Actuator Placement under Structural Controllability using Forward and Reverse Greedy Algorithms

Baiwei Guo*, Orcun Karaca*, Tyler Summers, Maryam Kamgarpour

Abstract—Actuator placement is an active field of research which has received significant attention for its applications in complex dynamical networks. In this paper, we study the problem of finding a set of actuator placements minimizing the metric that measures the average energy consumed for state transfer by the controller, while satisfying a structural controllability requirement and a cardinality constraint on the number of actuators allowed. As no computationally efficient methods are known to solve such combinatorial set function optimization problems, two greedy algorithms, forward and reverse, are proposed to obtain approximate solutions. We first show that the constraint sets these algorithms explore can be characterized by matroids. We then obtain performance guarantees for the forward and reverse greedy algorithms applied to the general class of matroid optimization problems by exploiting properties of the objective function such as the submodularity ratio and the curvature. Finally, we propose feasibility check methods for both algorithms based on maximum flow problems on certain auxiliary graphs originating from the network graph. Our results are verified with case studies over large networks.

I. INTRODUCTION

MANY large-scale complex dynamical networks, such as those arising in power grids [1], biological networks [2] and industrial systems [3] necessitate a resilient and efficient operation under dynamic and uncertain environments. Hence, there has been a surge of interest to study controller design in such large-scale networks [4]–[13]. A fundamental design problem is that of actuator placement in which the goal is to select a subset from a finite set of possible placements for actuators to optimize a desired network performance metric.

Variants of the actuator placement problem have been shown to be NP-hard in general, see [7], [9], [14]. Thus, it is desirable to obtain scalable algorithms with provable suboptimality bounds. Earlier studies have adopted the forward greedy algorithm. This algorithm extends the actuator set with the most beneficial actuator iteratively to derive an approximate solution [1]. Under a submodular network performance metric and a cardinality constraint on the number of actuators, the forward greedy algorithm is shown to enjoy a provable performance guarantee [15]. However, some metrics do not exhibit submodularity including the metric in this work, that is, the average energy required to reach any arbitrary direction

of the state space [16]. To alleviate this issue, submodularity has been extended to weak submodularity using the notion of submodularity ratio, quantifying how close a function is to being submodular [17], [18]. Given this ratio, it is possible to derive a performance guarantee for the forward greedy algorithm applied to a larger class of performance metrics [16].

Nonetheless, the guarantees above are restricted to problems subject to simple cardinality constraints. Given a cardinality constraint, the resulting actuator set might not be capable of moving the system over the entire state space, that is, might not render the system controllable. To address this issue, we need to include controllability as a constraint. However, to the best of our knowledge, there is no approach to quantify the forward greedy algorithm's performance with a nonsubmodular metric and a controllability constraint, nor to ensure feasibility of the iterates of the greedy algorithm in such problems. On the other hand, structural controllability constraints have been well-studied. This controllability concept exploits only the graphical interconnection structure of the dynamical system [5], [19]–[22]. Structurally controllable systems are those controllable after a slight perturbation of the system parameters corresponding to the fixed set of edges in the underlying network graph. The authors in [23] have studied a leader selection problem to obtain a structurally controllable system while minimizing a submodular objective function. The structural controllability constraint arising in the leader selection problem is proven to give rise to a matroid constraint enabling the application of the forward greedy algorithm [23]. However, the leader selection problem is different from the actuator placement problem. The former selects a set of leader nodes whose states can arbitrarily be dictated to steer the remaining nodes to desired states, while the latter does not permit the states to be dictated arbitrarily; instead, it selects a set of actuators which can influence all of the states through the dynamics. Hence, this paper pays special attention to formulating the structural controllability constraints of the actuator placement problem as a matroid constraint by proving the equivalence of this concept in both the leader selection problems and the actuator placement problems.

Given a matroid, [24] derives a performance guarantee for the forward greedy algorithm when optimizing a submodular objective. However, past work has not successfully derived performance guarantees for optimizing weakly submodular objective functions, such as the aforementioned average energy consumption metric, subject to a matroid. The first goal is to obtain a guarantee for this setting. In Appendix A, we discuss relevant existing guarantees from [15], [18], [24]–[28].

An inherent drawback of the forward greedy algorithm is

that any performance guarantee has to involve the objective function evaluated at the empty set as the reference value, since the actuator set expands starting from the empty set. This reference value is in general large for the average energy consumption metric, or even infinite [16], and it plays a great role towards the tightness of the guarantee. In addition, many works have reported the lack of ability of the forward greedy to correct errors made in earlier steps [29], [30]. An alternative is to adopt the reverse greedy, which excludes the least beneficial actuator iteratively starting from the full set. In this case, any potential performance guarantee would instead involve the objective function evaluated at the full set, which is in general small for the performance metric considered in this work.

Among the applications of the reverse greedy algorithm, [31] studied the special setting of metric k-median problem and this algorithm is shown to have a better performance than the forward greedy algorithm. The work of [32] provides a guarantee for minimizing a supermodular decreasing function under cardinality constraints by exploiting a notion of function steepness, while [33] extends this analysis to account for comatroid constraints.[1] Our paper in [34] provides a counterexample to the performance guarantee obtained in [33], and explains where the mistake originates from in their proof. Nevertheless, none of the problem settings can generalize the problem of actuator placement considered in this work. This is because, in addition to involving matroid constraints, via a reformulation, the objective function of our problem will be shown to exhibit weak supermodularity, which will be characterized by the notion of curvature [18], [28]. To the best of our knowledge, there is no performance guarantee for the reverse greedy algorithm applicable to optimizing weakly submodular and weakly supermodular objective functions (defined by submodularity ratio and curvature, respectively) subject to matroid constraints.

Our main contributions are as follows.

(i) We show that the minimization of the average energy consumption metric under structural controllability constraints can be reformulated as the maximization of a strictly increasing weakly submodular function subject to matroid constraints, see Lemma 2, Proposition 1, and Problem (5).

(ii) We obtain a performance guarantee for the forward greedy algorithm applied to this general class of matroid optimization problems, see Theorem 1.[2]

(iii) We show that the actuator placement problem has another reformulation as the minimization of a strictly increasing, weakly submodular, and weakly supermodular function subject to matroid constraints and a cardinality lower bound, see Lemma 3, Proposition 3, and Problem (11). This reformulation allows us to implement the reverse greedy algorithm.

(iv) For the reverse greedy algorithm, we obtain a performance guarantee employing both notions, see Theorem 2.

(v) The average energy consumption metric is well-defined only if we introduce a metric-modifying parameter [14]. To

[1]Comatroid is the complementary notion of a matroid, see [33].

[2]Theorems 1 and 2 could be of independent interest for researchers working on greedy algorithms. Preliminary results concerning the forward greedy— (i) and (ii) above—were presented in a conference paper in [35]. This paper significantly extends that work by contributions (iii) to (vii), and utilizes the newly introduced greedy notions of the curvature and the submodularity ratio.

this end, we design an algorithm with a provable performance to pick such parameters, see Proposition 6 and Algorithm 3.

(vi) For both algorithms, we show that the matroid feasibility checks for the actuator placement can be done efficiently by translating them into maximum flow problems over certain auxiliary graphs, see Propositions 7, 8. These results extend [5] which associates structural controllability with the existence of a perfect matching. We also provide a counterexample to a feasibility check in [23] for the leader selection problem.

Finally, we provide numerical case studies with models based on randomly generated networks and a large power grid. As an additional insight, we demonstrate that the forward greedy algorithm tends to pick higher degree actuators when compared to the optimal and the reverse greedy solutions.

In the remainder, Section II introduces the problem formulation and preliminaries. Sections III and IV apply the forward and the reverse greedy algorithms, respectively, and obtain guarantees. Section V proposes a method to pick a metric-modifying parameter and feasibility check methods for greedy algorithms. Numerical studies are presented in Section VI.

II. PROBLEM FORMULATION AND PRELIMINARIES

A. Problem formulation

Consider a linear system with state vector $x \in \mathbb{R}^n$. To each state variable $x_i \in \mathbb{R}$, we associate a node $v_i \in V := \{v_1, \ldots, v_n\}$. A control input $u_i \in \mathbb{R}$ can be exerted at each node v_i. Given a set $S \subset V$ chosen as the actuator set, the system dynamics can be written as

$$\dot{x} = Ax + B(S)u. \tag{1}$$

Above, $B(S) = \text{diag}(\mathbf{1}(S)) \in \mathbb{R}^{n \times n}$, where $\mathbf{1}(S)$ denotes a vector of size n whose ith entry is 1 if v_i belongs to S and 0 otherwise. Let $G = (V, E)$ denote a directed graph relating to system (1) with nodes V and edges E, where the edge $(v_j, v_i) \in E$ if $(A)_{ij} \neq 0$. Similar to several previous studies on structural controllability, e.g., [10], [23], throughout the paper, we assume that G is strongly connected, which will be discussed in Section V-B.

The pair $(A, B(S))$ is called *controllable* if for all $x_0, x_1 \in \mathbb{R}^n$ and $T > 0$ there exists a control input $u : [0, T] \to \mathbb{R}^n$ that steers the system from x_0 at $t = 0$ to x_1 at $t = T$. For linear time-invariant systems, controllability can be verified by the rank of the controllability matrix $P = \begin{bmatrix} B(S) & AB(S) & \cdots & A^{n-1}B(S) \end{bmatrix} \in \mathbb{R}^{n \times n^2}$. However, the entries in A are generally not exactly known but only approximately determined with small errors using system identification techniques. Moreover, when dealing with large-scale networked systems, it is often the case that we can only rely on the topology but not on the particular weights [11]. Motivated by these particularities, we consider structural controllability.

Definition 1: (A, B) and $(\hat{A}, \hat{B})$ with $A, B, \hat{A}, \hat{B} \in \mathbb{R}^{n \times n}$ are said to have the same structure if matrices $[A \ B]$ and $[\hat{A} \ \hat{B}]$ have zeros at the same entries. Given $S \subset V$, $(A, B(S))$ is *structurally controllable* if there exists a controllable pair $(\hat{A}, \hat{B})$ having the same structure as $(A, B(S))$.

As it turns out, structural controllability is a generic property, that is, the pair $(A, B(S))$ is structurally controllable if and only if almost all of the pairs with the same structure

are controllable [21]. This implies that whenever $(A, B(S))$ is not controllable but structurally controllable, it is possible to slightly perturb the entries to ensure controllability [19]. Observe that structural controllability depends on the positions of the nonzero entries. Later, this will allow us to determine this property by the graph G relating to the system.

Even if a system is controllable, an unacceptably large amount of energy might be needed to reach a desired state. Specifically, the work in [6] shows that if the number of actuators is kept constant, then certain controllable systems are practically uncontrollable since the energy consumption grows at least exponentially with the number of states n. Hence, it is crucial to minimize this energy consumption. The minimum energy required to steer the system from zero at $t = 0$ to $x \in \mathbb{R}^n$ at $t = T$ is given by $x^\top W_T^{-1}(S)x$, where $W_T(S) = \int_0^T e^{A\tau} B(S) B^\top(S) e^{A^\top \tau} d\tau$ is the controllability Gramian. To obtain an expression independent of the initial state x, calculate the average energy required over the unit sphere, $\|x\|_2 = 1$, as $F(S) := \mathrm{tr}(W_T^{-1}(S))$. This expression is well-defined only when the set S renders the system controllable. Inspired by [14], we introduce a small positive number $\epsilon \in \mathbb{R}_+$ to handle uncontrollable actuator sets and propose the metric $F_\epsilon : 2^V \to \mathbb{R}_+$,

$$F_\epsilon(S) = \mathrm{tr}((W_T(S) + \epsilon I)^{-1}), \ \forall S \subset V. \quad (2)$$

In Section V-A, we discuss the choice of ϵ.

To make a system easier to control, we seek a set $S \subset V$ minimizing the metric above. Since in a large-scale network, the number of actuators allowed is in general limited, we consider a cardinality bound of $K \in \mathbb{N}$ on the actuators. Additionally, we require that the actuators render the system structurally controllable. Our main problem is formulated as

$$\begin{aligned} \min_{S \subset V} \quad & F_\epsilon(S) \\ \text{s.t.} \quad & |S| \le K, \ (A, B(S)) \text{ is structurally controllable.} \end{aligned} \quad (3)$$

Assume K is large enough to ensure feasibility. In Section V-B, we discuss how to determine the smallest K. Problem (3) is a combinatorial optimization, and to the best of our knowledge, no computationally feasible solution method has ever been proposed. Existing works have either studied additive/modular objectives [8], [11], [36], [37] (e.g., actuator installation costs, minimizing K directly), or included only cardinality constraints [1], [16]. Notice that neither our objective is additive nor we have only cardinality constraints. Later, we will adopt efficient heuristics to derive solutions.

B. Preliminaries

We first introduce widely adopted notions for the properties of set functions and set constraints.

1) Properties of set functions: Given a ground set V and a set function $f : 2^V \to \mathbb{R}$, we say f is *(strictly) increasing* if $f(S_1) \le (<) f(S_2)$ for any $S_1 \subsetneq S_2 \subset V$. If $-f$ is (strictly) increasing, we say f *(strictly) decreasing*. For an increasing set function, the marginal gain from the addition of a certain element $v \in V$ to a set $S \subset V$ varies for different S. For many set functions in practical problems the marginal gain diminishes as S expands, see the examples in [38], [39].

Submodularity describes this property and submodularity ratio describes how far a nonsubmodular function is from being submodular. For the following, denote the marginal gains by $\rho_U(S) := f(S \cup U) - f(S), \ \forall S, U \subset V$. For notational simplicity, we use v and $\{v\}$ interchangeably for singleton sets.

Definition 2: For an increasing function $f : 2^V \to \mathbb{R}$, the *submodularity ratio* is the largest $\gamma \in \mathbb{R}_+$ such that $\gamma \rho_v(S \cup U) \le \rho_v(S), \ \forall S, U \subset V, \ \forall v \in V \setminus (S \cup U)$. It can be verified that $\gamma \in [0, 1]$. A set function f with submodularity ratio γ is called γ-submodular. A γ-submodular set function is said to be *submodular* if $\gamma = 1$ and *weakly submodular* if $0 < \gamma < 1$.

In Appendix B, we connect Definition 2 with another existing notion of submodularity ratio and discuss the necessity of introducing this notion as per Definition 2 for the guarantee derived for the forward greedy algorithm in Section III.

Other than submodularity, another widely-used notion is supermodularity, that is, the marginal gain from the addition of $v \notin S$ to the set S increases as S expands. By introducing supermodularity and the curvature, that is, how far a non-supermodular function is from being supermodular, we obtain a more precise description on how the marginal gains change.

Definition 3: For an increasing function $f : 2^V \to \mathbb{R}$, the *curvature* is the smallest $\alpha \in \mathbb{R}_+$ such that $\rho_v(S \cup U) \ge (1 - \alpha)\rho_v(S), \ \forall S, U \subset V, \ \forall v \in V \setminus (S \cup U)$. It can be verified that $\alpha \in [0, 1]$. Function f with curvature α is called α-supermodular. An α-supermodular function is *supermodular* if $\alpha = 0$ and *weakly supermodular* if $0 < \alpha < 1$.

To see how submodularity ratio and curvature are related, notice that for an increasing set function f the submodularity ratio γ and the curvature α satisfy

$$\gamma = \min_{\substack{S, U, \\ v \in V \setminus (S \cup U)}} \frac{\rho_v(S)}{\rho_v(S \cup U)} \le \max_{\substack{S, U, \\ v \in V \setminus (S \cup U)}} \frac{\rho_v(S)}{\rho_v(S \cup U)} = \frac{1}{1 - \alpha}. \quad (4)$$

2) Properties of set constraints: Many combinatorial optimization problems from the literature are subject to constraints that are more complex than simple cardinality constraints, see the examples in [40], [41]. Among those, we introduce matroids since they will generalize reformulations of the constraints found in Problem (3), and they allow performance guarantees for greedy algorithms [42].

Definition 4: A *matroid* $\mathcal{M}$ is an ordered pair $(V, \mathcal{F})$ consisting of a ground set V and a collection $\mathcal{F}$ of subsets of V which satisfies (i) $\emptyset \in \mathcal{F}$, (ii) if $S, S' \in \mathcal{F}$ and $S' \subset S$, then $S' \in \mathcal{F}$, (iii) if $S_1, S_2 \in \mathcal{F}$ and $|S_1| < |S_2|$, there exists $v \in S_2 \setminus S_1$ such that $v \cup S_1 \in \mathcal{F}$. Every set in $\mathcal{F}$ is called *independent*, and maximum independent sets refer to those with the largest cardinality.

To adopt the reverse greedy algorithm, an additional concept will be required, that is, the dual of a matroid.

Definition 5: Given a matroid $(V, \mathcal{F})$, let $\mathcal{F}^* = \{U \mid \exists$ a maximum independent set $M \in \mathcal{F}$ such that $U \subset V \setminus M\}$. The pair $(V, \mathcal{F}^*)$ is the *dual* of the matroid $(V, \mathcal{F})$.

We characterize its structure in the following lemma.

Lemma 1: The pair $(V, \mathcal{F}^*)$, the dual of a matroid $(V, \mathcal{F})$, is also a matroid.

Proof: Suppose $\{M_i\}_{i=1}^q$ is the collection of all maximum independent sets in matroid $(V, \mathcal{F})$. From [43, Ch. 2] we

have that $\{V \setminus M_i\}_{i=1}^q$ defines a collection of all maximum independent sets for another matroid denoted by $(V, \tilde{\mathcal{F}})$. In the following, we prove that $\mathcal{F}^* = \tilde{\mathcal{F}}$. For any $U \in \mathcal{F}^*$, there exists M, a maximum independent set in $\mathcal{F}$, such that $U \subset V \setminus M$. Since $V \setminus M \in \tilde{\mathcal{F}}$ and $(V, \tilde{\mathcal{F}})$ is a matroid, the set U also belongs to $\tilde{\mathcal{F}}$ from property (ii) in Definition 4. Conversely, if $U \in \tilde{\mathcal{F}}$, according to property (iii) in Definition 4, U is a subset of some maximum independent set in $\tilde{\mathcal{F}}$. Consequently, there exists a maximum independent set $M \in \mathcal{F}$ such that $U \subset V \setminus M$. Thus, $U \in \mathcal{F}^*$. This concludes that $\mathcal{F}^* = \tilde{\mathcal{F}}$ and thus $(V, \mathcal{F}^*)$ is also a matroid. ∎

[43, Ch. 2] defines the dual concept as $(V, \tilde{\mathcal{F}})$, and the proof above verifies that $(V, \mathcal{F}^*)$ we have in Definition 5 is an equivalent reformulation. This reformulation will help us present the proof of Proposition 3 in a more clear way.

III. FORWARD GREEDY ALGORITHM

In the following, we reformulate Problem (3) as the maximization of a strictly increasing weakly submodular function subject to matroid constraints. We then obtain a guarantee for a forward greedy algorithm over matroid constraints.

A. Properties of the objective

Intuitively, with more input nodes, system (1) would be easier to control and thus the metric F_ϵ in (2) would be smaller. This intuition can be readily verified as follows.

Lemma 2: The metric $F_\epsilon = \mathrm{tr}((W_T(S) + \epsilon I)^{-1})$ satisfies the following statements: (i) F_ϵ is strictly decreasing, (ii) $-F_\epsilon$ is weakly submodular with submodularity ratio $\gamma_\epsilon^{\mathsf{f}}$.

The proof is relegated to Appendix C. Together with the fact that the structural controllability is preserved under actuator set expansion, Lemma 2 implies that the optimal solution to Problem (3) should contain exactly K nodes.

B. Reformulation of the constraint set

In combinatorial optimization problems with only cardinality constraints, the forward greedy algorithm starts from the empty set and at t^{th} iteration, adds the most marginally beneficial node v_t^{f} to the actuator set. It terminates when the cardinality of the actuator set is K. When applied to Problem (3), this method might return an actuator set under which the system is not structurally controllable. To this end, we need to restrict the greedy iterates $S^t = \{v_1^{\mathsf{f}}, \ldots, v_t^{\mathsf{f}}\}$, for $t = 1, \ldots, K$, such that the set S^K returned by the forward greedy algorithm is guaranteed to satisfy structural controllability.

Since the optimal solution to Problem (3) contains exactly K nodes, we define $\mathcal{C}_K = \{S \subset V \,|\, |S| = K$ and the system is structurally controllable under $S\}$ and rewrite Problem (3) as the minimization of F_ϵ over the set collection $\mathcal{C}_K$. In the procedure of the forward greedy algorithm, the set S^t has to be a subset of some set in $\mathcal{C}_K$, since otherwise the greedy solution S^K would not belong to $\mathcal{C}_K$. Thus, define $\tilde{\mathcal{C}}_K = \{\Omega \,|\, \exists S \in \mathcal{C}_K$ such that $\Omega \subset S\}$ and reformulate (3) as

$$\max_{S \subset V} \; -F_\epsilon(S) \quad \text{s.t.} \; S \in \tilde{\mathcal{C}}_K. \tag{5}$$

The strict monotonicity of $-F_\epsilon$ ensures that the optimal solution to Problem (5) coincides with that of Problem (3).

As such, we consider solving Problem (5) as an equivalent characterization of Problem (3).

Next, we show that the feasible region of Problem (5) characterizes a matroid, which will allow us to derive performance guarantees for the greedy solution S^K.

Proposition 1: $\mathcal{M} = (V, \tilde{\mathcal{C}}_K)$ is a matroid.

To prove this, we establish the equivalence between structural controllability of $(A, B(S))$ in Problem (3) and structural controllability of the system with the set S chosen as a leader set in a corresponding leader selection problem. We then invoke a result from [23] proving the matroid structure of the structural controllability constraints in leader selection problems. For the details and additional discussions, we kindly refer to the proof in Appendix D. This equivalence result will later be utilized in Sections V-B and V-C to bring in results from the leader selection literature.

We now restrict the iterates of the forward greedy algorithm to lie in the set collection $\tilde{\mathcal{C}}_K$. As a remark, given $S^t \in \tilde{\mathcal{C}}_K$ for $t < K$, by Definition 4, we can always find a node $v \in V \setminus S^t$ such that $S^t \cup v \in \tilde{\mathcal{C}}_K$, as long as $\mathcal{C}_K \neq \emptyset$. Therefore, it is guaranteed that at iteration K we obtain an actuator set in $\mathcal{C}_K$.

C. Performance guarantee

In the previous section, we showed that the objective function $-F_\epsilon$ is $\gamma_\epsilon^{\mathsf{f}}$-submodular in Lemma 2 and the feasible region $\tilde{\mathcal{C}}_K$ characterizes a matroid in Proposition 1. Thus, Problem (5) falls into the following class of optimization problems:

$$\max_{S \subset V} \quad f(S), \text{strictly increasing and } \gamma\text{-submodular} \tag{6}$$
$$\text{s.t.} \quad S \in \mathcal{F}, \text{ where } \mathcal{M} = (V, \mathcal{F}) \text{ is a matroid,}$$

where the cardinality of any maximum independent set in $\mathcal{F}$ is K. Let S^* denote its optimal solution.

The forward greedy over a matroid was first introduced in [24] for submodular objectives. This algorithm is presented in Algorithm 1. At the tth iteration, we check the feasibility of the node with the largest marginal gain in $V \setminus S^{t-1}$. If the actuator set obtained by adding this node to S^{t-1} does not belong to $\mathcal{F}$, we exclude the node from consideration. Among the remaining ones, we check the feasibility of the node with the largest marginal gain until a feasible node v_t^{f} is found. Then $S^t = \{v_t^{\mathsf{f}}\} \cup S^{t-1}$ is the actuator set returned by the tth iteration. The final actuator set is $S^{\mathsf{f}} := S^K$. The feasibility check ensures that $S^t \in \mathcal{F}$ and hence S^{f} belongs to $\mathcal{F}$.

We use $U^t \subset V$ for $0 \leq t \leq K - 1$ to denote all the nodes having been considered by the feasibility check before v_{t+1}^{f}. We define the marginal gains as $\rho_t = f(S^{t+1}) - f(S^t)$.

Using the matroid structure and the submodularity ratio, we can state our first main result as follows.

Theorem 1: If Algorithm 1 is applied to Problem (6), then

$$\frac{f(S^{\mathsf{f}}) - f(\emptyset)}{f(S^*) - f(\emptyset)} \geq \frac{\gamma^3}{\gamma^3 + 1}. \tag{7}$$

The proof is relegated to Appendix E. The idea of the proof extends the work in [24], which derives a performance guarantee for matroid optimization featuring a submodular objective. When $\gamma = 1$, the guarantee in (7) coincides with that of [24], derived for a submodular f. As a remark, for

Algorithm 1 Forward Greedy Algorithm over Matroid

Input: set function f, ground set V and matroid $(V, \mathcal{F})$
Output: actuator set S^{f}
  function FORWARDOVERMATROID($f, V, \mathcal{F}$)
    $S^0 = \emptyset$, $U^0 = \emptyset$, $t = 1$
    while $U^{t-1} \neq V$ and $|S^{t-1}| < K$ **do**
      $i^*(t) = \arg\max_{i \in V \setminus U^{t-1}} \rho_i(S^{t-1})$
      if $S^{t-1} \cup \{i^*(t)\} \notin \mathcal{F}$ **then**
        $U^{t-1} \leftarrow U^{t-1} \cup \{i^*(t)\}$
      else
        $\rho_{t-1} \leftarrow \rho_{i^*(t)}(S^{t-1})$ and $v_t^{\mathsf{f}} = i^*(t)$
        $S^t \leftarrow S^{t-1} \cup \{v_t^{\mathsf{f}}\}$ and $U^t \leftarrow U^{t-1} \cup \{v_t^{\mathsf{f}}\}$
        $t \leftarrow t + 1$
      end if
    end while
    $S^{\mathsf{f}} \leftarrow S^{t-1}$
  end function

Problem (6), another performance guarantee is offered by [26] but in expectation for a randomized algorithm. We refer to Appendix B for a comparison of these two guarantees.[3]

Given any function f, it is difficult to derive its submodularity ratio because the computation in Definition 2 involves $\Omega(2^n)$ inequalities. In the proof of Theorem 1, only a subset of these inequalities are utilized. Via this observation, the following corollary proposes a computationally more efficient approach.

Corollary 1: Let γ^{fg} be the largest $\hat{\gamma}$ that satisfies (a) $f(S \cup S^{\mathsf{f}}) - f(S^{\mathsf{f}}) \leq \hat{\gamma}^{-1} \sum_{j \in S \setminus S^{\mathsf{f}}} \rho_j(S^{\mathsf{f}})$ for any S with $|S| = K$, (b) $\rho_j(S^{\mathsf{f}}) \leq \hat{\gamma}^{-1} \rho_j(S^{t-1}), \forall t \leq K, \forall j \in V$, (c) $f(S^{i_2+1}) - f(S^{i_2}) \leq \hat{\gamma}^{-1}(f(S^{i_1} \cup \{v_{i_2+1}\}) - f(S^{i_1}))$, for any $i_1 < i_2$. Then, γ^{fg} is called the greedy submodularity ratio for the forward greedy algorithm, with $\gamma^{\mathsf{fg}} \geq \gamma$, and

$$\frac{f(S^{\mathsf{f}}) - f(\emptyset)}{f(S^*) - f(\emptyset)} \geq \frac{(\gamma^{\mathsf{fg}})^3}{(\gamma^{\mathsf{fg}})^3 + 1}. \tag{8}$$

The greedy submodularity ratio can be obtained after the forward greedy algorithm is completed by analyzing $\mathcal{O}\left(\binom{n}{K}\right)$ inequalities. Since $\gamma^{\mathsf{fg}} \geq \gamma$, the performance guarantee in (8) is better than (7). Notice that γ^{fg} changes with the constraint set of the problem since the inequalities defining γ^{fg} would then be different. In contrast, submodularity ratio γ depends only on the objective function.

Next, we substitute $f = -F_\epsilon$ and $\mathcal{F} = \tilde{\mathcal{C}}_K$ into the performance guarantee (8) of the general setting (6).

Corollary 2: Suppose we apply Algorithm 1 to Problem (5). Denote the actuator set returned as S_ϵ^{f} and the greedy submodularity ratio of $-F_\epsilon$ as $\gamma_\epsilon^{\mathsf{fg}}$. Then, S_ϵ^{f} satisfies

$$\frac{F_\epsilon(\emptyset) - F_\epsilon(S_\epsilon^{\mathsf{f}})}{F_\epsilon(\emptyset) - F_\epsilon(S^*)} \geq \frac{(\gamma_\epsilon^{\mathsf{fg}})^3}{(\gamma_\epsilon^{\mathsf{fg}})^3 + 1}. \tag{9}$$

Since the forward greedy algorithm starts expanding from the empty set, performance guarantees can only assess $f(S^{\mathsf{f}})$ by considering $f(\emptyset)$ as the reference. If $f(\emptyset) = 0$, the performance guarantee (8) is reduced to $f(S^{\mathsf{f}})/f(S^*) \geq$

---
[3] The works in [25] and [18] utilize also the curvature to derive performance guarantees for the forward greedy applied to cardinality constrained problems. Exploiting this notion for matroid constraints is part of our ongoing work.

$\gamma^3/(1 + \gamma^3)$. In this case, we only lose a fraction of the optimal objective by adopting the forward greedy algorithm. However, for our actuator placement problem $F_\epsilon(\emptyset) = n\epsilon^{-1}$, and the performance guarantee (9) is equivalent to

$$F_\epsilon(S_\epsilon^{\mathsf{f}}) \leq \frac{1}{(\gamma_\epsilon^{\mathsf{fg}})^3 + 1} F_\epsilon(\emptyset) + \frac{(\gamma_\epsilon^{\mathsf{fg}})^3}{(\gamma_\epsilon^{\mathsf{fg}})^3 + 1} F_\epsilon(S^*). \tag{10}$$

Since ϵ is a small positive number and n is in general large, the guarantee above can be loose.[4] In the next section, we consider a variant of the greedy algorithm that comes along with a performance guarantee that does not depend on $F_\epsilon(\emptyset)$.

IV. REVERSE GREEDY ALGORITHM

To derive an alternative guarantee, we consider the reverse greedy algorithm (also called the stingy or greedy descent). This algorithm starts from the full set, and at each iteration, excludes the node with the least marginal gain from the actuator set of the previous iteration until a solution is reached. Such an approach allows to have the reference as $F_\epsilon(V)$, which is significantly smaller than $F_\epsilon(\emptyset)$ in practice.

A. Properties of the objective

For the reverse greedy algorithm, we reformulate our metric as $F_\epsilon^{\mathsf{r}}(R) := F_\epsilon(V \setminus R)$, for all $R \subset V$. The following lemma characterizes the properties of this function.

Lemma 3: The set function F_ϵ^{r} is strictly increasing, weakly submodular with submodularity ratio $\gamma_\epsilon^{\mathsf{r}} > 0$ and weakly supermodular with curvature $\alpha_\epsilon^{\mathsf{r}} < 1$.

Proof: Regarding the strict monotonicity, suppose $S_1 \subsetneq S_2$. Since F_ϵ is strictly decreasing and $V \setminus S_2 \subsetneq V \setminus S_1$, $F_\epsilon(V \setminus S_2) > F_\epsilon(V \setminus S_1)$, which implies $F_\epsilon^{\mathsf{r}}(S_2) > F_\epsilon^{\mathsf{r}}(S_1)$. Due to strict monotonicity of F_ϵ^{r}, it follows readily from the equalities shown in (4) that the submodularity ratio is strictly greater than 0 and the curvature is strictly less than 1. Thus, F_ϵ^{r} is weakly submodular with $\gamma_\epsilon^{\mathsf{r}} > 0$ and weakly supermodular with $\alpha_\epsilon^{\mathsf{r}} < 1$. ∎

Recall that the submodularity ratio of $-F_\epsilon$ is $\gamma_\epsilon^{\mathsf{f}}$. Now denote its curvature as $\alpha_\epsilon^{\mathsf{f}}$, which can easily be shown to satisfy $\alpha_\epsilon^{\mathsf{f}} < 1$. The following connects $(\gamma_\epsilon^{\mathsf{f}}, \alpha_\epsilon^{\mathsf{f}})$ and $(\gamma_\epsilon^{\mathsf{r}}, \alpha_\epsilon^{\mathsf{r}})$.

Proposition 2: $\gamma_\epsilon^{\mathsf{r}} = 1 - \alpha_\epsilon^{\mathsf{f}}$ and $\alpha_\epsilon^{\mathsf{r}} = 1 - \gamma_\epsilon^{\mathsf{f}}$.

The proof is relegated to Appendix F. The proposition above provides an insight into how the submodularity ratio and the curvature of F_ϵ^{r} relates to those of $-F_\epsilon$.

B. Reformulation of the constraint set

The reverse greedy algorithm has to return an exclusion set R^{r} such that the resulting actuator set $V \setminus R^{\mathsf{r}}$ contains K nodes, and renders the system structurally controllable, that is, $V \setminus R^{\mathsf{r}} \in \mathcal{C}_K$. We collect all such exclusion sets and form $\mathcal{R}_K = \{R \mid V \setminus R \in \mathcal{C}_K\}$. Suppose after the tth node exclusion of the reverse greedy algorithm, all the nodes excluded form a set $R^t = \{r_1, \dots, r_t\}$, where $r_i \in V$ for all i. The set R^t has to be a subset of some set in $\mathcal{R}_K$ for any $t = 1, \dots, N$, where $N := n - K$, since otherwise when N exclusions are

---
[4] If there exists an initial actuator set $S^{\text{ini}} \neq \emptyset$ rendering the system controllable, we can potentially mitigate this issue, since the reference of the guarantee would then be given by $F_\epsilon(S^{\text{ini}})$. Clearly, such applications also allow to set $\epsilon = 0$, and drop structural controllability constraints.

completed, the resulting actuator set would not belong to $\mathcal{C}_K$. Thus, define $\tilde{\mathcal{R}}_K := \{Q \,|\, \exists R \in \mathcal{R}_K \text{ such that } Q \subset R\}$, and reformulate Problem (3) as

$$\min_{R \subset V} \; F_\epsilon^{\mathrm{r}}(R) \;\; \text{s.t. } R \in \tilde{\mathcal{R}}_K \text{ and } |R| = N. \qquad (11)$$

The strict monotonicity of $F_\epsilon^{\mathrm{r}}(R)$ again ensures that the optimal solution to Problem (11) coincides with that of Problem (3). Note that cardinality constraint in (11) can equivalently be replaced with an inequality constraint $|R| \geq N$.

Next, we show that $\tilde{\mathcal{R}}_K$ characterizes a matroid.

Proposition 3: $\mathcal{M}^{\mathrm{r}} = (V, \tilde{\mathcal{R}}_K)$ is a matroid.

Proof: We prove that $(V, \tilde{\mathcal{R}}_K)$ is the dual of $(V, \tilde{\mathcal{C}}_K)$, that is, $\tilde{\mathcal{R}}_K = \tilde{\mathcal{C}}_K^*$. Note that we can then invoke Lemma 1 showing that the dual of a matroid is also a matroid. For any $Q \in \tilde{\mathcal{R}}_K$, according to the definition of $\tilde{\mathcal{R}}_K$, there exists $R \in \mathcal{R}_K$ such that $Q \subset R$. We have $S = V \setminus R \in \mathcal{C}_K$ for $R \in \mathcal{R}_K$. Since $|S| = K$, S is a maximum independent set in $\tilde{\mathcal{C}}_K$. Considering $Q \subset V \setminus S$, we conclude $Q \in \tilde{\mathcal{C}}_K^*$. Conversely, for any $Q \in \tilde{\mathcal{C}}_K^*$, there exists a maximum independent set $S \in \tilde{\mathcal{C}}_K$ such that $Q \subset V \setminus S$. From the definition of $\tilde{\mathcal{C}}_K$, we know that $S \in \mathcal{C}_K$. Thus, we obtain $Q \in \tilde{\mathcal{R}}_K$. This concludes the equivalence of $\tilde{\mathcal{R}}_K$ and $\tilde{\mathcal{C}}_K^*$. ∎

Similar to the discussions in Section III, by restricting the iterates of the reverse greedy algorithm to lie in the set collection $\tilde{\mathcal{R}}_K$, we obtain a final exclusion set in $\mathcal{R}_K$ with cardinality N. This implies that the final actuator set lies in $\mathcal{C}_K$.

C. Performance guarantee

In the previous section, we showed that the objective function F_ϵ^{r} is $\gamma_\epsilon^{\mathrm{r}}$-submodular and $\alpha_\epsilon^{\mathrm{r}}$-supermodular in Lemma 3, and the feasible region of $\tilde{\mathcal{R}}_K$ characterizes a matroid in Proposition 3. Thus, Problem (11) falls into the following class of optimization problems:

$$\min_{R \subset V} \quad f(R), \text{ strictly increasing, } \gamma\text{-submodular,}$$
$$\text{and } \alpha\text{-supermodular} \qquad (12)$$
$$\text{s.t.} \quad R \in \mathcal{F}, \; \mathcal{M} = (V, \mathcal{F}) \text{ is a matroid}, |R| \geq N,$$

where the cardinality of maximum independent sets in $\mathcal{F}$ is N.[5] Let R^* denote its optimal solution. Clearly, R^* is the set complement of S^*, that is, $R^* = V \setminus S^*$.

Define set function f° such that $f^{\circ}(R) = f(V \setminus R)$ for all R. In Problem (11), f° corresponds to F_ϵ. Observe that the forward greedy algorithm applied to the minimization of the function f is equivalent to the reverse greedy algorithm applied to the minimization of the function f°. This algorithm is presented in Algorithm 2. Different from Algorithm 1, at each iteration, Algorithm 2 implements the feasibility check on the node with the least marginal gain.

For Algorithm 2, the following definitions are in order. We define $\rho_j(R) := f(R \cup j) - f(R)$, $\rho_t := f(R^t) - f(R^{t-1})$ and $r_t := R^t \setminus R^{t-1}$. The set U^t denotes the set of nodes having been considered by the feasibility check before r_{t+1}. The final exclusion set is $R^{\mathrm{r}} := R^N$, and it lies in $\mathcal{R}_K$.

[5]The performance guarantee we derive in this section will be valid as long as the cardinality of maximum independent sets in $\mathcal{F}$ are larger than or equal to N, since this would ensure the feasibility of the problem.

Algorithm 2 Reverse Greedy Algorithm over Matroid

Input: set function f°, ground set V, matroid $(V, \mathcal{F})$
Output: exclusion set R^{r}
  function REVERSEOVERMATROID($f^{\circ}, V, \mathcal{F}$)
    $R^0 = \emptyset$, $U^0 = \emptyset$, $t = 1$
    while $U^{t-1} \neq V$ and $|R^{t-1}| < N$ **do**
      $j^*(t) = \arg\min_{j \in V \setminus U^{t-1}} \rho_j(R^{t-1})$
      if $R^{t-1} \cup j^*(t) \notin \mathcal{F}$ **then**
        $U^{t-1} \leftarrow U^{t-1} \cup j^*(t)$
      else
        $\rho_t \leftarrow \rho_{j^*(t)}(R^{t-1})$ and $r_t = j^*(t)$
        $R^t \leftarrow R^{t-1} \cup j^*(t)$ and $U^t \leftarrow U^{t-1} \cup j^*(t)$
        $t \leftarrow t + 1$
      end if
    end while
    $R^{\mathrm{r}} \leftarrow R^{t-1}$
  end function

A special case of Problem (12) was previously shown to be hard to approximate. Specifically, for the problem of minimizing a submodular increasing function over only a cardinality lower bound, the work in [44] shows that there is no bicriteria approximation performing better than $o(\sqrt{n/\log n})$, where n is the cardinality of the ground set.[6] Next, we extend this result by providing novel counterexamples showing that a strictly positive submodularity ratio and a curvature bounded away from 1 is indispensable to obtain any meaningful performance guarantee for Problem (12). The proofs of Propositions 4 and 5 are provided online in [45].

Proposition 4: In Problem (12), one cannot derive any upper bound on $(f(R^{\mathrm{r}}) - f(\emptyset))/(f(R^*) - f(\emptyset))$ if no strictly positive lower bound on γ is known.

Proposition 5: In Problem (12), one cannot derive any upper bound less than N on $(f(R^{\mathrm{r}}) - f(\emptyset))/(f(R^*) - f(\emptyset))$, if no upper bound less than 1 is known for α.

The propositions above conclude that we have to utilize both the submodularity ratio and the curvature. Our second main result is shown in the following theorem.

Theorem 2: If Algorithm 2 is applied to (12), then

$$\frac{f(R^{\mathrm{r}}) - f(\emptyset)}{f(R^*) - f(\emptyset)} \leq \frac{\gamma}{1 - \gamma}\left((2N+1)^{\frac{1-\gamma}{\gamma(1-\alpha)}} - 1\right). \qquad (13)$$

The proof extends the linear programming proofs utilized by [25], which considers the maximization of increasing submodular functions over matroid constraints, and by [18], which considers the maximization of increasing, nonsubmodular nonsupermodular functions over cardinality constraints. In contrast, our proof applies to the minimization of increasing, nonsubmodular nonsupermodular functions over matroids.

The main idea of the proof is to provide a series of inequalities that upperbound ρ_t by $f(R^*) - f(\emptyset)$, for each iteration t. This way, $f(R^{\mathrm{r}}) - f(\emptyset) = \sum_{t=1}^{N} \rho_t$ has an upper bound expressed by $f(R^*)$. For the following lemma, we recall that $R^t := \{r_1, \ldots, r_t\}$ is the set obtained by the greedy algorithm after the exclusion of r_t.

[6]Bicriteria approximation refers to approximating both the constraint requirement and the optimal objective. We refer to [44] for the exact description.

Lemma 4: For any $t \in \{0, \ldots, N-1\}$, ρ_t satisfies

$$f(R^*) - f(\emptyset) \geq (1 - \frac{1}{\gamma}) \sum_{i:r_i \in R^t \setminus R^*} \rho_i + \sum_{i:r_i \in R^t \cap R^*} \rho_i \quad (14)$$
$$+ (1-\alpha)(N-t)\rho_{t+1}.$$

Proof: Suppose $R^* = \{r_1^*, \ldots, r_N^*\}$. Rewrite $f(R^* \cup R^t)$ as two telescoping sums $f(R^* \cup R^t) = f(R^*) + \sum_{i=1}^{t} \rho_{r_i}(R^* \cup R^{i-1})$, and $f(R^* \cup R^t) = f(R^t) + \sum_{k=1}^{N} \rho_{r_k^*}(\{r_1^*, \ldots, r_{k-1}^*\} \cup R^t)$, which is directly obtained from the definition of ρ_r. For any i such that $r_i \in R^* \cap R^t$, we have $\rho_{r_i}(R^* \cup R^{i-1}) = 0$. Using this, and the fact that both telescoping sums above are equal to $f(R^* \cup R^t)$, we obtain

$$f(R^*) + \sum_{i:r_i \in R^t \setminus R^*} \rho_{r_i}(R^* \cup R^{i-1}) = f(R^* \cup R^t)$$
$$= f(R^t) + \sum_{k=1}^{N} \rho_{r_k^*}(\{r_1^*, \ldots, r_{k-1}^*\} \cup R^t). \quad (15)$$

Invoking the definitions of submodularity ratio and curvature, for each i such that $r_i \in R^t \setminus R^*$, we have

$$\rho_{r_i}(R^* \cup R^{i-1}) \leq \frac{1}{\gamma}\rho_i, \quad (16)$$

and for any $k \in \{1, \ldots, N\}$,

$$\rho_{r_k^*}(\{r_1^*, \ldots, r_{k-1}^*\} \cup R^t) \geq (1-\alpha)\rho_{r_k^*}(R^t). \quad (17)$$

By the definition of a matroid, there exists $R_c^t = \{r_{c_1}^*, \ldots, r_{c_{N-t}}^*\} \subset R^* \setminus R^t$ such that $R_c^t \cup R^t \in \mathcal{F}$. Consequently, for any $1 \leq i \leq N-t$, we have $R^t \cup r_{c_i}^* \in \mathcal{F}$. Thus, adding $r_{c_i}^*$ to R^t has to be feasible in the matroid. If $\rho_{r_{c_i}^*}(R^t) < \rho_{r_{t+1}}(R^t) = \rho_{t+1}$, $r_{c_i}^*$ could be added to R^t to form R^{t+1} instead of r_{t+1}. This yields a contradiction, implying that the inequality $\rho_{r_{c_i}^*}(R^t) \geq \rho_{r_{t+1}}(R^t) = \rho_{t+1}$ holds for any i. Hence, we obtain

$$\sum_{k=1}^{N} \rho_{r_k^*}(R^t) \geq \sum_{r_k^* \in R_c^t} \rho_{r_k^*}(R^t) \geq (N-t)\rho_{t+1}. \quad (18)$$

Next, by substituting (16), (17) and (18) into (15), we obtain

$$f(R^*) - f(\emptyset) + \sum_{\substack{i: \\ r_i \in R^t \setminus R^*}} \frac{\rho_i}{\gamma} \geq \sum_{i=1}^{t} \rho_i + (1-\alpha)(N-t)\rho_{t+1}. \quad (19)$$

By grouping the terms in (19), we obtain (14). ∎

Next, we can construct a linear program where the solution provides an upper bound for $\frac{f(R^r) - f(\emptyset)}{f(R^*) - f(\emptyset)}$.

Proof of Theorem 2: Let $x_i = \rho_i/(f(R^*) - f(\emptyset))$, we have $(f(R^r) - f(\emptyset))/(f(R^*) - f(\emptyset)) = \sum_{i=1}^{N} x_i$. Note that $x_i \geq 0$ for all i. Suppose $R^* \cap R^r = \{r_{i_1}, r_{i_2}, \ldots\}$. To give an upper bound for this ratio, we exploit the inequalities (14) and build the following linear programming problem to compute

the largest possible sum, $\sum_{i=1}^{N} x_i$,

$$Z(N,\gamma,\alpha) = \max \sum_{i=1}^{N} x_i, \text{ s.t. } x_i \geq 0 \text{ and}$$

$$\begin{bmatrix} (1-\alpha)N \\ -(1-\gamma)/\gamma & \begin{matrix}(1-\alpha)\\ \times(N-1)\end{matrix} \\ \vdots & \vdots & \ddots \\ -(1-\gamma)/\gamma & -(1-\gamma)/\gamma & \cdots & \begin{matrix}(1-\alpha)\\ \times(N-i_1+1)\end{matrix} \\ -(1-\gamma)/\gamma & -(1-\gamma)/\gamma & \cdots & 1 & \ddots \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ -(1-\gamma)/\gamma & -(1-\gamma)/\gamma & \cdots & 1 & \cdots & (1-\alpha) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{i_1} \\ \vdots \\ x_N \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}. \quad (20)$$

To get an upper bound for $\sum_{i=1}^{N} x_i$, we consider the following relaxed problem where the unit entries in (20) are replaced by $-(1-\gamma)/\gamma$,

$$\bar{Z}(N,\gamma,\alpha) = \max \sum_{i=1}^{N} x_i, \text{ s.t. } x_i \geq 0 \text{ and} \quad (21)$$

$$\begin{bmatrix} (1-\alpha)N \\ -(1-\gamma)/\gamma & (1-\alpha)(N-1) \\ \vdots & \vdots & \ddots \\ -(1-\gamma)/\gamma & -(1-\gamma)/\gamma & \cdots & (1-\alpha) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}.$$

Since we require that $x_i \geq 0$ for any i, any feasible solution to (21) is also feasible to (20). Thus, $\bar{Z}(N,\gamma,\alpha) \geq Z(N,\gamma,\alpha)$ for any N, γ and α. We claim that the optimum x^* of Problem (21) makes all the inequality constraints tight. This is easily seen by rewriting the inequalities as $x_t \leq \frac{1}{(1-\alpha)(N-t+1)}(1 + \frac{1-\gamma}{\gamma}\sum_{i=1}^{t-1} x_i)$, $t = 1, \ldots, N$. Notice that for submodular functions, we have $\gamma = 1$. Using the claim above and considering the fact that $x^{-1} < \ln(x + 1/2) - \ln(x - 1/2)$ for any $x \geq 1$, we can directly obtain the following guarantee

$$\frac{(f(R^r) - f(\emptyset))}{(f(R^*) - f(\emptyset))} \leq \sum_{i=1}^{N} x_i^* = \sum_{i=1}^{N} \frac{1}{i(1-\alpha)} \leq \frac{\ln(2N+1)}{1-\alpha}. \quad (22)$$

Next, we focus our efforts on the case in which $0 < \gamma < 1$ and $0 \leq \alpha < 1$. We obtain

$$\bar{Z}(N,\gamma,\alpha) = \frac{1}{b}\prod_{i=1}^{N}\left(1 + \frac{b}{(N-i+1)(1-\alpha)}\right) - \frac{1}{b}, \quad (23)$$

where $b = (1-\gamma)/\gamma$. Considering $\bar{Z}(N,\gamma,\alpha) \geq Z(N,\gamma,\alpha) \geq (f(R^r) - f(\emptyset))/(f(R^*) - f(\emptyset))$ and

$$\bar{Z}(N,\gamma,\alpha) = b^{-1}\exp\left(\sum_{i=1}^{N}\ln\left(1 + \frac{b}{(N-i+1)(1-\alpha)}\right)\right) - b^{-1}$$
$$\leq b^{-1}\exp\left(\frac{b}{(1-\alpha)}\sum_{i=1}^{N}\frac{1}{(N-i+1)}\right) - b^{-1}$$
$$\leq b^{-1}\exp\left(\frac{b}{(1-\alpha)}\left(\ln(N+\frac{1}{2}) - \ln\frac{1}{2}\right)\right) - b^{-1}$$
$$= b^{-1}(2N+1)^{\frac{b}{(1-\alpha)}} - b^{-1}. \quad (24)$$

The first equality rewrites the multiplication in $\bar{Z}$ into an exponential sum. The inequalities follow from the fact that $\ln(1 + x) < x$ for any $x > 0$ and $x^{-1} < \ln(x + 1/2) - \ln(x -$

TABLE I

COMPARISON BETWEEN PERFORMANCE GUARANTEES $\bar{Z}$ AND Z_u

(N, γ, α)	$\bar{Z}$	Z_u
(20,0.9,0.1)	4.87	5.25
(100,0.9,0.1)	7.87	8.32
(20,0.99,0.1)	4.07	4.21

1/2) for any $x \geq 1$. By substituting $b = (1 - \gamma)/\gamma$ back into the last term in (24), we get (13). ∎

Let $Z_u(N, \gamma, \alpha) = \frac{\gamma}{1-\gamma}\big((2N + 1)^{\frac{1-\gamma}{\gamma(1-\alpha)}} - 1\big)$. Table I illustrates how well the upper bound Z_u approximates the original guarantee $\bar{Z}$, stated in (23). For the supermodular case $\alpha = 0$, we obtain the guarantee $\frac{\gamma}{1-\gamma}\big((2N+1)^{\frac{1-\gamma}{\gamma}} - 1\big)$. Via the upperbound in (22), for the submodular case $\gamma = 1$, we obtain $\frac{\ln(2N+1)}{1-\alpha}$. As a remark, we can verify that the guarantee in (23) is not tight. Suppose f is modular, that is, both supermodular and submodular. Then, we have $\frac{(f(R^t)-f(\emptyset))}{(f(R^*)-f(\emptyset))} \leq \ln(2N + 1)$. However, modularity of f implies that the greedy algorithm returns the optimal solution [42]. One reason for this looseness is that, to ensure the tightness of the relaxation from (20) to (21), we must have $R^* \cap R^t = \emptyset$, which then contradicts the modularity of the objective function. To the best of our knowledge, Theorem 2 provides the first performance guarantee for the reverse greedy algorithm for this setting involving the submodularity ratio and the curvature.[7]

Similar to our analysis in Section III, we propose computationally more efficient approaches to deriving both the submodularity ratio and the curvature.

Corollary 3: Let γ^{rg} be the largest $\hat{\gamma}$ that satisfies $\rho_{r_t}(R \cup R^{t-1}) \leq \hat{\gamma}^{-1}\rho_t$, for all $t \leq N$, and R with $|R| = N$. Let α^{rg} be the smallest $\hat{\alpha}$ that satisfies $\rho_r(R \cup R^t) \geq (1-\hat{\alpha})\rho_{r_k^*}(R^t)$, for all $t \leq N$, R with $|R| = N - 1$. Then, γ^{rg} is called the greedy submodularity ratio for the reverse greedy algorithm, with $\gamma^{\mathsf{rg}} \geq \gamma$, and α^{rg} is called the greedy curvature for the reverse greedy algorithm, with $\alpha^{\mathsf{rg}} \leq \alpha$. The performance guarantee is given by

$$\frac{f(R^{\mathsf{r}}) - f(\emptyset)}{f(R^*) - f(\emptyset)} \leq Z_u(N, \gamma^{\mathsf{rg}}, \alpha^{\mathsf{rg}}). \tag{25}$$

The greedy submodularity ratio above can be obtained after the reverse greedy algorithm is completed by analyzing $N\binom{n}{N}$ inequalities, whereas the greedy curvature can be obtained by analyzing $N\binom{n}{N-1}$ inequalities. Since $\gamma^{\mathsf{rg}} \geq \gamma$ and $\alpha^{\mathsf{rg}} \leq \alpha$, it can easily be verified that $Z_u(N, \gamma^{\mathsf{rg}}, \alpha^{\mathsf{rg}}) \leq Z_u(N, \gamma, \alpha)$.

We substitute $f = F_\epsilon^{\mathsf{r}}$ and $\mathcal{F} = \tilde{\mathcal{R}}_K$ to conclude the following.

Corollary 4: Suppose we apply Algorithm 2 to Problem (11). Denote the exclusion set returned as R_ϵ^{r} and the greedy submodularity ratio of F_ϵ^{r} as $\gamma_\epsilon^{\mathsf{rg}}$ and the greedy

---

[7] [28, Thm 7] offers a guarantee for the forward greedy algorithm applied to minimizing increasing functions over a matroid as in Problem (12). This is given by $1/(1 - c)$, where c quantifies how far a function is from being modular. This novel notion is a significantly stronger requirement than having both the submodularity ratio and the curvature simultaneously, see [28, (6)]. Hence, it is not possible to compare it with our guarantee other than the case of a modular objective. In that case, setting $c = 0$ confirms the optimality of the greedy algorithm. Note that computing this novel notion requires an exhaustive enumeration and it does not allow any greedy computation, which can limit its applications.

Algorithm 3 Finding ϵ with provable performance

Input: approximation factor ξ, initial value ϵ_0
Output: parameter ϵ
  function PROPEREP(ξ, ϵ_0)
    $i = 0$
    while $\epsilon_i \geq \xi\lambda_1(S_{\epsilon_i})$ **do**
      let $\epsilon_{i+1} \leftarrow \frac{1}{2}\xi\lambda_1(S_{\epsilon_i})$ and $i \leftarrow i + 1$
    end while
  end function

---

curvature of F_ϵ^{r} as $\alpha_\epsilon^{\mathsf{rg}}$. Then, R_ϵ^{r} satisfies

$$\frac{F_\epsilon(V \setminus R_\epsilon^{\mathsf{r}}) - F_\epsilon(V)}{F_\epsilon(S^*) - F_\epsilon(V)} \leq Z_u^{\mathsf{rg}}, \text{ or equivalently,}$$
$$F_\epsilon(V \setminus R_\epsilon^{\mathsf{r}}) \leq Z_u^{\mathsf{rg}} F_\epsilon(S^*) + (1 - Z_u^{\mathsf{rg}})F_\epsilon(V), \tag{26}$$

where $Z_u^{\mathsf{rg}} := Z_u(N, \gamma^{\mathsf{rg}}, \alpha^{\mathsf{rg}})$.

In contrast to the forward greedy guarantee, $F_\epsilon(\emptyset)$ does not appear in the guarantee above, which is generally large. On the other hand, the guarantee above scales with the problem size, specifically, with $N = n - K$. In the numerics, we show that both greedy algorithms achieve comparable performance in our problem, and at the same time much better performance than what the theoretical guarantees suggest. In practice, it could be useful to implement both greedy algorithms (which can be done efficiently with polynomial time complexity) and choose the best out of the two.

V. IMPLEMENTATION ASPECTS

Two issues have to be addressed to implement Algorithms 1 and 2 for the actuator placement problem. First, we have to select a metric-modifying parameter ϵ. Second, we need feasibility check methods for the set collections $\tilde{\mathcal{C}}_K$ and $\tilde{\mathcal{R}}_K$.

A. An algorithm for picking a metric-modifying parameter ϵ

The performance guarantees (10) and (26) relate to F_ϵ instead of the original metric F. On the one hand, if ϵ is large, a performance guarantee on F_ϵ may not be applicable as a performance guarantee on F since $F_\epsilon(S_\epsilon) < F(S_\epsilon)$. On the other hand, if ϵ is small, the matrix $W_T(S)+\epsilon I$ may be close to singularity. Such ill-conditioned matrices can occur especially at the early stages of the forward greedy algorithm. Denote the actuator set returned by a greedy algorithm applied to F_ϵ as S_ϵ. This could be the solution returned by either Algorithm 1 or Algorithm 2. Given an approximation factor $\xi > 0$ as a design parameter, we propose an algorithm to pick ϵ such that $F(S_\epsilon) < (1+\xi)F_\epsilon(S_\epsilon)$. This inequality implies that guarantees in (10) and (26) translate into guarantees for the metric F. The method is presented in Algorithm 3. Denote the eigenvalues of the controllability Gramian $W_T(S)$ as $\lambda_1(S) \leq \ldots \leq \lambda_n(S)$.

Proposition 6: Suppose given any $\epsilon > 0$, $W_T(S_\epsilon)$ is invertible. Then, for any approximation factor $\xi > 0$ and any initial value $\epsilon_0 > 0$, Algorithm 3 returns (ϵ, S_ϵ) pair that satisfies $F(S_\epsilon) < (1 + \xi)F_\epsilon(S_\epsilon)$.

Proof: If the controllability Gramian $W_T(S_\epsilon)$ is invertible, then we have that $\lambda_1(S_\epsilon) > 0$. Since there are finitely many combinations of actuators, the set $\{\lambda_1(S_\epsilon)|\forall \epsilon > 0\}$ has a positive lower bound, denoted as λ_0. In the iterations of Algorithm 3, it holds that $\epsilon_{i+1} < \frac{1}{2}\epsilon_i$, because $\epsilon_i > \xi\lambda_1(S_{\epsilon_i})$

and $\epsilon_{i+1} = \frac{1}{2}\xi\lambda_1(S_{\epsilon_i})$. Hence, there exists some j such that $\epsilon_j < \xi\lambda_0 \leq \xi\lambda_1(S_{\epsilon_j})$. Then, we obtain $F(S_{\epsilon_j}) = \sum_{i=1}^{n} \frac{1}{\lambda_i(S_{\epsilon_j})} < \sum_{i=1}^{n} \frac{1+\xi}{\lambda_i(S_{\epsilon_j})+\epsilon_j} = (1+\xi)F_{\epsilon_j}(S_{\epsilon_j})$. This inequality concludes the proof. ∎

For the proof above, we assumed that given any $\epsilon > 0$, the controllability Gramian $W_T(S_\epsilon)$ is invertible. This is a strong assumption since, as previously mentioned, structural controllability does not imply controllability. In the numerics, we always ended up with a controllable system with any of the greedy algorithms. This can be explained either by the objective of the problem which is to minimize the average energy consumption or the choice of a large cardinality K.

We now provide the resulting performance guarantees.

Corollary 5: Given the factor ξ, suppose we apply Algorithm 3 to pick ϵ. From Corollaries 2 and 4, we have

$$F(S_\epsilon^{\mathsf{f}}) < (1+\xi)\left[\frac{1}{(\gamma_\epsilon^{\mathsf{fg}})^3 + 1}F(\emptyset) + \frac{(\gamma_\epsilon^{\mathsf{fg}})^3}{(\gamma_\epsilon^{\mathsf{fg}})^3 + 1}F(S^*)\right],$$

$$F(V \setminus R_\epsilon^{\mathsf{r}}) < (1+\xi)\left[Z_u^{\mathsf{rg}}F(S^*) + (1 - Z_u^{\mathsf{rg}})F_\epsilon(V)\right],$$

where S^* is the optimal solution to (3).

B. Feasibility check over $\tilde{\mathcal{C}}_K$

When applied to Problem (5), the forward greedy algorithm has to ensure that the actuator set returned by each iteration lies in $\tilde{\mathcal{C}}_K$. The work of [5], [46] proposes a method to determine whether a given set S with $|S| = K$ belongs to $\mathcal{C}_K$. This result is not directly applicable to answer whether an actuator set S with $|S| < K$ returned by a greedy iteration belongs to $\tilde{\mathcal{C}}_K$. In the following, we extend the work of [5], [46] for a feasibility check over $\tilde{\mathcal{C}}_K$ by constructing auxiliary bipartite graphs associating this check with the cardinality of a maximum matching and by formulating a maximum flow problem.

We introduce the concept of matchings and bipartite graphs. An undirected graph is called bipartite and denoted as (V^1, V^2, E) if its vertices are partitioned into V^1 and V^2 while any edge in E connects a vertex in V^1 to another in V^2. A matching m is a subset of E if no two edges in m share a vertex in common. Given a subset L of $V^1 \cup V^2$, we say L is covered by m if any $v \in L$ is connected to an edge in m. Matching m is maximum if it has the largest cardinality among all the matchings and is perfect if V^2 is covered.

Given the graph $G = (V, E)$ describing system (1), we first build the following auxiliary bipartite graph to determine whether an actuator set renders the system structurally controllable. Node sets $V' = \{v_1', \ldots, v_n'\}$ and $V'' = \{v_1'', \ldots, v_n''\}$ are built as two copies of $V = \{v_1, \ldots, v_n\}$. For any set $S \subset V$, two subsets $S' \subset V'$ and $S'' \subset V''$ denote two copies of the set S. For the bipartite graph, we then have $V^1 = V \cup S''$ and $V^2 = V'$. Next, we define the edge sets. The set E consists of undirected edges connecting v_i with v_j' if $(v_i, v_j) \in E$, whereas the edge set E_1 consists of undirected

edges connecting v_k' with v_k'' if $v_k \in S$. The bipartite graph is then defined by $\mathcal{H}_b(S) = (V \cup S'', V', \mathsf{E} \cup \mathsf{E}_1)$.

If the graph G and the set S pair satisfies the accessibility condition[8] (which is implied by the strong connectivity assumption on G), the set S achieves structural controllability if and only if there exists a perfect matching in $\mathcal{H}_b(S)$, see [46, Thm 2]. This equivalence directly follows from Hall's marriage theorem, which shows that there exists a perfect matching in $\mathcal{H}_b(S)$ if and only if, for any $U \subset V'$, the nodes in U have at least $|U|$ unique in-neighbors [10]. Intuitively, to control any node, we would influence the states of its in-neighbors in the graph. Then, to steer the nodes in U arbitrarily, this theorem implies that we should have at least $|U|$ in-neighbors. Otherwise, suppose two nodes share only a single in-neighbor. Then, these nodes would always be receiving a proportional influence, making it impossible to steer the system states arbitrarily. Using this result, [23] develops a feasibility check for leader selection. This method states that S lies in $\tilde{\mathcal{C}}_K$ if and only if there is a maximum matching for the bipartite graph $\mathcal{H}_b(\emptyset)$ with all the nodes in the set $S' \subset V'$ unmatched. However, this statement is true only if we consider the minimum required cardinality for the structural controllability of the system, see the proof of [23, Lemma 3]. Later in this section, we provide a counterexample where the feasibility check of [23] does not work.

We now provide our feasibility check in the following.

Proposition 7: Given the graph G, the cardinality limit K and an actuator set S with $|S| = k \leq K$, we have $S \in \tilde{\mathcal{C}}_K$ if and only if $|\bar{m}(S)| \geq n - K + k$, where $\bar{m}(S)$ is a maximum matching in $\mathcal{H}_b(S)$.

Proof: "$\Rightarrow$": If $S \in \tilde{\mathcal{C}}_K$, there exists $Q \in \mathcal{C}_K$ such that $S \subset Q$. We now invoke the equivalence result from [46, Thm 2]. This implies the following. By finding a maximum matching m in $\mathcal{H}_b(Q)$ that completely covers Q'' and then excluding from m the edges incident with $Q'' \setminus S''$, we can obtain a matching in $\mathcal{H}_b(S)$ containing $n - K + k$ edges.

"$\Leftarrow$": We pick any maximum matching in $\mathcal{H}_b(S)$ and denote it as m^*. Suppose P' is the largest subset in V' whose elements are all missed by m^*, we know $|P'| \leq K - k$. Denote the edge subset $\mathsf{E}_P \subset \mathsf{E}$ as the set that contains all undirected edges adjacent to v_k' for any k such that $v_k \in P$. Clearly, E_P covers P' and $m^* \cup \mathsf{E}_P$ covers V'. Since matching m^* and matching E_P have no common vertices, $m^* \cup \mathsf{E}_P$ is a perfect matching in $\mathcal{H}_b(S \cup P)$, which means with the actuator set $S \cup P$ the system is structurally controllable. Also considering $|S \cup P| \leq K$, and $S \cup P \in \mathcal{C}_K$, we obtain $S \in \tilde{\mathcal{C}}_K$. ∎

As a remark, [46, Thm 2] associates the existence of a perfect matching in $\mathcal{H}_b(S)$ with a membership of S to $\mathcal{C}_K$, whereas our result extends this previous result by associating the cardinality of a maximum matching in $\mathcal{H}_b(S)$ with a membership of S to $\tilde{\mathcal{C}}_K$. Since we invoke [46, Thm 2], our proposition also requires the accessibility condition. However, note that it is already satisfied by the strong connectivity of G. The proposition above also provides us with a systematic approach to calculate the smallest K required for a nonempty $\tilde{\mathcal{C}}_K$, since we have $\emptyset \in \tilde{\mathcal{C}}_K$ if and only if $|\bar{m}(\emptyset)| \geq n - K$ holds. This method for finding the smallest K coincides with the ones proposed in [8, Thm 4], [36, Thm 3].

---

[8]The graph G and the set S pair satisfies the accessibility condition if for every node in G there is at least one directed path reaching that node from some node in S [46] [10, Def. 2]. When G is strongly connected, this condition is attained irrespective of the nodes chosen in S [10]. We can invoke [46, Thm 2] because of the equivalence of structural controllability in leader selection and actuator placement we established in the proof of Proposition 1.
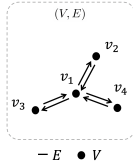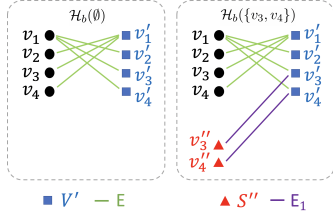
Fig. 1. Graph for the 4-node system

Fig. 2. Auxiliary bipartite graphs $\mathcal{H}_b(\emptyset)$ and $\mathcal{H}_b(\{v_3, v_4\})$

Example 1: Consider a system described by 4 nodes and the dynamic equations (1) where

$$A = \begin{bmatrix} 0 & -0.5 & -0.8 & -0.6 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

$G = (V, E)$ corresponding to this system is in Figure 1. For the metric in (2), let $T = 2$ and $\epsilon = 10^{-9}$.

Consider the actuator placement on this system. We first study the minimum required cardinality for structural controllability. In the auxiliary bipartite graph $\mathcal{H}_b(\emptyset)$ shown in Figure 2, any maximum matching consists of 2 edges, that is, $|\bar{m}(\emptyset)| = 2$. By Proposition 7, $\emptyset \in \tilde{\mathcal{C}}_K$ if and only if $\bar{m}(\emptyset) \geq 4 - K + 0$, that is, $K \geq 2$. Therefore, we need at least 2 actuators to render the system structurally controllable. Suppose $K = 2$. The solution of the forward greedy is $\{v_3, v_4\}$. We depict the auxiliary bipartite graph $\mathcal{H}_b(\{v_3, v_4\})$ in Figure 2 to check whether this actuator set is feasible. Maximum matching contains 4 edges, thus $\bar{m}(\{v_3, v_4\}) = 4 \geq 4 - 2 + 2 = n - K + k$. By Proposition 7, $\{v_3, v_4\}$ belongs to $\tilde{\mathcal{C}}_2$.

We now provide a counterexample based on the example above to show that the feasibility check method in [23] excludes feasible nodes from the consideration of the forward greedy algorithm. Suppose $K = 3$. The feasibility check method in [23] indicates that $\{v_1\} \notin \tilde{\mathcal{C}}_3$, because v_1' is not missed by any maximum matching in $\mathcal{H}_b(\emptyset)$. However, since $\{v_3, v_4\}$ is structurally controllable, so is $\{v_1, v_3, v_4\}$. Then, $\{v_1\} \subset \{v_1, v_3, v_4\}$ implies that $\{v_1\} \in \tilde{\mathcal{C}}_3$.

For our feasibility check, we still need a method to obtain a maximum matching in $\mathcal{H}_b(S)$. It is well-established that this can equivalently be done by solving a maximum flow problem [47]. We refer to Appendix G for details on formulating a maximum flow problem to obtain a maximum matching in $\mathcal{H}_b(S)$. There are several algorithms for solving maximum flow problems. For instance, the Edmonds-Karp algorithm that we adopt in the numerical studies requires $O(pq^2)$ steps, where p and q respectively denote node cardinality and edge cardinality in the flow graph generated based on $\mathcal{H}_b(S)$ [48]. For example, in $\mathcal{H}_b(\emptyset)$, $p = 2n + 2$ and $q = 2n + |E|$. Thus, at each forward greedy iteration, we can examine in polynomial time whether $v \cup S^t$ belongs to $\tilde{\mathcal{C}}_K$ by finding the cardinality of the maximum matching in $\mathcal{H}_b(v \cup S^k)$.

C. Feasibility check over $\tilde{\mathcal{R}}_K$

The reverse greedy algorithm has to determine whether $R \in \tilde{\mathcal{R}}_K$, or equivalently, whether any subset of the set $V \setminus R$ belongs to $\mathcal{C}_K$. Invoking the equivalence result of [46, Thm 2], we can conclude that there exists a subset of $V \setminus R$ belonging to $\mathcal{C}_K$ if and only if there exists a perfect matching in $\mathcal{H}_b(V \setminus R)$ that covers at most K elements of $V'' \setminus R''$. This holds, since if every perfect matching in $\mathcal{H}_b(V \setminus R)$ covers $K + 1$ or more nodes in $V'' \setminus R''$, it would not be possible to find K actuators from $V \setminus R$ satisfying structural controllability.

Recall that a maximum matching can be computed via the maximum flow algorithm. Analogous to the previous section, we need a feasibility check method for $\tilde{\mathcal{R}}_K$ by the means of the flow theory. We refer to Appendix G for the preliminaries regarding flows in graphs. We first build an auxiliary graph, denoted by $\mathcal{H}_r(S)$, containing all the nodes in $\mathcal{H}_b(S)$. We let s and t be the sink and source of the flow, respectively. In addition, we add node s'' to $\mathcal{H}_r(S)$, which will enable encoding the cardinality limit on $V'' \setminus R''$. The edge set in $\mathcal{H}_r(S)$ is the union of three sets, E_b^f, E_s^f and E_t^f, all of which are directed. The edge set E_b^f is a copy of $\mathsf{E} \cup \mathsf{E}_1$, originally from $\mathcal{H}_b(S)$, but directed from $V \cup S''$ to V' in $\mathcal{H}_r(S)$. The edge set E_s^f consists of edges from s to all the nodes in V and from s'' to all the nodes in S'' along with edge of from s to s''. Finally, the edge set E_t^f is composed of edges from all the nodes in V' to t. All the edges have unit capacity except the edge from s to s'' which has a capacity of K. Utilizing the graph $\mathcal{H}_r(S)$, we have the following proposition.

Proposition 8: Given the cardinality limit K and an exclusion set R, we have $R \in \tilde{\mathcal{R}}_K$ if and only if there exists a flow g in $(\mathcal{H}_r(V \setminus R), c, s, t)$ with val$(g) = n$.

Proof: From the definition of $\tilde{\mathcal{R}}_K$, $R \in \tilde{\mathcal{R}}_K$ is equivalent to the existence of $S \subset V \setminus R$ such that $S \in \mathcal{C}_K$. Via [46, Thm 2], we know that these two conditions are equivalent to the existence a perfect matching in $\mathcal{H}_b(V \setminus R)$ that covers at most K elements of $V'' \setminus R''$. For the following, we prove that this equivalent condition holds if and only if there exists a flow g in $(\mathcal{H}_r(V \setminus R), c, s, t)$ with val$(g) = n$.

"$\Rightarrow$": Given the perfect matching m^*, we use $\tilde{n}$ to denote the number of the elements in V'' that are adjacent to m^*. Clearly, $\tilde{n} \leq K$. For the following, we build the flow g as a function of the edges in $\mathcal{H}_r(V \setminus R)$. Suppose m^f is a subset of E_b^f that corresponds with m^* in $\mathsf{E} \cup \mathsf{E}_1$. We let $g(e) = 1$ if e belongs to m^f, $g((s, v)) = 1$ for any $v \in V$ incident with m^f, $g((s'', v'')) = 1$ for any $v'' \in V''$ incident with m^f, $g(e_t) = 1$ for any $e_t \in \mathsf{E}_t^f$ and finally $g((s, s'')) = \tilde{n}$. It is easy to check g is in fact a flow in $(\mathcal{H}_r(V \setminus R), c, s, t)$ with val$(g) = n$.

"$\Leftarrow$": Let $\mathsf{E}^m = \{e^f \in \mathsf{E}_b^f \mid g(e^f) = 1\}$. Then, define $m^* = \{e \in \mathsf{E} \cup \mathsf{E}_1 \mid \exists e^f \in \mathsf{E}^m \text{ where } e^f \text{ is a copy of } e\}$. It follows from val$(g) = n$ that m^* is a perfect matching in $\mathcal{H}_b(V \setminus R)$. Since the capacity limits are satisfied by the flow g, there are no more than K elements in $V'' \setminus R''$ covered by the perfect matching m^*. ∎

Example 2: We apply the reverse greedy algorithm to the system studied in Example 1 with $K = 2$. The first node excluded is v_1. To see that $v_1 \in \tilde{\mathcal{R}}_2$, we depict $\mathcal{H}_r(V \setminus v_1)$ in Figure 3. The maximum flow has a value of 4. Invoking
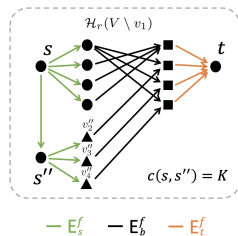
Fig. 3. Auxiliary graph $\mathcal{H}_r(V \setminus v_1)$

Fig. 4. Greedy selection versus the optimal selection

Proposition 8, we conclude that v_1 belongs to $\tilde{\mathcal{R}}_2$.

Similar to Section V-B, we adopt Edmonds-Karp algorithm to solve the maximum flow problem in the numerical studies. The algorithm requires $O(pq^2)$ steps where p and q are respectively the node cardinality and the edge cardinality of the flow graph $\mathcal{H}_r(V \setminus R)$. For example, in $\mathcal{H}_r(V)$, $p = 3n+3$ and $q = 3n+|E|+1$, where E is the edge set of $\mathcal{H}_r(V)$. Thus, at each reverse greedy iteration, we can examine in polynomial time whether $v \cup R^t$ belongs to $\tilde{\mathcal{R}}_K$.

Remark: Greedy algorithms can also be applied when strong connectivity assumption on G is relaxed. Suppose G is not strongly connected but it can be decomposed as $\cup_{i=1}^{l} G_i$, where G_i is strongly connected for any i. In this case, if we suppose at least one actuator is already chosen and assigned for each subgraph G_i, we would attain the accessibility condition discussed in Footnote 8. It is then possible to invoke [46, Thm 2] for the equivalence between perfect matching and structural controllability. This would make it possible to extend the proofs of Propositions 7 and 8 for the case when we are assigning additional actuators in such graphs. We kindly refer the readers to [10], [46] for a detailed discussion on this condition. Similar to our paper, many works in the literature, such as [10], [23], assume that G is strongly connected such that the accessibility condition is automatically attained.

VI. NUMERICAL RESULTS

In this section, we apply the greedy algorithms to problems based on randomly generated networks and a large power grid. All problems are solved on a computer equipped with 8 GB RAM and a 2.7 GHz dual-core Intel i5 processor.

A. Experiment on a 23-node network

We study a system model based on an undirected un-weighted graph given in Figure 4 generated via Octave Networks Toolbox [49]. Different degrees are assigned to each vertex such that we can compare the sets S_ϵ^f and $S_\epsilon^r := V \setminus R_\epsilon^r$ in terms of node connectivity. Specifically, vertex i has a degree of i if $i < 12$ and a degree of $24 - i$ if $i \geq 12$. If there is an edge between vertex i and j, we set $(A)_{ij} = (A)_{ji} = 1$, otherwise the corresponding entries are 0.

Let $T = 1$ and $K = 8$. We then apply Algorithm 3 to obtain a proper parameter ϵ for the forward greedy algorithm. We set $\xi = 2$ and $\epsilon_0 = 10^{-3}$ arbitrarily. The actuator set returned in the first iteration is denoted by $S_{\epsilon_0}^f$. The minimum eigenvalue corresponding to $W_T(S_{\epsilon_0}^f)$ is $\lambda_1(S_{\epsilon_0}^f) = 1.9 \times 10^{-4}$. Since $\epsilon_0 > 2\lambda_1(S_{\epsilon_0}^f)$, we continue with the second iteration. Let $\epsilon_1 = \lambda_1(S_{\epsilon_0}^f) = 1.9 \times 10^{-4}$, we now have $S_{\epsilon_1}^f = \{4, 6, 8, 10, 13, 16, 20, 21\}$ and $\lambda_1(S_{\epsilon_1}^f) = 2.0 \times 10^{-4} > \xi^{-1}\epsilon_1$.
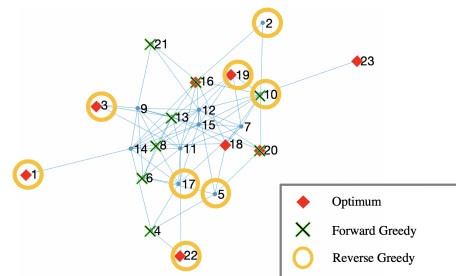
Thus, we can terminate the algorithm and pick $\epsilon_f = \epsilon_1$ for the forward greedy algorithm. Using the same procedure, we obtain $\epsilon_r = 1.4 \times 10^{-4}$ for the reverse greedy algorithm. In this case, the solution is $S_{\epsilon_r}^r = V \setminus R_{\epsilon_r}^r = \{1, 2, 3, 5, 10, 17, 19, 22\}$.

To assess the optimality of the sets $S_{\epsilon_f}^f$ and $S_{\epsilon_r}^r$, we generate the optimal solution $S^* = \{1, 3, 16, 18, 19, 20, 22, 23\}$ by enumerating all feasible solutions. The average energy consumptions for all actuator sets are given by $F(S_{\epsilon_f}^f) = 9226.5$, $F(S_{\epsilon_r}^r) = 12126.2$, and $F(S^*) = 6052.7$. For this example, the forward greedy algorithm returns a better solution than the reverse greedy algorithm. Later, in randomized examples we see that this is not generally the case.

Next, we analyze the performance guarantees in (10) and (26) under the sets $S_{\epsilon_f}^f$ and $S_{\epsilon_r}^r$. For the forward greedy algorithm, we computed $F_{\epsilon_f}(S_{\epsilon_f}^f) = 6188.2 = 0.66F(S_{\epsilon_f}^f)$. The greedy submodularity ratio for the forward greedy algorithm is computed as $\gamma_{\epsilon_f}^{fg} = 1$. Then, we obtain $F(S_{\epsilon_f}^f) = 1.5F_{\epsilon_f}(S_{\epsilon_f}^f) \leq 0.75F_{\epsilon_f}(\emptyset) + 0.75F_{\epsilon_f}(S^*) = 9.2 \times 10^4 + 0.75F_{\epsilon_f}(S^*)$. In this example, the appearance of $F_{\epsilon_f}(\emptyset)$ in the performance guarantee undermines its tightness. On the other hand, for the reverse greedy algorithm, the greedy submodularity ratio of the objective function $F_{\epsilon_r}^r$ is computed as $\gamma_{\epsilon_r}^{rg} < 0.01$. This value is negligibly small making the performance guarantee in (26) loose.

B. Node connectivity analysis on the sets $S_{\epsilon_f}^f$ and $S_{\epsilon_r}^r$

To gain additional insights into solution dependence on node connectivity, we now compare the greedy solutions with the optimal solutions in terms of the degrees of the selected actuators. In the previous study, the forward greedy algorithm selects the actuator set $S_{\epsilon_f}^f$ in the order of 16, 13, 5, 8, 6, 20, 10, 21. In this sequence, the first four nodes feature high degrees. This is because the high degree nodes generally result in larger marginal gains at the earlier stages of the forward greedy algorithm. Let $d_\Sigma(S)$ denote the sum of the degrees of all the nodes in set S. Observe that $d_\Sigma(S_{\epsilon_f}^f) = 54$, whereas $d_\Sigma(S_{\epsilon_r}^r) = 35$ and $d_\Sigma(S^*) = 30$. This demonstrates that the reverse greedy algorithm does not have a tendency to pick high degree nodes. We illustrate these sets in Figure 4.

To show that this observation is not restricted to this specific example, we build 20 random graphs with 23 nodes using Octave [49]. These graphs are built as follows. For $i = 1, 4, 7$, node i, node $i+1$ and node $i+2$ have randomized degrees between i and $i+2$. Node 10 and node 11 have randomized degrees between 10 and 11. Node 12 has exactly 12 neighbors. For $i > 12$, Node i has a degree number the same as that of Node $24 - i$. For each algorithm run, a proper parameter ϵ is

TABLE II
DEGREE SUM COMPARISONS OF DIFFERENT SOLUTIONS

Netw.	$d_\Sigma(S^f)$	$d_\Sigma(V \setminus R^r)$	$d_\Sigma(S_o^*)$	Netw.	$d_\Sigma(S^f)$	$d_\Sigma(V \setminus R^r)$	$d_\Sigma(S_o^*)$
1	43	46	46	11	41	38	42
2	47	47	49	12	60	44	53
3	48	34	41	13	50	47	47
4	49	35	43	14	55	49	41
5	45	43	32	15	53	41	43
6	58	39	41	16	56	43	46
7	44	45	49	17	48	52	70
8	52	25	44	18	57	48	54
9	55	37	45	19	76	60	50
10	61	38	59	20	57	43	44

TABLE III
METRIC COMPARISONS OF DIFFERENT SOLUTIONS

Netw.	$F(S^f)$	$F(V \setminus R^r)$	$F(S_o^*)$	Netw.	$F(S^f)$	$F(V \setminus R^r)$	$F(S_o^*)$
1	66252	13493	11327	11	11981	14993	7300
2	11527	10035	8461	12	9950	9388	9358
3	15398	17371	8461	13	32212	6684	9364
4	13679	15690	7593	14	9683	10681	6804
5	14835	14430	8406	15	11235	8540	7188
6	18207	15176	7870	16	14114	12004	6795
7	8980	12650	10515	17	8658	9163	8416
8	22600	26324	6638	18	8717	10587	6838
9	10633	10483	9690	19	13760	9818	9336
10	9676	13079	6173	20	10044	13169	9264

picked via Algorithm 3.

Comparisons of different actuator sets can be found in Tables II and III. The set S_o^* refers to the best solution out of 1×10^4 random selections of cardinality $K = 8$, while it is not computationally feasible to obtain the exact optimal solution for each case. Table II shows that the forward greedy generally yields an actuator set with a high degree sum when compared to the other solutions. Finally, Table III shows that in several cases the set returned by the forward greedy results in significantly worse value in the objective than the other two solutions. Generally, both greedy algorithms achieve comparable performance, and at the same time much better performance than what the theoretical guarantees suggest (as in Section VI-A). Thus, it could be useful to implement both polynomial-time algorithms, and choose the best solution.

The total computation time for 20 forward greedy algorithm runs is 8205.0 seconds, whereas the time for 20 reverse greedy algorithm runs is 665.0 seconds. It turns out that for this problem the reverse greedy algorithm requires fewer queries to the computationally expensive feasibility check problem when compared to the forward greedy algorithm.

C. Power electronic actuator placement for 118-bus system

We illustrate our results by placing power electronic actuators that can modulate power injections in the IEEE 118-bus test system, provided in [50]. Similar to [1, §4.B], each bus is assumed to follow the linearized swing equations, that is, $M_i \ddot{\theta}_i + D_i \dot{\theta}_i = P_i - \sum_j a_{ij}(\theta_i - \theta_j)$ for all $i \in \{1, \ldots, 118\}$, where P_i is the net power injection and a_{ij} is characterized by the line parameters. We kindly refer to [51] for the modeling details. If bus i is not in our actuator set, we have $P_i = 0$. Buses without generators are assumed to have no inertia (even when they have loads connected), and they have a one-dimensional state, since $M_i = 0$ (inertia). Buses with generators have inertias, and they are instead associated with a two-dimensional state vector that includes both θ_i and $\dot{\theta}_i$. We highlight that each state corresponds to a new node in our system graph G, which in some sense represents an extended version of the original 118-bus power network.

TABLE IV
METRIC COMPARISONS FOR THE IEEE 118-BUS TEST SYSTEM

K	$F(S^f)$	$F(V \setminus R^r)$	$F(S_o^*)$
50	6.96×10^7	3.01×10^7	4.48×10^9
70	1.94×10^5	1.25×10^5	4.43×10^6

Our goal is to choose $K = 50, 70$ buses out of 118 buses to inject power to minimize the average energy consumption given by (2) ($T = 1$), while ensuring structural controllability. We take into account the fact that it is not possible to actuate some of the nodes of our system graph G by excluding them from the actuator sets and the feasibility check methods. These nodes correspond to θ_i originating from buses with inertia, since the dynamics $d\theta_i/dt = \dot{\theta}_i$ cannot be actuated.

Next, we implement the greedy algorithms. The results are shown in Table IV, where S_o^* is the best structurally controllable solution out of 10^5 random selections with cardinality K. The parameters for both greedy algorithms can also be summarized as follows: $\epsilon = 1 \times 10^{-10}$ for $K = 50$ and $\epsilon = 1 \times 10^{-8}$ for $K = 70$, chosen arbitrarily. Both greedy algorithms perform significantly better than random selections. Note that there are 6.2×10^{33} and 3.2×10^{33} possible combinations to check for $K = 50$ and $K = 70$, respectively. Since the optimal solution is computationally out of reach, we will not analyse the performance guarantees in (10) and (26) as we did in the previous sections. In both cases $K = 50$ and $K = 70$, the reverse greedy performed slightly better than the forward greedy algorithm. Finally, as we expected, choosing a larger K reduces the control cost for all three solution concepts. One can decide on K by evaluating the overall cost reductions from the reductions in the metric and comparing them with the actuator installation costs.

VII. CONCLUSIONS

In this paper, our goal was to pick an actuator set to minimize a controllability metric based on average energy consumption while ensuring that the system is structurally controllable. To this end, we reformulated our problem as matroid optimization problems to apply both the forward and reverse greedy algorithms. For each algorithm, we provided a novel performance guarantee. For the implementation of the algorithms, we proposed feasibility check methods. In the numerics, we studied networks that are randomly generated based on degree lists. We observed that the forward greedy tended to select high-degree nodes in the early stages, whereas the overall performance of both algorithms were comparable.

Our future work involves exploiting the curvature of the objective function to derive a better performance guarantee for the forward greedy algorithm. We will exploit the problem structure to explain why algorithms performs significantly better than their performance guarantees. We aim to investigate other structural controllability concepts from the literature.

REFERENCES

[1] T. H. Summers, F. L. Cortesi, and J. Lygeros, "On submodularity and controllability in complex dynamical networks," *IEEE Trans. on Contr. of Netw. Syst.*, vol. 3, no. 1, pp. 91–101, 2015.

[2] F.-J. Müller and A. Schuppert, "Few inputs can reprogram biological networks," *Nature*, vol. 478, no. 7369, p. E4, 2011.

[3] A. Banerjee and Y. Arkun, "Control configuration design applied to the tennessee eastman plant-wide control problem," *Comp. & Chemical Engineering*, vol. 19, no. 4, pp. 453–480, 1995.

[4] M. Van De Wal and B. De Jager, "A review of methods for input/output selection," *Automatica*, vol. 37, no. 4, pp. 487–510, 2001.

[5] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, "Controllability of complex networks," *Nature*, vol. 473, no. 7346, p. 167, 2011.

[6] F. Pasqualetti, S. Zampieri, and F. Bullo, "Controllability metrics, limitations and algorithms for complex networks," *IEEE Trans. on Contr. of Netw. Syst.*, vol. 1, no. 1, pp. 40–52, 2014.

[7] A. Olshevsky, "Minimal controllability problems," *IEEE Trans. on Contr. of Netw. Syst.*, vol. 1, no. 3, pp. 249–258, 2014.

[8] S. Pequito, S. Kar, and A. P. Aguiar, "A framework for structural input/output and control configuration selection in large-scale systems," *IEEE Trans. on Aut. Contr.*, vol. 61, no. 2, pp. 303–318, 2015.

[9] A. Jadbabaie, A. Olshevsky, G. J. Pappas, and V. Tzoumas, "Minimal reachability is hard to approximate," *IEEE Trans. on Aut. Contr.*, vol. 64, no. 2, pp. 783–789, 2019.

[10] A. Clark, B. Alomair, L. Bushnell, and R. Poovendran, "Submodularity in input node selection for networked linear systems: Efficient algorithms for performance and controllability," *IEEE Contr. Syst. Magazine*, vol. 37, no. 6, pp. 52–74, 2017.

[11] O. Romero and S. Pequito, "Actuator placement for symmetric structural controllability with heterogeneous costs," *IEEE Contr. Syst. Letters*, vol. 2, no. 4, pp. 821–826, 2018.

[12] S. Pequito, G. Ramos, S. Kar, A. P. Aguiar, and J. Ramos, "The robust minimal controllability problem," *Automatica*, vol. 82, pp. 261–268, 2017.

[13] V. Tzoumas, A. Jadbabaie, and G. J. Pappas, "Robust and adaptive sequential submodular optimization," *arXiv preprint arXiv:1909.11783*, 2019.

[14] V. Tzoumas, M. A. Rahimian, G. J. Pappas, and A. Jadbabaie, "Minimal actuator placement with bounds on control effort," *IEEE Trans. on Contr. of Netw. Syst.*, vol. 3, no. 1, pp. 67–78, March 2016.

[15] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions-I," *Math. Prog.*, vol. 14, no. 1, pp. 265–294, 1978.

[16] T. Summers and M. Kamgarpour, "Performance guarantees for greedy maximization of non-submodular controllability metrics," in *18th Eur. Contr. Conf.*, 2019, pp. 2796–2801.

[17] A. Das and D. Kempe, "Submodular meets spectral: greedy algorithms for subset selection, sparse approximation and dictionary selection," in *28th Int. Conf. on Mach. Lrn.*, 2011, pp. 1057–1064.

[18] A. A. Bian, J. M. Buhmann, A. Krause, and S. Tschiatschek, "Guarantees for greedy maximization of non-submodular functions with applications," in *34th Int. Conf. on Mach. Lrn.*, 2017, pp. 498–507.

[19] C.-T. Lin, "Structural controllability," *IEEE Trans. on Aut. Contr.*, vol. 19, no. 3, pp. 201–208, June 1974.

[20] R. Shields and J. Pearson, "Structural controllability of multi-input linear systems," *IEEE Trans. on Aut. Contr.*, vol. 21, no. 2, pp. 203–212, 1976.

[21] J.-M. Dion, C. Commault, and J. Van Der Woude, "Generic properties and control of linear structured systems: a survey," *Automatica*, vol. 39, no. 7, pp. 1125–1144, 2003.

[22] G. Ramos, A. P. Aguiar, and S. Pequito, "Structural systems theory: an overview of the last 15 years," *arXiv preprint arXiv:2008.11223*, 2020.

[23] A. Clark, L. Bushnell, and R. Poovendran, "On leader selection for performance and controllability in multi-agent systems," in *51st IEEE Conf. on Dec. and Contr.*, 2012, pp. 86–93.

[24] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey, "An analysis of approximations for maximizing submodular set functions-II," in *Polyhedral combinatorics*. Springer, 1978, pp. 73–87.

[25] M. Conforti and G. Cornuéjols, "Submodular set functions, matroids and the greedy algorithm: tight worst-case bounds and some generalizations of the rado-edmonds theorem," *Discrete App. Math.*, vol. 7, no. 3, pp. 251–274, 1984.

[26] L. Chen, M. Feldman, and A. Karbasi, "Weakly submodular maximization beyond cardinality constraints: Does randomization help greedy?" in *35th Int. Conf. on Mach. Lrn.*, 2018, pp. 804–813.

[27] L. F. Chamon, A. Amice, and A. Ribeiro, "Matroid-constrained approximately supermodular optimization for near-optimal actuator scheduling," in *To appear in 58th IEEE Conf. on Dec. and Contr.*, 2019.

[28] M. Sviridenko, J. Vondrák, and J. Ward, "Optimal approximation for submodular and supermodular optimization with bounded curvature," *Math. of Oper. Res.*, vol. 42, no. 4, pp. 1197–1218, 2017.

[29] T. Zhang, "Adaptive forward-backward greedy algorithm for learning sparse representations," *IEEE Trans. on Inf. Theory*, vol. 57, no. 7, pp. 4689–4708, 2011.

[30] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. on Inf. Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.

TABLE V
PERFORMANCE GUARANTEES FOR THE FORWARD GREEDY ALGORITHM
APPLIED TO THE MAXIMIZATION OF INCREASING SET FUNCTIONS

Setting	submodular	submodular with known curvature	known submodularity ratio	known submodularity ratio and known curvature
cardinality constraints	$1-e^{-1}$ [15]	$(1-e^{-\alpha})/\alpha$ [25]	$1-e^{-\gamma}$ [18]	$(1-e^{-\alpha\gamma})/\alpha$ [18]
general matroid constraints	1/2 [24]	$1/(1+\alpha)$ [25]	$\gamma^3/(1+\gamma^3)$ in this paper see Appendix B for [26]	open

[31] M. Chrobak, C. Kenyon, and N. Young, "The reverse greedy algorithm for the metric k-median problem," *Inf. Processing Letters*, vol. 97, no. 2, pp. 68–72, 2006.

[32] V. P. Il'ev, "An approximation guarantee of the greedy descent algorithm for minimizing a supermodular set function," *Discrete App. Math.*, vol. 114, no. 1-3, pp. 131–146, 2001.

[33] V. Il'ev and N. Linker, "Performance guarantees of a greedy algorithm for minimizing a supermodular set function on comatroid," *Eur. J. of Oper. Res.*, vol. 171, no. 2, pp. 648–660, 2006.

[34] O. Karaca, B. Guo, and M. Kamgarpour, "A comment on performance guarantees of a greedy algorithm for minimizing a supermodular set function on comatroid"," *Eur. J. of Oper. Res.*, 2020.

[35] B. Guo, O. Karaca, T. Summers, and M. Kamgarpour, "Actuator placement for optimizing network performance under controllability constraints," in *To appear in 58th IEEE Conf. on Dec. and Contr.*, 2019.

[36] S. Pequito, S. Kar, and A. P. Aguiar, "A structured systems approach for optimal actuator-sensor placement in linear time-invariant systems," in *Amer. Contr. Conf.* IEEE, 2013, pp. 6108–6113.

[37] M. Doostmohammadian, H. R. Rabiee, and U. A. Khan, "Structural cost-optimal design of sensor networks for distributed estimation," *IEEE Signal Proc. Let.*, vol. 25, no. 6, pp. 793–797, 2018.

[38] A. Krause and C. E. Guestrin, "Near-optimal nonmyopic value of information in graphical models," in *Conf. on Uncertainty in AI*, 2005.

[39] F. Bach, "Learning with submodular functions: A convex optimization perspective," *Found. and Tr. in Mach. Lrn.*, vol. 6, no. 2-3, pp. 145–373, 2013.

[40] A. Krause and D. Golovin, "Submodular function maximization," in *Tractability: Prac. App. to Hard Problems*. Cambridge, 2014.

[41] V. Tzoumas, A. Jadbabaie, and G. J. Pappas, "Resilient non-submodular maximization over matroid constraints," *arXiv preprint arXiv:1804.01013*, 2018.

[42] J. Edmonds, "Matroids and the greedy algorithm," *Math. Prog.*, vol. 1, no. 1, pp. 127–136, 1971.

[43] D. Welsh, *Matroid Theory*. Dover Publications, 2010.

[44] Z. Svitkina and L. Fleischer, "Submodular approximation: Sampling-based algorithms and lower bounds," *SIAM J. on Comp.*, vol. 40, no. 6, pp. 1715–1737, 2011.

[45] B. Guo, O. Karaca, T. Summers, and M. Kamgarpour, "Actuator placement under structural controllability using forward and reverse greedy algorithms," *arXiv:1912.05149*, 2019.

[46] Y.-Y. Liu, J.-J. E. Slotine, and A.-L. Barabási, "Controllability of complex networks : Supplementary information," 2011.

[47] L. Lovász and M. Plummer, *Matching Theory*, ser. North-Holland Math. Studies. Elsevier Science, 1986.

[48] J. Edmonds and R. M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *J. of the ACM*, vol. 19, no. 2, pp. 248–264, 1972.

[49] G. Bounova, "Octave networks toolbox," doi: 10.5281/zenodo.22398.

[50] R. Christie, "Power systems test case archive," *U of Washington*, 2017. [Online]. Available: https://www2.ee.washington.edu/research/pstca/

[51] A. R. Bergen and D. J. Hill, "A structure preserving model for power system stability analysis," *IEEE Trans. on Pow. App. and Syst.*, no. 1, pp. 25–35, 1981.

[52] K. B. Petersen and M. S. Pedersen, *The Matrix Cookbook*. Technical University of Denmark, 2012.

APPENDIX

A. Performance guarantees from the literature

Before summarizing the relevant guarantees from the literature, we highlight that if we assign each actuator with a specific control cost, we would obtain a modular objective. There are many works that study modular objectives with or without structural controllability type constraints [1], [11], [37]. In case our objective is additive/modular, greedy algorithm on a matroid is known to always return an optimal solution [42].

Table V summarizes the performance guarantees for the forward greedy algorithm applied to the maximization of increasing set functions. As a remark, [18] defines the curvature as we defined in Definition 3, whereas [25] considers only the case where $S = \emptyset$ and $U = V \setminus \{v\}$ for all $v \in V$ and they call this notion the total curvature. In addition, the work of [28] provides $1 - (\alpha/e)$ guarantee that holds by high probability for maximizing increasing submodular functions with curvature over an arbitrary matroid constraint. This result relies on linear extensions of the objective by implementing the continuous greedy algorithm, which is subject to potential deviations from the guarantee due to the random rounding procedures. We refer to [28] for other guarantees from the literature relying on variations of this algorithm. Finally, [26] relies on a randomized forward greedy providing guarantees only in expectation, see Appendix B. A very recent work [27] provides $\gamma/2$ guarantee for the forward greedy algorithm applied to our setting with the submodularity ratio in Definition 2. This result does not rely on constructing a linear program. Instead, it relies on [27, Lem. 1], which has a mistake in its proof. Using the notation of [27], let $\mathcal{G}_t$ be the greedy solution at iteration t, $\mathcal{X}^\star$ be the optimum, and $\mathcal{I}$ be the feasible region of the matroid. In the proof, the authors need to find an enumeration $\{x_1^\star, \ldots, x_n^\star\} = \mathcal{X}^\star$ such that $\mathcal{G}_t \cup \{x_{t+1}^\star\} \in \mathcal{I}$ for any t. Starting from $\mathcal{G}_0 = \emptyset$, their inductive proof iteratively builds this enumeration by choosing $x_{t+1}^\star \in \mathcal{X}^\star$ with $\mathcal{G}_t \cup \{x_{t+1}^\star\} \in \mathcal{I}$. The proof states "$|\mathcal{G}_t| < |\mathcal{X}^\star|$ implies there exist an element $x_{t+1}^\star \in \mathcal{X}^\star$ such that $\mathcal{G}_t \cup \{x_{t+1}^\star\} \in \mathcal{I}$ (property (iii) of Definition 4)". However, for their inductive proof to be correct, they instead have to prove that there exists an element $x_{t+1}^\star \in \mathcal{X}^\star \setminus \{x_k^\star\}_{k=1}^t$ such that $\mathcal{G}_t \cup \{x_{t+1}^\star\} \in \mathcal{I}$. We think that this statement could also be proved. In this case, their guarantee would be better.

The guarantees above take as a reference objective evaluated at the empty set. The second part of this work studies the reverse greedy algorithm where the guarantees are with respect to the objective evaluated at the full set (see (26) and (9)).

B. Definitions of submodularity ratio

Let γ_1 denote the submodularity ratio of f from Definition 2. Observe that $\gamma = \gamma_1$ satisfies

$$\gamma \rho_U(S) \leq \sum_{v \in U \setminus S} \rho_v(S), \quad \forall S, U \subset V, \quad (27)$$

which can easily be obtained by decomposing the term on the left via telescoping sum. However, the largest γ satisfying the above set of inequalities, denoted as γ_2, does not necessarily satisfy the inequalities in Definition 2. This is true since the inequalities in (27) can be regarded as a relaxation of those in Definition 2. Hence, we have $\gamma_2 \geq \gamma_1$. There are previous studies in the literature defining the submodularity ratio as γ_2 instead of γ_1 [16], [18], [26]. In the proof of Theorem 1, as we are deriving (30), we use the inequalities from Definition 2. One can verify that the inequalities in (27) would not allow us to derive (30). Hence, the performance guarantee (7) does not extend to the submodularity ratio γ_2.

In addition, the work of [16] obtains a lower bound for γ_2 for the metric $-F$ in (2) based on eigenvalue inequalities for
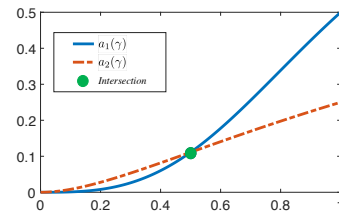

Fig. 5. A comparison between two guarantees

sum and product of matrices. One can easily verify that this lower bound is also applicable to γ_1 from Definition 2.

The work of [26] exploited the submodularity ratio defined by (27) and obtained a guarantee in expectation for the residual random (forward) greedy algorithm for matroid optimization problems featuring weakly submodular objective functions. We denote the final set returned by this algorithm as S^{RRG}. The guarantee provided in [26] for this class of randomized algorithms is $\frac{\mathbb{E}[f(S^{\text{RRG}})] - f(\emptyset)}{f(S^*) - f(\emptyset)} \geq \frac{\gamma_2^2}{(1+\gamma_2)^2}$. Let γ denote the theoretical lower bound derived in [16] for $-F$ in (2). This lower bound satisfies $\gamma_2 \geq \gamma_1 \geq \gamma$. Since γ is applicable to both (7) and the guarantee in [26], we let $a_1(\gamma) = \gamma^3/(1+\gamma^3)$ and $a_2(\gamma) = \gamma^2/(1+\gamma)^2$ denote the theoretical guarantees associated with Theorem 1 and the one in [26], respectively. Two functions are plotted in Figure 5. The guarantee we derived in Theorem 1 is tighter than the one from [26], if the lower bound $\gamma > 0.5$ (it is also an ex-post guarantee).

C. Proof of Lemma 2

(i) For any $S \subset V$ and any $v \in V \setminus S$, let $H(z) = (W_T(S) + zW_T(\{v\}) + \epsilon I)^{-1}$. Notice that $\text{tr}(H(1)) = \text{tr}((W_T(S \cup \{v\}) + \epsilon I)^{-1}) = F_\epsilon(S \cup \{v\})$, since $W_T(S) = \int_0^T e^{A\tau} B(S) B^\top(S) e^{A^\top \tau} d\tau$ is additive, that is, $W_T(S) + W_T(\{v\}) = W_T(S \cup \{v\})$. Via the matrix inverse formula [52], if $H(z)$ is invertible $\forall z \in (0,1)$, then $\text{tr}(H(z))$ is continuous and differentiable, and we have $\frac{d(\text{tr}(H(z)))}{dz} = -\text{tr}(H(z)W_T(\{v\})H(z)) < 0$. This inequality holds since $H(z)$ is invertible and symmetric, and $W_T(\{v\})$ is positive semidefinite. Invoking the mean-value theorem, we have $\text{tr}(H(1)) - \text{tr}(H(0)) < 0$.

(ii) Recall from (4) that the submodularity ratio of $-F_\epsilon$, denoted as $\gamma_\epsilon^{\text{f}}$, satisfies $\gamma_\epsilon^{\text{f}} = \min_{S,U,v \in V \setminus (S \cup U)} \frac{\rho_v(S)}{\rho_v(S \cup U)}$. Since $-F_\epsilon$ is strictly increasing, $\rho_v(S) > 0$ and $\rho_v(S \cup U) > 0$ for any $v \in V \setminus (S \cup U)$. Thus $\gamma_\epsilon^{\text{f}} > 0$. ∎

D. Proof of Proposition 1

To prove this theorem, we show that given an actuator set S, structural controllability of $(A, B(S))$ can equivalently be formulated as structural controllability of the system with the set S chosen as a leader set. Then, we use a result from [23] showing the matroid structure of the structural controllability constraints in leader selection problems. This result builds on [5], which shows the equivalence between structural controllability and existence of a perfect matching in an auxiliary bipartite graph whenever the graph G is strongly connected.

Define $N = V \setminus S$ and partition the state vector x into x_S and x_N. Our dynamics can equivalently be written as

$$\begin{bmatrix} \dot{x}_N \\ \dot{x}_S \end{bmatrix} = \begin{bmatrix} A_{NN} & A_{NS} \\ A_{SN} & A_{SS} \end{bmatrix} \begin{bmatrix} x_N \\ x_S \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & I_{|S|} \end{bmatrix} u, \quad (28)$$

where $I_{|S|} \in \mathbb{R}^{|S| \times |S|}$ is the identity matrix.

In the leader selection problem, if the set S is chosen as a leader set, it is assumed that the values of x_S are directly dictated and are not influenced by the dynamics of x_N. Under this assumption, by treating x_S as the input, the dynamics of x_N are given by $\dot{x}_N = A_{NN}x_N + A_{NS}x_S$. Then, the leader set S achieves structural controllability if (A_{NN}, A_{NS}) is structurally controllable, which would allow the values of x_N to be steered to desired positions. Note that it is not clear whether we would achieve structural controllability when this is chosen as the set of actuators in actuator placement problem.

From Definition 1, the actuator set S makes the system structurally controllable if and only if there exists a pair $(\hat{A}, \hat{B})$ with the same structure as $(A, B(S))$ such that the controllability matrix $P \in \mathbb{R}^{n \times n^2}$,

$$P = \begin{bmatrix} 0 & 0 & 0 & \hat{A}_{NS} & 0 & \hat{A}_{NN}\hat{A}_{NS} + \hat{A}_{NS}\hat{A}_{SS} & \cdots \\ 0 & I_{|S|} & 0 & \hat{A}_{SS} & 0 & \hat{A}_{SN}\hat{A}_{NS} + \hat{A}_{SS}^2 & \cdots \end{bmatrix},$$

has full rank. Next, we claim that P has full rank if and only if the following matrix $\tilde{P}_1 \in \mathbb{R}^{|N| \times n^2}$ has full rank, $\tilde{P}_1 = \begin{bmatrix} 0 & 0 & 0 & \hat{A}_{NS} & \cdots & 0 & \hat{A}_{NN}^{j-1}\hat{A}_{NS} & \cdots \end{bmatrix}$. To see this, notice that P has full rank if and only if the submatrix $P_1 \in \mathbb{R}^{|N| \times n^2}$ containing the first $|N|$ rows of P has full rank. One can then show that there exists an upper triangular matrix $U \in \mathbb{R}^{n^2 \times n^2}$ with unit diagonal entries such that $\tilde{P}_1 = P_1 U$. Since U is invertible, $\tilde{P}_1$ and P_1 have the same rank.

Then, we further claim that $\tilde{P}_1$ has full rank if and only if the following matrix $\bar{P}_1$ has full rank $\bar{P}_1 = \begin{bmatrix} \hat{A}_{NS} & \hat{A}_{NN}\hat{A}_{NS} & \cdots & \hat{A}_{NN}^{|N|-1}\hat{A}_{NS} \end{bmatrix}$. Considering $|S| > 0$ and thus $|N| - 1 \leq n - 2$, for any $i > |N| - 1$, $\hat{A}_{NN}^i\hat{A}_{NS}$ is in the span of the matrices $\hat{A}_{NN}^j\hat{A}_{NS}$, $j = \{0, 1, \ldots, |N| - 1\}$ by Cayley-Hamilton theorem. Hence, $\bar{P}_1$ has the same rank as $\tilde{P}_1$. This proves the claim.

In summary, P has full rank if and only if $\bar{P}_1$ has full rank. By the definition of $\bar{P}_1$, $\bar{P}_1$ being full rank is equivalent to controllability of $(\hat{A}_{NN}, \hat{A}_{NS})$. Hence, structural controllability of $(A, B(S))$ is equivalent to that of (A_{NN}, A_{NS}).

Now, define $\mathcal{L}_K = \{S \mid |S| = K \text{ and } (A_{NN}, A_{NS}) \text{ is structurally controllable}\}$ and conclude that $\mathcal{L}_K = \mathcal{C}_K$. The set collection $\mathcal{L}_K$ consists of all the K cardinality leader sets achieving structural controllability. From [23, Thm 4], we have that the pair $(V, \tilde{\mathcal{L}}_K)$, where $\tilde{\mathcal{L}}_K := \{\Omega \mid \exists\ S \in \mathcal{L}_K \text{ such that } \Omega \subset S\}$, is a matroid if the graph G is strongly connected. Therefore, the pair $(V, \tilde{\mathcal{C}}_K)$ is also a matroid. ∎

E. Proof of Theorem 1

The idea of the proof extends the work in [24], which derives a performance guarantee for matroid optimization featuring a submodular objective. To assess the suboptimality of the actuator set S^{f}, we need to find an upper bound for $f(S^*) - f(S^{\mathrm{f}})$. We denote $S^* = \{v_1^*, \ldots, v_K^*\}$ and notice

$$f(S^*) - f(S^{\mathrm{f}}) \leq f(S^* \cup S^{\mathrm{f}}) - f(S^{\mathrm{f}})$$
$$= \sum_{k=1}^{K} \rho_{v_k^*}(\{v_1^*, \ldots, v_{k-1}^*\} \cup S^{\mathrm{f}}) \leq \gamma^{-1} \sum_{j \in S^* \setminus S^{\mathrm{f}}} \rho_j(S^{\mathrm{f}}), \quad (29)$$

where the first inequality is due to the monotonicity of f and the equality follows from a telescoping sum. The last inequality is from Definition 2. To further bound $\sum_{j \in S^* \setminus S^{\mathrm{f}}} \rho_j(S^{\mathrm{f}})$, we have the following lemmas. For these lemmas, define $U^{-1} = \emptyset$, $U^K = V$, and $s_t = |S^* \cap (U^{t+1} \setminus U^t)|$.

Lemma 5: $\sum_{j \in S^* \setminus S^{\mathrm{f}}} \rho_j(S^{\mathrm{f}}) \leq \gamma^{-1} \sum_{t=1}^{K} \rho_{t-1} s_{t-1}$.

Proof: From Definition 2, we have

$$\rho_j(S^{\mathrm{f}}) \leq \gamma^{-1} \rho_j(S^{t-1}), \forall t \leq K, \forall j \in V. \quad (30)$$

Since $U^{t_1} \subset U^{t_2}$ for any $t_1 < t_2$, notice that $V = U^K = \bigcup_{t=0}^{K}(U^t \setminus U^{t-1})$. Considering $U^{t_1} \setminus U^{t_1-1}$ and $U^{t_2} \setminus U^{t_2-1}$ are disjoint, we know that these sets constitute a partition of V. Since there is no subset of U^0 belonging to $\mathcal{F}$, we have $S^* \cap U^0 = \emptyset$. Using the partition of V, we can partition S^* as: $S^* = \bigcup_{t=1}^{K}(S^* \cap (U^t \setminus U^{t-1}))$. Combining this with (30),

$$\sum_{j \in S^* \setminus S^{\mathrm{f}}} \rho_j(S^{\mathrm{f}}) \leq \sum_{j \in S^*} \rho_j(S^{\mathrm{f}}) = \sum_{t=1}^{K} \sum_{j \in S^* \cap (U^t \setminus U^{t-1})} \frac{1}{\gamma} \rho_j(S^{t-1}). \quad (31)$$

Notice that all the nodes in U^{t-1} have been considered by the feasibility check before v_t^{f}. Since the greedy algorithm first checks the elements in $V \setminus U^{t-1}$ with larger marginal gains when added to U^{t-1}, we have that $\rho_{t-1} = \max_{j \in V \setminus U^{t-1}} \rho_j(S^{t-1})$. Considering $V \setminus U^{t-1} = \cup_{i=t}^{K}(U^i \setminus U^{i-1})$, for any $t' \geq t$,

$$\rho_{t-1} \geq \rho_j(S^{t-1}), \forall j \in U^{t'} \setminus U^{t'-1}. \quad (32)$$

Thus, for any $j \in S^* \cap (U^t \setminus U^{t-1})$, we have $\rho_j(S^{t-1}) \leq \rho_{t-1}$ and

$$\sum_{j \in S^* \cap (U^t \setminus U^{t-1})} \rho_j(S^{t-1}) \leq \rho_{t-1} s_{t-1}. \quad (33)$$

Now combining (31) and (33), it is straightforward that $\sum_{j \in S^* \setminus S^{\mathrm{f}}} \rho_j(S^{\mathrm{f}}) \leq \sum_{t=1}^{K} \gamma^{-1} \rho_{t-1} s_{t-1}$. ∎

Lemma 6: For any $t \in \{1, \ldots, K\}$, we have $\sum_{i=1}^{t} s_{i-1} \leq t$.

The above lemma is proven by [24] for $\gamma = 1$, and it holds also when $\gamma \neq 1$ since its proof exploits only the matroid structure. Hence, the proof is omitted.

We use Lemma 6 to obtain an upper bound to the right-hand side of Lemma 5 and consequently to derive an upper bound of $f(S^*) - f(S^{\mathrm{f}})$. The following explains these steps.

Proof of Theorem 1: First, we consider the case in which ρ_i, $i = 0, \ldots, K-1$, are distinct. We define t_1 such that ρ_{t_1-1} is the largest among $\rho_0, \rho_1, \ldots, \rho_{K-1}$ and t_2 such that ρ_{t_2-1} is the largest among $\rho_{t_1}, \rho_{t_1+1}, \ldots, \rho_{K-1}$. Following the same pattern we have $t_1, t_2, \ldots, t_p$, where $t_p = K$. Since $s_i \geq 0$ is bounded by Lemma 6, to give an upper bound to the right-hand side of Lemma 5, we construct a linear program as follows,

$$\max_{s_0, \ldots, s_{K-1} \geq 0} \sum_{i=1}^{K} \rho_{i-1} s_{i-1} \text{ s.t. } \sum_{i=1}^{t} s_{i-1} \leq t, \ t = 1, \ldots, K. \quad (34)$$

Let s_{i-1}^*, $i = 1, 2, \ldots, K$, denote the optimal solution. We claim $s_{t_1-1}^* = t_1$. Otherwise, $s_{t_1-1}^* < t_1$ and due to Lemma 6 two situations might happen, a) $\sum_{i=1}^{t_1} s_{i-1}^* = t_1$ or b) $\sum_{i=1}^{t_1} s_{i-1}^* < t_1$.

For case a), we obtain $\sum_{i=1}^{t_1-1} s_{i-1}^* > 0$. It follows that there exists $l < t_1$ such that $s_{l-1}^* > 0$. Then, we decrease

s^*_{l-1} by $\delta > 0$ and increase $s^*_{t_1-1}$ also by δ. The value of δ is small enough so that $s^*_{l-1} > 0$. This operation decreases $\sum^t_{i=1} s^*_{i-1}$ for $l \le t \le t_1 - 1$ and keeps the sum unchanged for any other t, so the constraints of (34) are not violated. Also considering that $\rho^*_{t_1-1} > \rho^*_{l-1}$, after these changes, the objective function is strictly greater than the value obtained at the original optimum. Thus, case a) is impossible. For case b), we collect all the integers $l > t_1$ satisfying $s^*_{l-1} > 0$. Assume $l_q > \cdots > l_1 > t_1$. We have $q \ge 1$. Otherwise, $s^*_{l-1} = 0$ for any $l > t_1$ and we can increase $s^*_{t_1-1}$ by a small amount to obtain a greater value of the objective without violating constraints. Knowing that $s^*_{l_1-1} > 0$ and following the same reasoning provided for the case a), we increase $s^*_{t_1-1}$ and decrease $s^*_{l_1-1}$ with the same amount. This way, an objective value is obtained larger than that evaluated at the original optimum. Thus, case b) is impossible.

In conclusion, $s^*_{t_1-1} = t_1$ and (34) is equivalent to

$$\max_{s_{t_1},\ldots,s_{K-1} \ge 0} \sum^K_{i=t_1+1} \rho_{i-1} s_{i-1} \tag{35}$$
$$\text{s.t.} \sum^t_{i=t_1+1} s_{i-1} \le t - t_1, \ t = t_1+1,\ldots,K.$$

We determine $s^*_{t_2-1}$ in the same way as we determine $s^*_{t_1-1}$ in (34). By repeating the above procedure we obtain the solution

$$s^*_{i-1} = \begin{cases} t_1, & \text{if } i = t_1, \\ t_j - t_{j-1}, & \text{if } i = t_j \text{ and } j \ne 1, \\ 0, & \text{otherwise.} \end{cases} \tag{36}$$

If ρ_i, $i = 0,\ldots,K-1$ are not distinct and there exist $i_1 < i_2 < \cdots < i_q$ with $\rho_{i_1} = \rho_{i_2} = \cdots = \rho_{i_q}$. We let $s^*_{i_1} = \cdots = s^*_{i_{q-1}} = 0$ and obtain the same solution as (36). Next, notice

$$\rho_{i_2} = f(S^{i_2+1}) - f(S^{i_2}) \le \gamma^{-1}(f(S^{i_1} \cup v^{\mathsf{f}}_{i_2+1}) - f(S^{i_1})) \le \frac{\rho_{i_1}}{\gamma}, \tag{37}$$

where the first inequality comes from the definition of submodularity ratio, while the second is due to (32). Substituting the optimal solution into the objective, considering (37), we have

$$\sum^K_{i=1} \rho_{i-1} s^*_{i-1} = t_1 \rho_{t_1-1} + \cdots + (t_p - t_{p-1})\rho_{t_p-1}$$
$$\le \gamma^{-1} \sum^p_{k=1} \sum^{t_k}_{i=t_{k-1}+1} \rho_{i-1} = \gamma^{-1} \sum^K_{i=1} \rho_{i-1} = \gamma^{-1}(f(S^{\mathsf{f}}) - f(\emptyset)). \tag{38}$$

Combining (29), Lemma 5 and (38), we have $f(S^*) - f(S^{\mathsf{f}}) \le \gamma^{-1} \sum_{j \in S^* \setminus S^{\mathsf{f}}} \rho_j(S^{\mathsf{f}}) \le \gamma^{-2} \sum^K_{i=1} \rho_{i-1} s^*_{i-1} \le \gamma^{-3}(f(S^{\mathsf{f}}) - f(\emptyset))$. By rewriting this, we have $\frac{f(S^{\mathsf{f}}) - f(\emptyset)}{f(S^*) - f(\emptyset)} \ge \frac{\gamma^3}{\gamma^3+1}$. ∎

F. Proof of Proposition 2

Let $\rho_i(S) = -F_\epsilon(S \cup \{i\}) - (-F_\epsilon(S))$. Given S and U, we denote $Y = U \setminus S$ and $R = V \setminus (S \cup U)$. Notice that if $i \notin S \cup U$, $\frac{\rho_i(S)}{\rho_i(S \cup U)} = \frac{F^{\mathsf{r}}_\epsilon(R \cup Y) - F^{\mathsf{r}}_\epsilon(R \cup Y \setminus \{i\})}{F^{\mathsf{r}}_\epsilon(R) - F^{\mathsf{r}}_\epsilon(R \setminus \{i\})}$. From Definitions 2 and 3 we know that for all possible combinations of S and U, the left-hand side has the least upper bound $1/(1 - \alpha^{\mathsf{f}}_\epsilon)$ and the greatest lower bound $\gamma^{\mathsf{f}}_\epsilon$ while the right-hand side has the least upper bound $1/\gamma^{\mathsf{r}}_\epsilon$ and the greatest lower bound $1 - \alpha^{\mathsf{r}}_\epsilon$. Consequently, we obtain $\gamma^{\mathsf{r}}_\epsilon = 1 - \alpha^{\mathsf{f}}_\epsilon$ and $\alpha^{\mathsf{r}}_\epsilon = 1 - \gamma^{\mathsf{f}}_\epsilon$. ∎

G. Formulation of the maximum flow problem

Given a directed graph D with nodes s (the source) and t (the sink), denote the node set as $V(D)$ and the edge set as $E(D)$. Suppose no edge is directed into s or out of t. Let $c : E(D) \to \mathbb{R}_+$ assign to any edge (u,v) in $E(D)$ $c(u,v) \in \mathbb{R}_+$ called the capacity. Any $g : E(D) \to \mathbb{R}_+$ is called a flow in (D, c, s, t) if it satisfies: a) $\sum_u g(u,w) = \sum_v g(w,v)$ for any $w \in V(D) \setminus \{s,t\}$ and b) $g(u,v) \le c(u,v)$ for all $(u,v) \in E(D)$. The first is the balance, whereas the second is the capacity limits. $\sum_w g(s,w)$ is called the value of g denoted as $\mathrm{val}(g)$. The problem is to find the flow with the maximum value. For the undirected bipartite graph $\mathcal{H}_b(S)$, we direct all the edges from V to V' and place nodes s and t. Directed edges are built from s to all the nodes in $V \cup V''$ and from all the nodes in V' to t. Based on this D, we construct a capacity function such that $c(u,v) = 1$ for any $(u,v) \in E(D)$. The maximum value of a flow in (D, c, s, t) is equivalent to the cardinality of the maximum matching in $\mathcal{H}_b(S)$ [47].
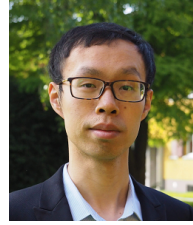
Baiwei Guo received his B.Sc. and M.Sc. degrees in Electrical Eng. respectively from Harbin Institute of Technology in 2017 and ETH Zürich in 2019. He is currently a Ph.D. candidate at the Automatic Control Laboratory at EPF Lausanne. His research interests include combinatorial optimization and statistical learning with applications in control theory.

Orcun Karaca received his B.Sc. degree in Electrical Eng. in 2014 from Boğaziçi Univ., Turkey, and his M.Sc. degree in Electrical Eng. and Information Technology in 2016 from ETH Zürich, Switzerland. He is currently a Ph.D. candidate at the Automatic Control Laboratory at ETH Zürich. His research interests include nonconvex optimization and game theory with applications in power grid systems and electricity markets.

Tyler Summers is an assistant professor at UT Dallas. He was a postdoctoral fellow at ETH in 2011-2015. He received a B.S. degree in Mech. Eng. from Texas Christian Univ. in 2004 and an M.S. and Ph.D. degree in Aerospace Eng. at UT Austin in 2007 and 2010. He was a Fulbright Postgraduate Scholar at the Australian National Univ. in 2007-2008. His research is on feedback control and optimization in complex dynamical networks, emphasizing theoretical tools and computational methods and driven by applications to power networks and robotics.

Maryam Kamgarpour is with the faculty of Electrical and Computer Engineering at the University of British Columbia. She holds a Doctor of Philosophy in Engineering from UC Berkeley and a Bachelor of Applied Sci. from Univ. of Waterloo. Her research is on safe decision-making and control under uncertainty, game theory and mechanism design, mixed integer and stochastic optimization and control. Her theoretical research is motivated by control challenges arising in intelligent transportation networks, robotics, power grid systems and healthcare. She is the recipient of NASA High Potential Individual Award, NASA Excellence in Publication Award, and the European Union (ERC) Starting Grant.