jadbabaie, Ali (2020). LQG control and sensing co-design. *IEEE Transactions on Automatic Control*, *66*(4), 1468–1483.

Tzoumas, Vasileios, Jadbabaie, Ali, & Pappas, George J. (2016). Sensor placement for optimal Kalman filtering: Fundamental limits, submodularity, and algorithms. In *Proc. American Control Conference* (pp. 191–196).

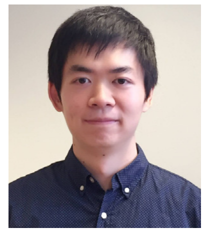
Ye, Lintao, & Gupta, Vijay (2021). Client scheduling for federated learning over wireless networks: A submodular optimization approach. In *Proc. IEEE Conference on Decision and Control* (pp. 63–68).

Ye, Lintao, Paré, Philip E., & Sundaram, Shreyas (2021). Parameter estimation in epidemic spread networks using limited measurements. *SIAM Journal on Control and Optimization*, *60*(2), S49–S74.

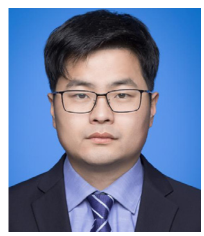
Ye, Lintao, & Sundaram, Shreyas (2019). Sensor selection for hypothesis testing: Complexity and greedy algorithms. In *Proc. IEEE Conference on Decision and Control* (pp. 7844–7849).

Ye, Lintao, Woodford, Nathaniel, Roy, Sandip, & Sundaram, Shreyas (2021). On the complexity and approximability of optimal sensor selection and attack for Kalman filtering. *IEEE Transactions on Automatic Control*, *66*(5), 2146–2161.

Zhang, Haotian, Ayoub, Raid, & Sundaram, Shreyas (2017). Sensor selection for Kalman filtering of linear dynamical systems: Complexity, limitations and greedy algorithms. *Automatica*, *78*, 202–210.



Lintao Ye is a Lecturer in the School of Artificial Intelligence and Automation at the Huazhong University of Science and Technology, Wuhan, China. He received his B.E degree in Material Science and Engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2015. He received his M.S. degree in Mechanical Engineering in 2017, and his Ph.D. degree in Electrical and Computer Engineering in 2020, both from Purdue University, IN, USA. He was a Postdoctoral Researcher at the University of Notre Dame, IN, USA. His research interests are in the areas of optimization algorithms, control theory, estimation theory, and network science.



Zhi-Wei Liu is a Professor at the School of Artificial Intelligence and Automation at the Huazhong University of Science and Technology, Wuhan, China. He received the B.S. degree in Information Management and Information System from Southwest Jiaotong University, Chengdu, China, in 2004, and the Ph.D. degree in Control Science and Engineering from the Huazhong University of Science and Technology in 2011. His current research interests include cooperative control and optimization of distributed network systems.

Ming Chi is a Professor in the School of Artificial Intelligence and Automation at the Huazhong University of Science and Technology, Wuhan, China. He received the Ph.D. degree in Control Science and Engineering from the Huazhong University of Science and Technology in 2013. His research interests include networked control systems, multi-agent systems, complex networks, and hybrid control systems.

Vijay Gupta is the Elmore Professor of Electrical and Computer Engineering at Purdue University. He received his B. Tech degree at Indian Institute of Technology, Delhi, and his M.S. and Ph.D. at California Institute of Technology, all in Electrical Engineering. He received the 2018 Antonio J Rubert Award from the IEEE Control Systems Society, the 2013 Donald P. Eckman Award from the American Automatic Control Council and a 2009 National Science Foundation (NSF) CAREER Award. His research and teaching interests are broadly in the interface of communication, control, distributed computation, and human decision making.