2021 60th IEEE Conference on Decision and Control (CDC)
December 13-15, 2021. Austin, Texas

Actuator Placement for Structural Controllability beyond Strong Connectivity and towards Robustness

Baiwei Guo, Orcun Karaca, Sepide Azhdari, Maryam Kamgarpour, Giancarlo Ferrari-Trecate

2021 60th IEEE Conference on Decision and Control (CDC) | 978-1-6654-3659-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/CDC45484.2021.9683002

Abstract— **Actuator placement is a fundamental problem in control design for large-scale networks. In this paper, we study the problem of finding a set of actuator positions by minimizing a given metric, while satisfying a structural controllability requirement and a constraint on the number of actuators. We first extend the classical forward greedy algorithm for applications to graphs that are not necessarily strongly connected. We then improve this greedy algorithm by extending its horizon. This is done by evaluating the actuator position set expansions at the further steps of the classical greedy algorithm. We prove that this new method attains a better performance, when this evaluation considers the final actuator position set. Moreover, we study the problem of minimal backup placements. The goal is to ensure that the system stays structurally controllable even when any of the selected actuators goes offline, with minimum number of backup actuators. We show that this problem is equivalent to the well-studied hitting set problem. Our results are verified by a numerical case study.**

I. Introduction

The steady progress in computation and communication technologies is enabling the deployment of large networks of systems, which require optimal and robust coordination. Prominent examples include power grids [1] and industrial control systems [2]. A significant amount of research is currently focusing on the control design for such networks in order to achieve better performance and security [3], [4]. A fundamental design problem concerns actuator placement which aims to select a subset from all possible actuator positions to place actuators such that a chosen network metric is minimized. Typical metrics are set functions that map the actuator positions to certain costs, such as controllability metrics [1] and LQG costs [5]. An optimal actuator placement can potentially bring in great amount of savings during long-term system operation. However, it is in general NP-hard to find the optimal set of actuators positions [6].

The sets minimizing these aforementioned metrics are not guaranteed to make the resulting system controllable [7]. In view of this issue, structural controllable systems constitute a desirable class. These systems are the ones that attain controllability after a slight perturbation of the system parameters corresponding to edge weights in the underlying network graph [8]. The concept of structural controllability is based only on the graphical interconnection structure of the dynamical system and the actuator positions. With structural controllability as a hard constraint, the work in [9] considers leader selection to minimize control errors due to noisy communication links and [7] places actuators to reduce the approximate control energy metric discussed in [6]. However, both studies assume strongly connected network graphs. In this case the corresponding problem can be formulated as a matroid optimization, allowing the use of efficient greedy heuristics. Hence, the first goal of this paper is to extend the non-modular metric minimization problem under structural controllability constraints to arbitrary graphs.

In [7], [9], Forward Greedy algorithm (**FG**) iteratively adds the most beneficial node to the leader or the actuator position sets under the constraints of structural controllability and a cardinality upper bound. This algorithm provides a solution that approximately minimizes the given metrics. However, the solution is not always satisfactory [10], [11]. Several variants, including the Continuous Greedy Algorithm [12], potentially enjoy better theoretical cost upperbounds than **FG** in a probabilistic sense. However, they do not offer any ex-post performance improvement. Thus, the second goal of this paper is to derive a heuristic algorithm which is guaranteed to perform at least as well as **FG** while maintaining its polynomial complexity.

Another issue in actuator placement is related to susceptibility of the actuators to faults. In extreme conditions, the actuators may go offline and thus may not be able to achieve the expected performance. Related research is focused on robust optimization for the worst-case scenario [13], [14] and efficient methods for contingency analysis [15]. To the best of our knowledge, no previous work considers maintaining structural observability/controllability in case of offline sensor/actuators. Hence, in this paper, the third goal is to study methods to deploy backup actuators to enhance robustness of the network for structural controllability.

Targeting the aforementioned goals, this paper[1] has three main contributions. First, for non-strongly-connected graphs, we propose a method to select actuators and, under certain conditions, guarantee the structural controllability. Second, based on **FG**, we propose a novel method, called Long-Horizon Forward Greedy Algorithm (**LHFG**). The performance of the actuator position set derived is guaranteed to be at least as well as that of **FG**. Third, we formulate

The work of B. Guo and G. Ferrari-Trecate received support from the Swiss National Science Foundation under the NCCR Automation (grant agreement 51NF40_180545)

B. Guo, S. Azhdari and G. Ferrari-Trecate are with Automatic Control Laboratory, Institute of Mechanical Engineering, École Polytechnique Fédérale de Lausanne, Switzerland, e-mails: {baiwei.guo, sepide.azhdari, giancarlo.ferraritrecate}@epfl.ch

O. Karaca is with ABB Corporate Research Center, Baden, Switzerland. email: orcun.karaca@ch.abb.com

M. Kamgarpour is with the Electrical and Computer Eng. at the University of British Columbia, Canada. email: maryamk@ece.ubc.ca

[1] There is a technical report for this paper, see [16]

the problem of minimal backup placements. With these backups, the system can maintain its structural controllability even a node failure happens. We show that this problem is equivalent to the hitting set problem.

Remarks on notation: We use v and $\{v\}$ interchangeably for singleton sets. Within pseudocodes of algorithms, we introduce several symbols for sets and variables to describe the steps of the algorithms. These notations will be used in the proofs.

II. Problem formulations and preliminaries

A. Problem formulations

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 $v_i \in V := \{v_1, \dots, v_n\}$. If we place actuators on the set of nodes $S \subset V$, called the actuator position set, and apply an input vector $u \in \mathbb{R}^n$, the system dynamics can be written as

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

where $B(S) = \mathrm{diag}(\mathbf{1}(S)) \in \mathbb{R}^{n\times n}$ and $\mathbf{1}(S)$ 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 weighted directed graph associated with the adjacency matrix A with nodes V and edges E, where $|E| = l$ and the directed edge $(v_j, v_i) \in E$ if the associated weight $(A)_{ij}$ is non-zero.

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 $P = \begin{bmatrix} B(S) & AB(S) & \cdots & A^{n-1}B(S) \end{bmatrix} \in \mathbb{R}^{n\times n^2}$. Due to potential errors in the identification of the edge weights, most of the time we can only rely on the topology but not on the particular weights. Motivated by this particularity, we introduce the weaker notion of structural controllability.

Definition 1: We say that (A, B) and $(\hat{A}, \hat{B})$ with $A, B, \hat{A}, \hat{B} \in \mathbb{R}^{n\times n}$ 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 shown in [8], structural controllability of the pair (A, B) further implies that even when (A, B) is not controllable, it is always possible to slightly perturb some non-zero edge weights to ensure controllability.

With this notion, the first problem we aim to solve is the following.

P1: Suppose a potentially non-modular and non-increasing metric $f : 2^V \to \mathbb{R}$ is given. Find K actuators attaining structural controllability while minimizing the metric at hand:

$$\begin{aligned} \min_{S\subset V, |S|=K} \quad & f(S) \\ \text{s.t.} \quad & (A, B(S)) \text{ is structurally controllable.} \end{aligned} \tag{2}$$

Suppose S is a feasible solution to **P1**, which could be either optimal or approximately optimal. The second problem considers the robustness of $(A, B(S))$ against failures, which will be further motivated in Section IV.

P2: Given S, find the minimal backup actuator positions such that structural controllability can be retained when any single actuator at $v \in S$ malfunctions:

$$\begin{aligned} \min_{\mathcal{B}\subset V} \quad & |\mathcal{B}| \\ \text{s.t.} \quad & \forall v \in S, \exists b_v \in \mathcal{B}, (A, B(\{S \setminus v\} \cup b_v)) \text{ is} \\ & \text{structurally controllable.} \end{aligned} \tag{3}$$

To define the constraints on these two problems, we need an efficient characterization of structural controllability.

B. Review: characterization of structural controllability

Structural controllabilty boils down to checking two graphical properties. A system $(A, B(S))$ is structurally controllable if and only if it satisfies *accessibility* and *dilation-freeness* [8, Theorem 1], which are defined as follow.

Definition 2: The system $(A, B(S))$ satisfies accessibility (or the nodes in V are accessible by S) if for any $v \in V$ there exists a path from a node in S to v in $G = (V, E)$.

Note that one can use Breadth First Search to verify accessibility condition. For dilation-freeness, we need a method to distinguish a node with an actuator from other nodes in the graph $G = (V, E)$. Let $S'' = \{v''_{s_1},, \dots, v''_{s_K}\}$ denote a copy of the actuator position set $S = \{v_{s_1}, \dots, v_{s_K}\}$ and E'' denote edges connecting v''_{s_i} to v_{s_i}, $i = 1, \dots, K$. Then, the graph $G'' = (V \cup S'', E \cup E'')$ illustrates explicitly how the actuators are connected to the nodes in V.

Definition 3: The system $(A, B(S))$ is dilation-free if in G'' the in-neighbor set $N(U) := \{\tilde{v} \in V \,|\, \exists v_u \in V \cup S'' \text{ s.t. } (\tilde{v}, v_u) \in E \cup E''\}$ for any subset $U \subset V$ satisfies $|N(U)| \geq |U|$.

In the following, we introduce methods for dilation-freeness checks via matchings in bipartite graphs [17], [18].

We define concepts related to 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 undirected edge in E connects a vertex in V^1 to another in V^2. A matching m is a subset of E where no two edges in m share a vertex in common. Given a subset $L \subset V^1 \cup V^2$, we say L is covered by m if any $v \in L$ is incident to an edge in m. The matching m is called maximum if it has the largest cardinality among all possible matchings and is called perfect if V^2 is covered.

We utilize an auxiliary bipartite graph to check dilation-freeness. It is constructed as follows. Let node sets $V' = \{v'_1, \dots, v'_n\}$ and $V'' = \{v''_1, \dots, v''_n\}$ be two copies of $V = \{v_1, \dots, v_n\}$. As for the edges, 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_S consists of undirected edges connecting v'_k with v''_k if $v_k \in S$. The auxiliary bipartite graph is then given by $\mathcal{H}_b(S) := (V \cup S'', V', \mathsf{E} \cup \mathsf{E}_S)$. With this graph at hand, the following lemma [7, Proposition 7] provides an efficient method to check dilation-freeness, where we need two definitions $\mathcal{C}_K := \{S \subset V \,|\, |S| = K$ and the pair $(A, B(S))$ is dilation-free$\}$ and $\tilde{\mathcal{C}}_K := \{\Omega \,|\, \exists S \in$

$\mathcal{C}_K$ such that $\Omega \subset S\}$

Lemma 1 (Dilation-freeness check): With $\bar{m}(S)$ as a maximum matching in $\mathcal{H}_b(S)$, we have

(i) $S \in \mathcal{C}_K$ if and only if $|S| = K$ and $\bar{m}(S) = n$,
(ii) $S \in \tilde{\mathcal{C}}_K$ if and only if $|S| \leq K$ and $\bar{m}(S) \geq n - K + |S|$.

Note that a maximum matching can be derived via the Edmonds-Karp Algorithm with complexity $O(nl^2)$ [19], where we recall $l = |E|$. Checking membership with respect to $\tilde{\mathcal{C}}_K$ is essential for implementing the greedy algorithm [20] to approximately solve **P1**. We will discuss this algorithm in Section II-C. An example illustrating dilation-freeness checks on a four-node system can be found in the technical report [16].

C. Forward greedy algorithm for structural controllability

A special instance of **P1** has been studied in [7], [9], where strong connectivity of G, the directed graph corresponding to the adjacency matrix A, is assumed. With this assumption, $(A, B(S))$ satisfies accessibility whenever $|S| > 0$. Consequently, if $|S| \neq 0$, dilation-freeness implies structural controllability. **P1** is then reduced to minimization of $f(S)$ with the constraint $S \in \tilde{\mathcal{C}}_K$, which is a matroid optimization problem [7].

The study in [7] proposes to apply the Forward Greedy Algorithm (**FG**) through the function $\mathrm{FG}(S^0, d)$ shown in Algorithm 1. Among the inputs, S^0 denotes the initial set before expansion and d, called the *depth*, denotes an upper bound for the number of expansions. In the pseudocode of Algorithm 1, the function IsMember is implemented using the efficient graph-theoretical techniques introduced in Section II-B. This algorithm can be justified by the following.

Lemma 2 ([7]): If G is strongly connected, with $d = K - |S_0|$ and $S^0 \in \tilde{\mathcal{C}}_K$, $S_\mathsf{f} = \mathrm{FG}(S^0, d)$ satisfies $S_\mathsf{f} \in \mathcal{C}_K$.

The proof of this result relies on the strong connectivity of G. If the digraph G is not strongly connected, dilation-freeness of $(A, B(S))$ does not necessarily lead to structural controllability. In view of this, the greedy solution may fail to attain structural controllability.

In Section III below, targeting **P1**, we 1) extend **FG** beyond the assumption of strong connectivity by providing an efficient method to attain the accessibility condition and 2) propose a novel algorithm based on **FG** while ensuring a better performance. In Section IV, we address **P2**.

III. Extensions to the greedy heuristics

A. Beyond strong connectivity

In case the graph $G = (V, E)$ is not strongly connected, we consider finding an initial set S_0 before running **FG** such that all the nodes in V are accessible by S_0.

We start by highlighting that the well-established Kosaraju's algorithm [17] can find all the strongly connected components in the directed graph G in linear time, that is, with complexity $O(n + l)$, where l is the number of edges in G. Whenever G is not strongly connected, there would be components with no incoming edges and we denote them as $\mathcal{T}^1, \dots, \mathcal{T}^{n_\mathcal{T}}$. If all the nodes in V are accessible by a set S, then for any $1 \leq i \leq n_\mathcal{T}$ the component $\mathcal{T}^i$ guarantees that $\mathcal{T}^i \cap S \neq \emptyset$, otherwise the nodes in $\mathcal{T}^i$ would not be accessible. Based on this idea, we propose Algorithm 2 to construct S^0 when G is not strongly connected.

Algorithm 1 Forward Greedy Algorithm

```
Input: initial set $S^0$ and depth $d$
Output: full actuator position set $S_\mathsf{f}$
  function FG($S^0, d$)
      $U^0 = V \setminus S^0$, $t = 1$
      while $U^{t-1} \neq V$ , $|S^{t-1}| < K$ and $t \leq d$ do
          $v^* = \arg\max_{i \in V \setminus U^{t-1}} f(S^{t-1}) - f(S^{t-1} \cup i)$
          if IsMember($S^{t-1} \cup v^*, \tilde{\mathcal{C}}_K$) then
              $v_\mathsf{f}^t \leftarrow v^*$
              $S^t \leftarrow S^{t-1} \cup v_\mathsf{f}^t$ and $U^t \leftarrow U^{t-1} \cup v_\mathsf{f}^t$
              $t \leftarrow t + 1$
          else
              $U^{t-1} \leftarrow U^{t-1} \cup v^*$
          end if
      end while
      $S_\mathsf{f} \leftarrow S^{t-1}$
  end function
```

Algorithm 2 Initial Set Construction

```
Input: graph $G = (V, E)$
Output: an actuator position set $S^0$
  function IniS($G$)
      Find all strongly connected components with no incoming edges, denoted as $\mathcal{T}^i = \{v_1^i \dots, v_{p_i}^i\}$ for any $1 \leq i \leq n_\mathcal{T}$ using Kosaraju's algorithm
      $S^{0,0} = \emptyset$
      for $t = 1, \dots, n_\mathcal{T}$ do
          Find $\mathcal{P}^t = \{v \in \mathcal{T}^t | S^{0,t-1} \cup v \in \tilde{\mathcal{C}}_K\}$.
          $v_{j_*^t}^t = \arg\min_{v_j^t \in \mathcal{P}^t} f(S^{0,t-1} \cup v_j^t)$
          $S^{0,t} \leftarrow S^{0,t-1} \cup v_{j_*^t}^t$
      end for
      $S^0 \leftarrow S^{0,n_\mathcal{T}}$
  end function
```

The following proposition proves that under certain conditions one can use Algorithm 1 to further expand S^0 such that the derived system is structurally controllable.

Proposition 1: Let k be the smallest integer such that $\mathcal{C}_k \neq \emptyset$. If $n_\mathcal{T}$ satisfies $n > K \geq k + n_\mathcal{T}$, then

(i) the set S^0 derived through Algorithm 2 belongs to $\tilde{\mathcal{C}}_K$,
(ii) By applying Algorithm 1 to further expand S^0 and obtaining $S_\mathsf{f} = \mathrm{FG}(S^0, K - |S^0|)$, the resulting system $(A, B(S_\mathsf{f}))$ is structurally controllable.

The proof can be found in the technical report [16]. In case the expenditure for deploying actuators is low and thus a large number of actuators is allowed, the condition "$n > K \geq k + n_\mathcal{T}$" is not restrictive. We should also notice that this condition is not necessary. Even if it does not hold, we may still obtain $S^0 \in \tilde{\mathcal{C}}_K$. In this case we can use **FG** to expand S^0.

To summarize, in case G is not strongly connected, our overall method for actuator placement is to first run

Algorithm 3 Long-horizon Greedy Algorithm

```
Input: actuator position set $S^0$ and horizon $d_\mathrm{h}$
Output: an actuator position set $S_\mathsf{lf}$
  function LHFG($S^0, d_\mathrm{h}$)
      $t \leftarrow 1$ and $S_\mathsf{lf}^0 = S^0$
      while $|S_\mathsf{lf}^{t-1}| < K$ do
          Find $\mathcal{R}^t = \{v \mid \textsc{IsMember}(S_\mathsf{lf}^{t-1} \cup v, \tilde{\mathcal{C}}_K) = 1\}$
          Calculate by enumeration $v_\mathsf{lf}^t = \arg\min_{v \in \mathcal{R}^t} f(\mathrm{FG}(S_\mathsf{lf}^{t-1} \cup v, d_\mathrm{h}))$
          $S_\mathsf{lf}^t \leftarrow S_\mathsf{lf}^{t-1} \cup v_\mathsf{lf}^t$ and $t \leftarrow t+1$
      end while
      $S_\mathsf{lf} \leftarrow S_\mathsf{lf}^{t-1}$
  end function
```

Algorithm 2 to obtain the initial set $S^0 = \textsc{IniS}(G)$ and then to execute Algorithm 1 to obtain the actuator position set $S_\mathsf{f} = \mathrm{FG}(S^0, K - |S^0|)$. Since this procedure is not the standard greedy algorithm, the upper bounds for the suboptimality gaps in previous works are not applicable. We leave deriving a guarantee towards the performance of the actuator position set $\mathrm{FG}(\textsc{IniS}(G), K - |S_0|)$ as a future work.

B. Beyond myopic decisions

This section derives a method that improves upon **FG** in terms of the performance of the final actuator position set.

For the iterations in Algorithm 1, the marginally most beneficial nodes, $v_\mathsf{f}^1, \ldots, v_\mathsf{f}^{K-|S^0|}$, are added one after the other. These decisions are myopic, since a node added at some iteration might make it harder to further decrease the given metric in later iterations. To mitigate this issue, we equip **FG** with a long horizon, as shown in Algorithm 3. This way, we can evaluate whether to add a node based on how the addition of this node influences the actuator position set expansion in the future. We call this new method Long-Horizon Greedy Algorithm (**LHFG**). We achieve the longer horizon by using **FG** to further expand starting from the node we are considering. The depth of this embedded **FG** is called the horizon of **LHFG**.

The following proposition proves that with a horizon long enough, **LHFG** has a performance no worse than **FG**.

Proposition 2: For an arbitrary network, if $S^0 \in \tilde{\mathcal{C}}_K$, $f(S_\mathsf{lf}) \leq f(S_\mathsf{f})$, where $S_\mathsf{lf} = \mathrm{LHFG}(S^0, K - |S^0|)$ and $S_\mathsf{f} = \mathrm{FG}(S^0, K - |S^0|)$.

Proof: Following the proof of Proposition 1 regarding dilation-freeness, one can see that S_lf and S_f are both well-defined and belong to the set $\mathcal{C}_K$. Next, we show S_lf achieves a metric value no more than that achieved by S_lf.

By denoting $\bar{S}_\mathsf{lf}^t := \mathrm{FG}(S_\mathsf{lf}^t, K - |S^0|)$, we claim that $f(\bar{S}_\mathsf{lf}^t) \geq f(\bar{S}_\mathsf{lf}^{t+1})$ for $t < K - |S_0|$. To prove this, we notice that $\bar{S}_\mathsf{lf}^t \setminus S_\mathsf{lf}^t \subset \mathcal{R}^{t+1}$ and there exists $\tilde{v} \in \bar{S}_\mathsf{lf}^t \setminus S_\mathsf{lf}^t$ such that $\tilde{v} \in \mathcal{R}^{t+1}$ and $\bar{S}_\mathsf{lf}^t = \mathrm{FG}(S_\mathsf{lf}^t \cup \tilde{v}, K - |S^0|)$, therefore, $f(\bar{S}_\mathsf{lf}^{t+1}) = \arg\min_{v \in \mathcal{R}^{t+1}} f(\mathrm{FG}(S_\mathsf{lf}^t \cup v, K - |S^0|)) \leq f(\mathrm{FG}(S_\mathsf{lf}^t \cup \tilde{v}, K - |S^0|)) = f(\bar{S}_\mathsf{lf}^t)$.

To proceed, we recall that $v_\mathsf{lf}^1, \ldots, v_\mathsf{lf}^{K-|S_0|}$, in order, are added to form S_lf while $v_\mathsf{f}^1, \ldots, v_\mathsf{f}^{K-|S_0|}$, in order, are added to form S_f. If i is the smallest integer such that $v_\mathsf{lf}^i \neq v_\mathsf{f}^i$, $f(S_\mathsf{f}) = f(\bar{S}_\mathsf{lf}^{i-1}) \geq f(\mathrm{FG}(S_\mathsf{lf}^{i-1} \cup v_\mathsf{lf}^i, K - |S^0|)) = f(\bar{S}_\mathsf{lf}^i) \geq \bar{S}_\mathsf{lf}^{K-|S^0|}$. By noticing that $S_\mathsf{lf} = \bar{S}_\mathsf{lf}^{K-|S^0|}$, we have $f(S_\mathsf{lf}) \leq f(S_\mathsf{f})$. ■

We cannot guarantee that $f(S_\mathsf{lf}) < f(S_\mathsf{f})$ because of two potential cases, i) $S_\mathsf{lf} = S_\mathsf{f}$ and ii) $f(S_\mathsf{lf}) = f(S_\mathsf{f})$ even if $S_\mathsf{lf} \neq S_\mathsf{f}$. We refer the readers to Section V for a numerical case study where $f(S_\mathsf{lf})$ is significantly smaller than $f(S_\mathsf{f})$.

Notice that to derive $S_\mathsf{lf} = \mathrm{LHFG}(\textsc{IniS}(G), K - |S_0|)$ most of the computational time would be spent on long horizon evaluations. During these evaluations, we execute **FG** for at most $K(2n - K + 1)/2$ times. Suppose the computation of $f(S)$ takes $q(n)$ floating-point operations. Note that the complexity of ISMEMBER is $\mathcal{O}(nl^2)$ from Edmonds-Karp algorithm. One can then verify that the complexity of **LHFG** is $\mathcal{O}(Kn^3(q(n) + l^2))$. To reduce the computational complexity, one can shorten the horizon d_h of **LHFG**. However, it is then not possible to guarantee better performance than **FG**, as in the proposition above. In Section V, we study the computational time and the derived actuator sets in a numerical example for $d_\mathrm{h} < K - |S_0|$.

IV. Backup placements for ensuring structural controllability in response to failures

In many applications, the selected actuators S, which we call *primary*, derived through **LHFG** may be offline due to potential damages/failure. An offline primary actuator can potentially make the system uncontrollable, which would be unacceptable. In this section, we first list two assumptions that motivate **P2**. We then show that **P2** is equivalent to the hitting set problem, which has been well studied in the combinatorial optimization literature.

Assumption 1: Only one actuator at a time can be offline.

This assumption can hold if primary actuator failures are not frequent and/or offline actuators can be restored quickly. If several actuators can go offline at the same time, at the current stage of our research, we need to enumerate all possible combinations of failures and it would be quite conservative to deploy backups for the worst case scenario.

Assumption 2: There exists at least one actuator v such that $(A, B(S \setminus v))$ is not structurally controllable.

A primary actuator is called *essential* if its failure violates the structural controllability. It is necessary to have backups for essential primary actuators. Under these two assumptions, solving **P2** (see Section II-A) gives us the minimal backup position set $\mathcal{B}$. With these backup actuators of $\mathcal{B}$, we can still retain structural controllability by replacing any single offline primary actuator.

To solve **P2**, we characterize its constraint set in a tractable manner. For this purpose, following definitions are in order.

Definition 4: Given S such that $(A, B(S))$ is structurally controllable, we say v is a DFR (dilation-freeness-recovering) backup position for $v_\mathrm{off} \in S$ if $(A, B(S \setminus v_\mathrm{off} \cup v))$ is dilation-free. We say v is a feasible backup position for v_off if $(A, B(S \setminus v_\mathrm{off} \cup v))$ is structurally controllable.

We will now provide a tractable characterization of the feasible backup positions for a given v_off. To start with, consider the DFR backup positions. Recall that $S \in \mathcal{C}_K$ if and only if there exists a perfect matching m_0 in the bipartite graph $\mathcal{H}_b(S)$. With the primary actuator at v_{off} going offline, checking whether node v is a DFR backup position is equivalent to checking whether there exists a perfect matching in $\mathcal{H}_b(S \setminus v_{\text{off}} \cup v)$. For any $v \in V$, one can run the Edmonds-Karp Algorithm on $\mathcal{H}_b(S \setminus v_{\text{off}} \cup v)$. By iteratively doing so, we can obtain all DFR backup positions for v_{off}, with the complexity of $n^2 l^2$. Such a naive approach fails to exploit the properties of matchings in bipartite graphs.

The following theorem characterizes all the DFR backup positions for v_{off} in a computationally efficient way.

Theorem 1: Let $S \in \mathcal{C}_K$ be the set of actuators. Suppose $v_{\text{off}} \in S$ is offline and there does not exist a perfect matching in $H_1 := \mathcal{H}_b(S \setminus v_{\text{off}})$. A node $v \neq v_{\text{off}}$ is DFR for v_{off} if and only if there exist a perfect matching m_0 in $\mathcal{H}_b(S)$ and an alternating path $p = (v'_{\text{off}}, v_{p_1}, v'_{p_2}, \ldots, v_{p_r}, v')$ where edges $(v'_{\text{off}}, v_{p_1}), (v'_{p_2}, v_{p_3}), \ldots, (v'_{p_{r-1}}, v_{p_r}) \notin m_0$, $(v_{p_1}, v'_{p_2}), \ldots, (v_{p_r}, v') \in m_0$.

Proof: For both parts, observe that since there does not exist a perfect matching in $\mathcal{H}_b(S \setminus v_{\text{off}})$, the matching m_0 must contain the edge $(v''_{\text{off}}, v'_{\text{off}})$.

Sufficiency: By utilizing the alternating path p, we exclude from m_0 the edges $(v''_{\text{off}}, v'_{\text{off}})$, $(v_{p_1}, v'_{p_2}), \ldots, (v_{p_r}, v')$ and include $(v'_{\text{off}}, v_{p_1}), (v'_{p_2}, v_{p_3}), \ldots, (v'_{p_{r-1}}, v_{p_r}), (v', v'')$ to form a new edge set m_2. One can verify that all the edges in m_2 are contained in $H_2 := \mathcal{H}_b(S \setminus v_{\text{off}} \cup v)$ and m_2 is a perfect matching in H_2, in other words, v is a DFR backup for v_{off}.

Necessity: Since v is a DFR backup position for v_{off}, there exists a perfect matching in H_2. In this bipartite graph with the non-perfect matching $m_0 \setminus (v''_{\text{off}}, v'_{\text{off}})$, there exists an augmenting path p_α and by augmentation on this path one can obtain in H_2 a perfect matching [17], denoted as m_α. Due to the fact that there does not exist a perfect matching in H_1, the node v'' must be incident to p_α. Moreover, the node v'_{off} is also incident to p_α, otherwise the perfect matching m_α formed by augmentation cannot cover v'_{off} which contradicts the perfectness. We trim p_α to form p_β such that p_β only contains the part of p_α between v'_{off} and v'. One can verify that p_β is an alternating path in $\mathcal{H}_b(S)$ with respect to the matching $m_0 \setminus (v''_{\text{off}}, v'_{\text{off}})$ and satisfies the characterization specified in the theorem. ■

Remark: If the assumptions of Theorem 1 do not hold, i.e., there exists a perfect matching in H_1, the actuator v_{off} going offline does not affect the dilation-freeness property. Hence, this actuator is not essential and any node can be a DFR backup position.

To find all the alternating paths described in Theorem 1, we can use Breadth-First Search, whose complexity is $O(n+l)$. This allows us to efficiently find the DFR backup position set $\mathcal{D}(v)$, for any $v \in V$. Note that $v \in \mathcal{D}(v)$.

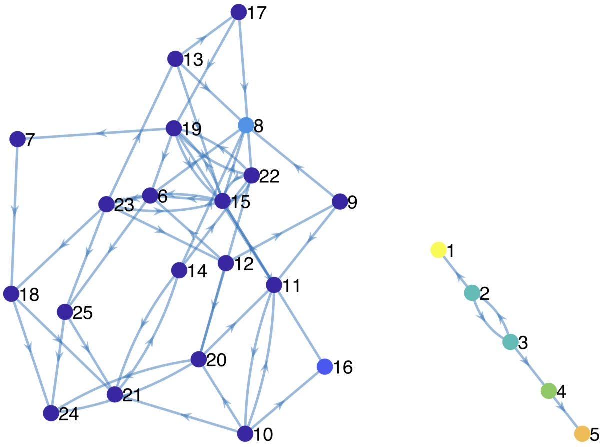

Fig. 1. Digraph G under consideration: strongly connected components are marked by different colors

It is then tractable to derive the feasible backup position set $\mathcal{F}(v_{\text{off}})$ from the DFR backup position sets. It holds that $\mathcal{F}(v_{\text{off}}) = \mathcal{D}(v_{\text{off}})$ except for the following situation. Recall that through Algorithm 2 we derive the strongly connected components with no incoming edges $\mathcal{T}^1, \ldots, \mathcal{T}^{n_\mathcal{T}}$. If there exists $i \in \mathbb{Z}^+$ such that $v_{\text{off}} \in \mathcal{T}^i$ and $S \cap \mathcal{T}^i = v_{\text{off}}$, then the actuator at v_{off} is the only one in $\mathcal{T}^i$. In case it goes offline, the system $(A, B(S \setminus v_{\text{off}}))$ no longer satisfies accessibility. For this specific v_{off}, we should have $\mathcal{F}(v_{\text{off}}) = \mathcal{D}(v_{\text{off}}) \cap \mathcal{T}^i$.

With the feasible backup position sets $\mathcal{F}(v_1), \ldots, \mathcal{F}(v_K)$ obtained via Theorem 1, **P2** can be reformulated as follows.

Corollary 1: Given the primary actuator positions S, the optimization problem **P2** is equivalent to the following:

$$\begin{aligned} \min_{\mathcal{B} \subset V} \quad & |\mathcal{B}| \\ \text{s.t.} \quad & \forall v \in S, \exists b_v \in \mathcal{B}, \text{ with } b_v \in \mathcal{F}(v). \end{aligned} \tag{4}$$

This is the classical hitting set problem [21], which is well-known to be NP-hard. There are extensive studies proposing efficient approximate solutions with provable approximation ratios, e.g., the LP-based approach in [22] and the randomized algorithm in [23].

As a summary, our overall method is to first find the primary actuators using the methods in Section III. We then construct the feasible backup position sets for the essential ones by using the equivalent condition in Theorem 1. Finally, we solve the hitting set problem.

V. A NUMERICAL CASE STUDY

We test our algorithms on a linear system whose system matrix A corresponds to the digraph G illustrated in Figure 1.[2] In this graph, the edge weights are set to be all 1 and there are 7 strongly connected components.

The metric under investigation is an approximate controllability metric $f = F_\epsilon$, where $F_\epsilon(S) = \text{tr}((W_T(S) + \epsilon I)^{-1})$, and $W_T(S) = \int_0^T e^{A\tau} B(S) B^\top(S) e^{A^\top \tau} d\tau$ is the controllability Grammian. This metric measures the average energy required for steering the system from x_0 with $||x_0||_2 = 1$ at $t = 0$ to zero state at $t = T$. The constant term ϵI allows for **FG** and **LHFG** to evaluate an uncontrollable actuator position set $S^0 \notin \mathcal{C}_K$ [7], [13]. We let $\epsilon = 10^{-12}$ and

[2] The code for this numerical case study is publicly available at https://github.com/odetojsmith/Actuator-Placement-beyond-SC-and-towards-Robustness

$T = 1$. Our goal is to find an actuator set S with cardinality 8 that minimizes the metric $F_\epsilon(S)$ while ensuring structural controllability of the resulting system. In practice, one can choose a suitable cardinality by a trade-off between actuator installation costs and the cost of system operations.

The initial actuator position set derived through Algorithm 2 is $S^0 = \{16, 2\}$, which is straightforward from the observation that, apart from the big strongly connected component consisting of nodes from 6 to 25, the node set $\{2, 3\}$ is the only strongly connected component without incoming edges.

The actuator position sets derived through the forward greedy algorithm and the long-horizon greedy algoithm are respectively $S_{\mathsf{f}} = \mathrm{FG}(S^0, \infty) = \{16, 2, 8, 18, 11, 3, 12, 5, 1\}$ and $S_{\mathsf{lf}} = \mathrm{LHFG}(S^0,\ \infty) = \{16, 2, 1, 13, 5, 8, 24, 14, 18\}$. The nodes in the sets follow the order they are selected by the algoirthm. The metrics associated with these two sets are $F_\epsilon(S_{\mathsf{f}}) = 7.56 \times 10^6$ and $F_\epsilon(S_{\mathsf{lf}}) = 1.08 \times 10^5$. **LHFG** gives 98.6% improvement. The computation time is 1.3 seconds for deriving S_{f}, whereas 84.7 seconds for deriving S_{lf}.[3] To reduce the computational complexity, we let the horizon of **LHFG** be $d_{\mathrm{h}} = 3$ and obtain $S'_{\mathsf{lf}} = \mathrm{LHFG}(S^0, 3) = \{16,\ 2, 25, 1, 12, 5, 8, 20, 24\}$ with $F_\epsilon(S'_{\mathsf{lf}}) = 1.34 \times 10^5$. As discussed before, in this case, we have no performance guarantees. The computation of S'_{lf} takes 64.6 seconds.

By checking the structural controllability when a single primary actuator gets offline, we find that the actuators at Node 1 and Node 2 are essential and they require backups. The feasible backup position sets are derived as $\mathcal{G}(v_1) = \{v_1, v_3\}$ and $\mathcal{G}(v_2) = \{v_2\}$. Thus, for this system, one can select the backup position set as $\{v_2, v_3\}$. Suppose we place primary actuators at S_{lf} and v_1 goes offline. By activating the backup actuators at v_3, the metric is 1.07×10^5. As a remark, in this example, the minimum number of actuators for structural controllability is $k = 5$, that is, less than $K = 8$. Even $K > k$ there are essential actuators, whose failures can result in the loss of structural controllability. Thus, it is necessary to detect essential actuators and deploy backups.

VI. Conclusion

In this paper, we studied the actuator placement problem minimizing a nonsubmodular and a nonsupermodular metric under the constraint of structural controllability. We extended the forward greedy algorithm (**FG**) to be applicable to arbitrary graphs and we proposed a novel algorithm, **LHFG**, which was proven to outperform **FG**. Then, to achieve robustness, we studied the minimal backup actuator placement problem and we showed that it is equivalent to the NP-hard hitting set problem.

Our future work will focus on improving the computational complexity of **LHFG** and studying backup placement problem in case several primary actuators can go offline, simultaneously.

[3] The case study is executed in Matlab R2019b on a computer equipped with 32GB RAM and a 2.3GHz Intel Core i9 processor.

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] 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.

[3] T. L. Magnanti and R. T. Wong, "Network design and transportation planning: Models and algorithms," *Transportation science*, vol. 18, no. 1, pp. 1–55, 1984.

[4] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, "Power awareness in network design and routing," in *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*. IEEE, 2008, pp. 457–465.

[5] V. Tzoumas, L. Carlone, G. J. Pappas, and A. Jadbabaie, "Lqg control and sensing co-design," *IEEE Transactions on Automatic Control*, 2020.

[6] 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.

[7] B. Guo, O. Karaca, T. Summers, and M. Kamgarpour, "Actuator placement under structural controllability using forward and reverse greedy algorithms," *IEEE Transactions on Automatic Control*, 2020.

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

[9] 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.

[10] D. Tihanyi, Y. Lu, O. Karaca, and M. Kamgarpour, "Multi-robot task allocation for safe planning under dynamic uncertainties," *arXiv preprint arXiv:2103.01840*, 2021.

[11] O. Karaca, D. Tihanyi, and M. Kamgarpour, "Performance guarantees of forward and reverse greedy algorithms for minimizing nonsupermodular nonsubmodular functions on a matroid," *arXiv preprint arXiv:2103.01135*, 2021.

[12] M. Sviridenko, J. Vondrák, and J. Ward, "Optimal approximation for submodular and supermodular optimization with bounded curvature," 2014.

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

[14] Q. Hou and A. Clark, "Robust maximization of correlated submodular functions under cardinality and matroid constraints," *IEEE Transactions on Automatic Control*, 2021.

[15] Z. Chu and F. Teng, "Short circuit current constrained uc in high ibg-penetrated power systems," *arXiv preprint arXiv:2101.04918*, 2021.

[16] B. Guo, O. Karaca, S. Azhdari, M. Kamgarpour, and G. Ferrari-Trecate, "Actuator placement for structural controllability beyond strong connectivity and towards robustness," *arXiv preprint arXiv:2104.01850*, 2021.

[17] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2009.

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

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

[20] 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.

[21] E. Angel, E. Bampis, and L. Gourvès, "On the minimum hitting set of bundles problem," *Theoretical computer science*, vol. 410, no. 45, pp. 4534–4542, 2009.

[22] M. Krivelevich, "Approximate set covering in uniform hypergraphs," *Journal of Algorithms*, vol. 25, no. 1, pp. 118–143, 1997.

[23] M. El Ouali, H. Fohlin, and A. Srivastav, "A randomised approximation algorithm for the hitting set problem," *Theoretical Computer Science*, vol. 555, pp. 23–34, 2014.