Structural Controllability Recovery via the Minimum-edge Addition

Shuo Zhang[1] Stephen D. Wolthusen [2]

***Abstract*— Identifying a set of inputs is a way to recover structural controllability of a structurally uncontrollable system, but it is meaningless if recovery needs more number of inputs than that of actually valid ones. Given a structurally uncontrollable system with given inputs, we recover its structural controllability. By graph-theoretical conditions of a structurally controllable system, we add a minimum set of edges into a digraph that represents the given system via its one maximum matching, so that the final digraph represents a structurally controllable system. Compared with the existing edge-addition method, for the worst-case execution time, our minimum edge-addition can be done in more efficient polynomial time.**

I. Introduction

Efficient recovery of structural controllability is necessary to enhance resilience of control systems and defend against control hijack [1], where attackers maliciously effect some system components to force system along with their purposes if the current system is out of control. Although identifying a set of inputs is a way to recover structural controllability of structurally uncontrolable systems [2], [3], after severe attack or failure on a structurally controllable system, recovery might need more number of inputs, than the actually available ones. Hence, it is no longer useful to only identify a set of inputs in order to still structurally control the residual system. Clearly, it is thus essential to concern practical constrains on inputs during the recovery of structural control into the residual system.

Therefore, given a structurally uncontrollable linear time-invariant (LTI) system, we efficiently recover its structural controllability with given inputs, where the input matrix of this given system is always fixed during recovery. Based on graph-theoretical conditions of structural controllability [4], [5], because the topology of LTI system can be represented by a digraph, which is called the system network in this paper, and the system network of a structurally controllable system contains a set of disjoint cacti [4]. We thus add a minimum set of edges into the system network of the given structurally uncontrollable system, to eventually construct a digraph spanned by disjoint cacti, which represents a structurally controllable system. In terms of constructing a structurally controllable system with given inputs, our problem can be solved by the existing edge-addition scenario [6], while it is low efficient in the worst-case execution time, and we are also motived to raise a minimum edge-addition scenario with higher efficiency.

To effectively guide the edge addition and ensure that the final digraph is spanned by disjoint cacti, given a system network of a structurally uncontrollable system, based on a maximum matching of it, we raise an edge-addition scenario of two steps, which are designed to detect and remove the dilation and inaccessible vertices [4] within this given system network. According to this scenario, when added edges can be reduced by the most in number, is further discussed. We conclude that the number of added edges in the first step is a constant value for the given system network. And the number of added edges in the second step is various. It is thus possible to achieve the minimum-edge addition for the given system network. As a result, our scenario can eventually construct a digraph spanned by a set of disjoint cacti, and the time complexity is the same as that of identifying a maximum matching of this system network. For our contribution, given a structurally uncontrollable system with given inputs, let m, n be the number of edges and vertices of the system network, the worst-case execution time of our minimum-edge addition to recover structural controllability is $O(\sqrt{n} \cdot m)$, which is more efficient than the edge-addition approach of [6], whose worst-time execution time is $O(n^3)$.

In the following paper, section II introduces the structural controllability of a LTI system; section III illustrates related works; section IV constructs a set of disjoint cacti, and the last section concludes this paper.

II. Structural controllability

According to the control theory [7], [8], a linear time-invariant (LTI) system can be expressed by a so-called state equation:

$$\dot{x}(t) = \mathbf{A}x(t) + \mathbf{B}u(t) \tag{1}$$

where $x(t) = (x_1(t), x_2(t), \ldots, x_N(t))^T$, $x(t) \in \mathbb{R}^N$ is the state vector, capturing the state of each system component at time t. $u(t) = (u_1(t), u_2(t), \ldots, u_M(t))^T (M \leq N)$, $u(t) \in \mathbb{R}^M$ is the input vector, holding external inputs at time t. $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the state matrix and shows the interaction among N system components, while input matrix $\mathbf{B} \in \mathbb{R}^{N \times M}$ shows interactions among N system components and M inputs. A system described by equation 1 is controllable if and only if the matrix $\mathbf{C} = [\mathbf{B}, \mathbf{AB}, \mathbf{A}^2\mathbf{B}, \ldots, \mathbf{A}^{N-1}\mathbf{B}]$ ($\mathbf{C} \in \mathbb{R}^{N \times N \cdot M}$), has full rank, noted by $rank(\mathbf{C}) = N$, and called the controllability rank condition.

However, exact value of non-zero entries of $\mathbf{A}$, $\mathbf{B}$ [4] is difficult to measure, and calculating the rank of $\mathbf{C}$ is

[1] Shuo Zhang is with School of Mathematics and Information Security, Royal Holloway, University of London, Egham TW20 0EX, UK MYVA375@live.rhul.ac.uk

[2] Stephen D. Wolthusen with School of Mathematics and Information Security, Royal Holloway, University of London, Egham TW20 0EX, UK stephen.wolthusen@rhul.ac.uk

with $O(2^N)$ time complexity [9]. Thus, controllability of the structured system is concerned [10], where only poisitions of zero and non-zero entries of $\mathbf{A}$ and $\mathbf{B}$ are concerned. If a system's state equation has exact value of non-zero entries, it is called an instance of a strucutred system [11]. Lin *et al*. [4] [12] defined the structural controllability below in order to derive a completely controllable system by using the structured system, when exact values of non-zero entries of both input and state matrices are unknown.

Definition 1 (**Structural Controllability** [4]). *A system described by equation 1 is structurally controllable if and only if there is at least one instance of it satisfying the controllability rank condition.*

Simultaneously, conditions of structurally controllable system are clarified by theorem 1 below with the following definition 2-6:

Definition 2 (**System Network**). *Given $\boldsymbol{A}$, $\boldsymbol{B}$ of equation 1, let $G(\boldsymbol{A},\boldsymbol{B}) = (V_1 \cup V_2, E_1 \cup E_2)$ be a non-empty system network, and $\alpha : \{\boldsymbol{A},\boldsymbol{B}\} \to G(\boldsymbol{A},\boldsymbol{B})$ be a bijection. For each non-zero $a_{ij} \in \boldsymbol{A}$, $b_{pq} \in \boldsymbol{B}$, there are $\alpha : a_{ij} \to \overrightarrow{\langle v_j, v_i\rangle}$ and $\alpha : b_{pq} \to \overrightarrow{\langle u_q, v_p\rangle}$, where $\overrightarrow{\langle v_j, v_i\rangle} \in E_1$, $\overrightarrow{\langle u_q, v_p\rangle} \in E_2$, $\{v_i, v_j, v_p\} \subseteq V_1$ and $u_q \in V_2$.*

Definition 3 (**Stem & Bud**[4]). *By definition 2, a stem in $G(\boldsymbol{A},\boldsymbol{B})$ is a directed path only starting from a node of V_2. A bud is a directed cycle plus an arc, whose head is shared with this cycle, and this arc is called the distinguished edge.*

Definition 4 (**Dilation**[4]). *In $G(\boldsymbol{A},\boldsymbol{B}) = (V_1 \cup V_2, E_1 \cup E_2)$, $S \subseteq V_1$ is a set of nodes, $T(S) \subseteq V_1 \cup V_2$ is set of vertices as tails of the arcs whose heads are in S. When $G(\boldsymbol{A},\boldsymbol{B})$ contains a dilation, if and only if $|S| > |T(S)|$.*

Definition 5 (**Inaccessibility** [4]). *In $G(\boldsymbol{A},\boldsymbol{B}) = (V_1 \cup V_2, E_1 \cup E_2)$, a node of V_1 that can not be visited through directed paths starting from any node of V_2 is inaccessible.*

Definition 6 (**Cactus**[4]). *Let $\{B_1, B_2, \ldots, B_l\}$ be a set of buds, and let S_1 be a stem, $S_1 \cup B_1 \cup B_2, \ldots, \cup B_l$ is a cactus if and only if the tail of the distinguished edge of $B_i (1 \le i \le l)$ is not the top node of S_1 but the only common node of $S_1 \cup B_1 \cup B_2, \ldots \cup B_{i-1}$. Besides, a stem is also a cactus.*

Theorem 1 (**Structural Controllability Theorem** [4], [5]). *Following statements are equivalent:*

1) *System described by equation 1 is structurally controllable.*
2) *$G(\boldsymbol{A},\boldsymbol{B})$ of definition 2 contains neither inaccessible nodes nor a dilation.*
3) *$G(\boldsymbol{A},\boldsymbol{B})$ is spanned by a set of disjoint cacti.*

In particular, according to [4], [5], when statement one is satisfied, statement two can be implied by statement one, and it can finally imply statement three. Nevertheless, statement three can always imply both statement one and two. Thus, we conclude corollary 1:

Corollary 1. *By theorem 1, a system described by equation 1 is structurally controllable if and only if its system network excludes inaccessible nodes and a dilation, and spanned by a set of disjoint cacti.*

A. Problem Formulation

Given a LTI system, which was structurally controllable, but it is now structurally uncontrollable due to severe attack or failure, and is described by a state equation 2 below:

$$\dot{x}(t) = \mathbf{A}^{'}x(t) + \mathbf{B}^{'}u(t) \tag{2}$$

where $\mathbf{A}^{'} \in \mathbb{R}^{N\times N}$ and $\mathbf{B}^{'} \in \mathbb{R}^{N\times M}$. Particularly, matrix $\mathbf{B}^{'}$ is fixed. Then, we recover structural controllability of this system, and the problem is identify a matrix with the minimum number of non-zero entries, noted by $\mathbf{A}^{''} \in \mathbb{R}^{N\times N}$, so that the resulting system described by equation 3 below:

$$\dot{x}(t) = (\mathbf{A}^{'} + \mathbf{A}^{''})x(t) + \mathbf{B}^{'}u(t) \tag{3}$$

is structurally controllable.

By corollary 1 and definition 2, given a system network mapped by $\{\mathbf{A}^{'}, \mathbf{B}^{'}\}$, to recover structural controllability with given inputs, our solution is to only add a minimum set of edges into this network, to construct a set of disjoint cacti in final system network, where any added edge is not incident to nodes mapped by $\mathbf{B}^{'}$.

III. Literature Review

Structural controllability recovery attracts increasing attention. Based on the grah-theoretical way to derive structrual controllability, it can be recovered by identifying a maximum matching [13] or a power dominating set [14] of the given network mapped by state matrix of a structurally uncontrollable system. For instance, authors of [3], [15] recover the structural control into LTI system after single vertex removal or addition by efficiently identifying a maximum matching of a network mapped by residual state matrix via the bijection of definition 2, rather than recomputing a maximum matching. On the other hand, according to original research of [16] [17], Alwasel *et al*. in [2] [18] [19] recovered structural controllability of the *Erdős-Rényi* random digraph in LTI model after removing multiple vertices by maintaining an approximated power dominating set [14]. Similarly, Alcaraz *et al*. [20] relies on the power dominating set to recover structural controllability of general power-law and scale-free digraphs against both edge and vertex removals. However, these related works either recover structural controllability against very limited failure or attack, or neglect constrains on inputs, and all of them just identify a set of vertices that should be directly injected by inputs.

By contrast, it is more realistic and sufficient to concern constrains on inputs during the process of recovering structural controllability, which might include the number of inputs, or the adjacency between inputs and system components. Literally, via a given system network, such kind of structural controllability recovery is related to the problem of deriving structural controllability with given inputs. Generally, it requires extra modification, such as adding edges into the original system network. In [6], Chen $et\ al.$ propose to get structural controllability by the minimal edge addition. Their edge-addition scenario is mainly based on the work of [21] and [22], which obtains the structural controllability by identifying a set of dedicated inputs and according to existing strongly connected components of the network mapped by the state matrix. Nevertheless, edge-addition scenario of [6] considers some unnecessary data structures to obtain the resulting digraph spanned by disjoint cacti. As a result, the minimal edge addition of [6] is implemented with low efficiency, whose time complexity is proportional to the cubic number of vertices of the initial system network.

IV. Disjoint Cacti Construction

In this section, by corollary 1, we construct a graph spanned by disjoint cacti via adding a minimum set of edges into our input digraph of definition 7, which is a system network mapped by a structurally uncontrollable system.

Definition 7 (**Input Digraph**). *Let $D = (V \cup U, E)$ be a finite digraph excluding self loops, isolated vertices and parallel arcs. Also, let $V = \{v_i | 1 \le i \le N\}$ and $U = \{u_r | 1 \le r \le M\} (M \le N)$ be two independent vertex sets, where each node of U has no in degree and is a tail of the arc whose head is only a node of V. Besides, E is a set of edges among vertices of $V \cup U$.*

We add edges into (V, E) to construct a set of disjoint cacti in the final digraph, and all nodes of U should be the starting vertices of all stems of the constructed cacti, where we mainly rely on the maximum matching of a digraph, which is defined:

Definition 8 (**Maximum Matching of Digraphs** [23], [24]). *In digraphs, a matching is a set of arcs without common tails and heads, and a maximum matching is a matching with the highest cardinality. For any given vertex, it is an unmatched node with respect to a maximum matching, if and only if it is not the head of any arc of this maximum matching. Otherwise, it is matched.*

In general, a maximum matching can be effectively identified in polynomial time by the Hopcroft-Karp algorithm [13], and there might be multiple maximum matchings in a same digraph. Given a digraph with n nodes and m arcs, finding a maximum matching of it costs $O(\sqrt{n} \cdot m)$ steps at most [13]. In the remaining parts of this section, by theorem 1, we show a scenario of constructing a graph spanned by disjoint cacti in section IV-A and IV-B. Based on this scenario, in IV-C, we confirm the minimum number of added edges into D, and the related algorithm is shown in section IV-D eventually.

A. The first edge-addition step

From theorem 1 and corollary 1, since the finally resulting digraph derived by adding edges into $D = (V \cup U, E)$ of definition 7 should have no dilations. We thus use the dilation of D as the initial clue to guide our edge addition in the beginning, and we also conclude lemma 1 to justify the first edge-addition step, which relies on an arbitrarily identified maximum matching of D to just sufficiently detect the nonexistence of the dilation of D.

Lemma 1. *In $D = (V \cup U, E)$, let M_D be an arbitrarily identified maximum matching of D. Then, if each vertex of V is a matched node related to M_D, D excludes the dilation.*

Proof: If each vertex of V is matched nodes with respect to M_D. Then, in M_D, by definition 8 each node of V must be a vertex as a head of an edge of M_D, whose tail is either a node of V or U and such node can not be an terminal of a path of M_D. Also, in $M_D \cup U$, because the number of heads and that of tails are same, $U \cup M_D$ spans D and excludes the dilation. Besides, since any single edge of $E \setminus M_D$ added into $U \cup M_D$ can not increase the number of vertices as heads of arcs of E, while it can only increase the number of nodes as tails of arcs of E. Otherwise, maximality of M_D is contradicted. Therefore, there can not be the dilation after adding all edges of $E \setminus M_D$ into $U \cup M_D$, and D thus excludes the dilation. □

According to lemma 1, our first edge-addition step is clearly to add edges among different vertices of V to eliminate all unmatched nodes of V with respect to M_D, and the addition can not produce any self loops. Let E_{n_1} be a set of edges added into D in the first step, and $|E_{n_1}| = n_1$. In detail, for each edge of E_{n_1}, noted by e, its head is an unmatched node of V related to M_D, and its tail could be either the ending node of a path of $M_D \cup \{E_{n_1} \setminus e\}$, or an currently existing unmatched node of V related to M_D. After this, there is no dilation in the resulting digraph, because $M_D \cup E_{n_1}$ is the maximum matching of $(V \cup U, E \cup E_{n_1})$. Otherwise, maximality of M_D is contradicted. Also, all vertices of existing paths of $\{M_D \cup E_{n_1}\}$ could be now accessible from nodes of U.

Since the number of unmatched nodes with respect to any maximum matching of a digraph is a constant value, n_1 is thus constant according to D, and our first edge-addition step requires a constant number of added edges.

B. The second edge-addition step

After implementing the first edge-addition step for given $D = (V \cup U, E)$ of definition 7, we now detect and eliminate vertices of V that are inaccessible from nodes of U in the resulting digraph $(V \cup U, E \cup E_{n_1})$. To do this, we also conclude lemma 2, which uses strongly connected components of D to identify such inaccessible nodes and guide following edge addition.

Lemma 2. *Given $D = (V \cup U, E)$ and E_{n_1}, let M_D be a maximum matching of it. Then, digraph $(V \cup U, E \cup E_{n_1})$ contains vertices of V that are inaccessible from nodes of U, if and only if there are strongly connected components only including one or more disjoint cycles of M_D, which exclude nodes as heads of arcs whose tails out of them.*

Proof: A strongly connected component of a digraph is a subgraph whose any pair of vertices are connected through at least one existing directed path.

Necessity: Within $(V \cup U, E \cup E_{n_1})$, if there are strongly connected components that only involve one or more disjoint cycles of M_D, and exclude nodes pointed by vertices out of them. Obviously, vertices of these strongly connected components can not be approached by nodes out of these components through existing paths, so that they are inaccessible vertices of V from nodes of U in $(V \cup U, E \cup E_{n_1})$.

Sufficiency: If $(V \cup U, E \cup E_{n_1})$ contains inaccessible vertices of V from U. After the first edge-addition step of section IV-A according to M_D, inaccessible nodes of V from nodes of U in $(V \cup U, E \cup E_{n_1})$ are only contained by inaccessible cycles of M_D by nodes of U. This is because $(V \cup U, E \cup E_{n_1})$ has no unmatched nodes from V, where each node of V is either in a path starting from a node of U or in the cycle of M_D, and any cycle of M_D excludes unmatched nodes related to M_D. Let C_i be an arbitrary inaccessible cycle of M_D from nodes of U, then, C_i could have no vertices pointed by any other vertex out of it, so that nodes of U can not visit it through existing paths on the one hand. On the other hand, each cycle that has nodes visiting nodes of C_i via existing path in $(V \cup U, E \cup E_{n_1})$ must be inaccessible from nodes of U in $(V \cup U, E \cup E_{n_1})$. Further, because D and E_{n_1} are finite, in $(V \cup U, E \cup E_{n_1})$, if there are strongly connected components, there must be at least one strongly connected component, which either contains a single cycle of M_D, or only contains disjoint cycles of M_D, whose vertices are inaccessible from vertices vertices out of them. Therefore, when $(V \cup U, E \cup E_{n_1})$ contains inaccessible nodes of V from U, there are strongly connected components only involving one or more disjoint cycles of M_D and excluding nodes pointed by vertices out of them. □

According to lemma 2, within $(V \cup U, E \cup E_{n_1})$ after the first edge-addition step, the second edge-addition step is to remove inaccessible nodes of V from U. Let E_{n_2} be the set of added edges, and $|E_{n_2}| = n_2$, which is the number of strongly connected components only containing one or more disjoint cycles of M_D, and excluding nodes pointed by vertices out of them. The second edge-addition step is as follows: for each added edges of E_{n_2}, its head should be involved into such a strongly connected component, and its tail can be a vertex of a path of $M_D \cup E_{n_1}$.

Because $(V \cup U, E \cup E_{n_1})$ is a finite digraph, n_2 is bounded by the finite number of strongly connected components of $(V \cup U, E \cup E_{n_1})$. Then, corollary 2 clarifies the correctness of our two edge-addition steps:

Corollary 2 (**Edge-addition scenario**). *Given $D = (V \cup U, E)$ of definition 7, let M_D be a maximum matching of it, E_{n_1} and E_{n_2} be two added edge sets of the first and the second edge-addition steps by M_D. Then, after the first and second edge addition, digraph $(V \cup U, \{E \cup E_{n_1}\} \cup E_{n_2})$ could be spanned by a set of disjoint cacti, whose all stems only start from nodes of U.*

Proof: By lemma 1, after the first step, the obtained digraph $(V \cup U, E \cup E_{n_1})$ has no dilations, and can be spanned by $M_D \cup E_{n_1}$, which can contain disjoint paths only starting from nodes of U and disjoint cycles of M_D. Also, adding E_{n_2} into $(V \cup U, E \cup E_{n_1})$ is to ensure that nodes of all cycles are accessible from U by lemma 2. Besides, because any cycles of M_D is not affected by adding E_{n_1} into D, $E_{n_1} \cap E_{n_2} = \emptyset$. Thus, by definition 6, $(V \cup U, E \cup E_{n_1} \cup E_{n_2})$ is spanned by a set of disjoint cacti, whose disjoint stems can start from nodes of U. And the number of added edges is $n_1 + n_2$. □

Next, we discuss when the number of added edges by those two steps can be reduced.

C. The Minimum Number of Added Edges

In this section, we confirm the minimum number of added edges by scenario summarized by corollary 2. Because the number of unmatched nodes with respect to any maximum matching of $D = (V \cup U, E)$ of definition 7 is constant, the minimum number of added edges only depends on the minimum number of added edges required by the second edge-addition step of section IV-B. Firstly, we conclude theorem 2 to indicate how to reduce the number of added edges by one, where few essential items are defined:

Definition 9 (**Scc**). *Given $D = (V \cup U, E)$, let M_D be a maximum matching of D. Then, with M_D, let S_{cc} be a set of all strongly connected components that only include one or more disjoint cycles of M_D, and exclude nodes as heads of arcs whose tails are out of them in D.*

Theorem 2. *Given $D = (V \cup U, E)$, M_D, and Scc, let $\overrightarrow{\langle v_i, v_j \rangle}$ be an arc of a cycle involved into an element of Scc and $v_k \in V$ be an unmatched node related to M_D. Then, by edge-addition scenario of corollary 2, the total number of added edges into D according to M_D is reduced by one, if and only if arc $\overrightarrow{\langle v_i, v_k \rangle}$ exists, and v_j is an unmatched node related to a maximum matching different from M_D.*

Proof: Let $M_D^{'}$ be a maximum matching of D, and $M_D^{'} \neq M_D$. Since the number of edges added in the first step is constant, we thus prove that the number of added edges in the second step according to $M_D^{'}$ is less than that according to M_D by one by lemma 2, if and only if arc $\overrightarrow{\langle v_i, v_k \rangle}$ exists, and v_j is an unmatched node related to $M_D^{'}$.

Necessity: If arc $\overrightarrow{\langle v_i, v_k \rangle} \in E \setminus M_D$ exists, and v_j is an unmatched node related to $M_D^{'}$. Then, in $M_D^{'}$, a path can exist, which starts from v_j, contains all vertices of a cycle of M_D that involves $\overrightarrow{\langle v_i, v_j \rangle}$, and a path of M_D starting from v_k together. After the first edge-addition step by $M_D^{'}$, we can observe that all nodes of an element of Scc involving $\overrightarrow{\langle v_i, v_j \rangle}$ is accessible from nodes of U, and there is no need for extra edges to make this element of Scc accessible again by lemma 2. Nevertheless, by M_D, after the dilation removal, such same element of Scc is still inaccessible from nodes of U, which thus requires an edge to make nodes of it accessible by lemma 2. Hence, the total number of added edges into D according to $M_D^{'}$ is less than that according to M_D by one.

Sufficiency: After the first edge-addition step, based on lemma 2, if the number of added edges to remove nodes of V that are inaccessible from U according to $M_D^{'}$ is less than that according to M_D by one. Let $Scc^{'}$ be a set of all strongly connected components that only involve one or more disjoint cycles of $M_D^{'}$, and exclude nodes pointed by vertices out of them after removing dilation of D. Then, because adding edges according to any maximum matching of D in the first edge-addition step does not influence any cycle of this maximum matching. The number of elements of Scc is thus more than that of $Scc^{'}$ by one, or $|Scc^{'}| = |Scc| - 1$. Also, since any two maximum matchings of a same digraph can be transformed into each other by exchanging vertices or edges, it is possible that one element that is originally contained by Scc should be out of Scc and $Scc^{'}$ later, which requires that all nodes of a cycle of this element of Scc should be contained into a path of $M_D^{'}$. By definition 9, since each element of Scc is spanned by disjoint cycles and has no incoming arcs from other vertices out of them in D. For the element of Scc only involving a single cycle of M_D, we should make its vertices be contained by a path and without changing the number of unmatched nodes of D. On the other hand, for the element of Scc only involving multiple disjoint cycles of M_D, we can make any involved single cycle's vertices be contained by a path and without changing the number of unmatched nodes of D, in which other cycles of this element would be excluded by any set like Scc, because a vertex of them has incoming edges now. To do this, given M_D, nodes of a path of $M_D^{'}$ are all vertices of a cycle and a path of M_D, in which the starting vertex of this path is noted by v_k, and an edge of this cycle is noted by $\overrightarrow{\langle v_i, v_j \rangle}$. Then, there must be $\overrightarrow{\langle v_i, v_k \rangle} \in M_D^{'}$, $\overrightarrow{\langle v_k, v_j \rangle} \notin M_D$, and v_j is the starting vertex of this path of $M_D^{'}$. □

According to theorem 2, corollary 3 reduces the maximum number of added edges involved into the edge-addition scenario of corollary 2:

Corollary 3. *Given $D = (V \cup U, E)$, M_D, and Scc, let $S_{sub} \subseteq Scc$, and each element of S_{sub} has a vertex as the tail of an arc whose head is an unmatched node of V related to M_D. Then, by the scenario of corollary 2, the total number of added edges into D according to M_D is reduced by the most by theorem 2, if cardinality of S_{sub} is maximum.*

Proof: Given any element of S_{sub}, based on theorem 2, the number of added edges into D according to M_D can be reduced by one. Also, because elements of S_{cc} are not connected, once a vertex of any element of S_{sub} can be an unmatched node with respect to a maximum matching different from M_D, there can be a common maximum matching different M_D for all such vertex. Besides, since D is a finite graph, $S_{sub} \subseteq S_{cc}$ is a finite strongly connected component. Therefore, in aggregation, when cardinality of S_{sub} is the maximum, the number of added edges by scenario of corollary 2 is reduced by the most. □

D. Execution

According to corollary 2, theorem 2 and corollary 3, we execute the entire process of constructing a graph spanned by a set of disjoint cacti via adding a minimum set of edges into $D = (V \cup U, E)$ of definition 7, which is systematically illustrated by algorithm 1. Here, M_D and S_{cc} of definition 9 are used, and let $v_h \in V$ be any vertex not incident to edges of M_D. Besides, let s_i be an element of S_{cc}, p_i be a path of M_D, and G be an initially empty set. Also, let $v_k \in V$ be an starting node of p_i, c_i be a cycle of s_i, and $\overrightarrow{\langle v_i, v_j \rangle} \in E$ be an arc of c_i.

Algorithm 1: Construct a digraph spanned by cacti

Input: $D = (V \cup U, E)$ of definition 7, G
Output: A digraph spanned by disjoint cacti
1 Starting from each node of U, find M_D by running the algorithm of [13] ;
2 Identify each c_i, p_i of M_D by DFS algorithm of [25], and each v_h;
3 Add each v_h into G;
4 Identify S_{cc} from each c_i in D; $G^{'} = G \cup S_{cc}$;
5 Identify each $s_i \in S_{cc}$, for $c_i \in s_i$, where $\overrightarrow{\langle v_i, v_j \rangle} \in c_i$, and $\overrightarrow{\langle v_i, v_k \rangle} \in E \setminus M_D$;
6 **for** *each identified s_i in line 5* **do**
7 $\quad G^{'} = G \setminus s_i$; $S_{cc}^{'} = S_{cc} \setminus s_i$;
8 $\quad$ Construct a path, which is $\{c_i \setminus \overrightarrow{\langle v_i, v_j \rangle}\} \cup \{p_i \cup \overrightarrow{\langle v_i, v_k \rangle}\}$;
9 $\quad$ Add this path into G;
10 $\quad M_D^{'} = M_D \setminus \{p_i, c_i\}$;
11 $G^{'} = G \cup M_D$;
12 Add arcs into G from each existing zero-outdegree node of V to each zero-indegree node of V until there is no zero-indegree node of V ;
13 Add arcs into G from any node of V and out of S_{cc} to each element of S_{cc} until each element of S_{cc} has an incoming edge;
14 **return** $D \cup G$;

Proof: Initially, this procedure identifies a maximum matching M_D to detect the dilation and inaccessible nodes of D in line 1. Particularly, M_D is identified from edges incident to vertices of U in order to sufficiently use vertices of U. Then, based on theorem 2 and corollary 3, line 2-4 of this algorithm identify S_{cc} according to cycles of M_D, and S_{cc} is added into G. By theorem 2 and corollary 3, line 5 and 6 modifies some strongly connected components of S_{cc} collected in line 5 to reduce the number of added edges according to M_D by the most. During the modification, paths obtained by line 8 are added into G, while those related elements, and involved paths of M_D that start from nodes of V are removed from G and S_{cc}. After running the line 11, G contains disjoint paths of M_D and $E \setminus M_D$, single vertices not incident to M_D, and remaining S_{cc} whose all elements have no incoming edges from nodes out of them. Clearly, in G, all vertices of V without indegree are unmatched nodes related to a common maximum matching different from M_D. Later, the first edge-addition step is executed by line 12, which adds edges to remove all unmatched nodes of V and related to a common maximum matching that is different from M_D, so that the resulting digraph has no dilation and all paths starting from nodes of U. In the following, line 13 removes inaccessible nodes of V from U by lemma 2. Specifically, those inaccessible nodes are removed through vertices of remaining elements of S_{cc} in G. Additionally, since G contains $V \cup U$, and also G contains a set of disjoint cacti by corollary 2, whose stems only start from nodes of U. $D \cup G$ is therefore spanned by disjoint cacti whose stems starting from nodes of U.

For the worst-case execution time of this algorithm, it is the sum of running time of each line. Identifying a maximum matching of D costs $O(\sqrt{|V \cup U|} \cdot |E|$ by algorithm [13], and identifying cycles and paths of M_D cost $O(|V \cup U| + |E|)$ by DFS algorithm. Then, for running line 4 and 5, it requires to visit edges whose tails are involved into identified cycles, which thus costs $O(|V||E|)$. Next, running the **for** loop of line 6, since combining nodes of each s_i and a related p_i into a path of $E \setminus M_D$ costs $O(1)$, the worst-case execution time of this procedure is $O(|E|)$. Later, adding edges of the first step in line 12 requires to identify zero-indegree and zero-outdegree nodes, which can be done in $O(|V|)$ steps at most. As for the second edge-addition step of line 13, since S_{cc} is already known, the edge addition only depends on the existence of elements of S_{cc}, which thus costs $O(|E|)$ steps at most. Above all, time complexity of this algorithm is $O(\sqrt{|V \cup U|} \cdot |E|$ for $D = (V \cup U, E)$. □

Furthermore, by when the given system network D is a sparse ER random digraph [4], the performance of our edge addition scenario might be performed more efficiently with the average time complexity $O\Big(|E| \cdot log(|V \cup U|)\Big)$ [26].

V. Conclusion

Given a structurally uncontrollable system, recovery of its structural controllability can be done by various methods and requirements. In this paper, our recovery is constrained by inputs and time complexity, where the input matrix is fixed. For our solution, we add a minimum set of edges into the given system network to obtain a digraph spanned by a set of disjoint cacti, so that the system represented by this digraph is structurally controllable. For the time complexity of executing our entire operations in the worst case, it is equivalent to that of identifying a maximum matching of the system network. In our future work, we would recover structural controllability with given inputs by only rewiring edges of the system network, so that the cost of recovery can be further decreased.

References

[1] A. Chapman and M. Mesbahi, "Security and infiltration of networks: A structural controllability and observability perspective," in *Control of Cyber-Physical Systems*. Springer, 2013, pp. 143–160.

[2] B. Alwasel and S. D. Wolthusen, "Recovering structural controllability on erdős-rényi graphs via partial control structure re-use," in *International Conference on Critical Information Infrastructures Security*. Springer, 2014, pp. 293–307.

[3] S. Zhang and S. D. Wolthusen, "Efficient control recovery for resilient control systems," in *Networking, Sensing and Control (ICNSC), 2018 IEEE 15th International Conference on*. IEEE, 2018, pp. 1–6.

[4] C. T. Lin, "Structural controllability," *Automatic Control, IEEE Transactions on*, vol. 19, no. 3, pp. 201–208, 1974.

[5] H. Mayeda, "On structural controllability theorem," *IEEE Transactions on Automatic Control*, vol. 26, no. 3, pp. 795–798, 1981.

[6] X. Chen, S. Pequito, G. J. Pappas, and V. M. Preciado, "Minimal edge addition for network controllability," *IEEE Transactions on Control of Network Systems*, 2018.

[7] R. E. Kalman, "Mathematical description of linear dynamical systems," *Journal of the Society for Industrial and Applied Mathematics, Series A: Control*, vol. 1, no. 2, pp. 152–192, 1963.

[8] R. Kalman, "On the general theory of control systems," *Automatic Control, IRE Transactions on*, vol. 4, no. 3, pp. 110–110, 1959.

[9] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, "Controllability of complex networks," *Nature*, vol. 473, no. 7346, pp. 167–173, 2011.

[10] 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.

[11] H. Mayeda and T. Yamada, "Strong structural controllability," *SIAM Journal on Control and Optimization*, vol. 17, no. 1, pp. 123–138, 1979.

[12] R. Shields and J. Pearson, "Structural controllability of multiinput linear systems," *IEEE Transactions on Automatic control*, vol. 21, no. 2, pp. 203–212, 1976.

[13] J. E. Hopcroft and R. M. Karp, "An nˆ5/2 algorithm for maximum matchings in bipartite graphs," *SIAM Journal on computing*, vol. 2, no. 4, pp. 225–231, 1973.

[14] T. W. Haynes, S. M. Hedetniemi, S. T. Hedetniemi, and M. A. Henning, "Domination in graphs applied to electric power networks," *SIAM Journal on Discrete Mathematics*, vol. 15, no. 4, pp. 519–529, 2002.

[15] S. Zhang and S. D. Wolthusen, "Iterative recovery of controllability via maximum matching," in *Automation Science and Engineering (CASE), 2017 13th IEEE Conference on*. IEEE, 2017, pp. 328–333.

[16] A. Aazami and K. Stilp, "Approximation algorithms and hardness for domination with propagation," *SIAM Journal on Discrete Mathematics*, vol. 23, no. 3, pp. 1382–1399, 2009.

[17] J. Guo, R. Niedermeier, and D. Raible, "Improved algorithms and complexity results for power domination in graphs," *Algorithmica*, vol. 52, no. 2, pp. 177–202, 2008.

[18] B. Alwasel and S. D. Wolthusen, "Recovering structural controllability on erdős-rényi graphs in the presence of compromised nodes," in *International Conference on Critical Information Infrastructures Security*. Springer, 2015, pp. 105–119.

[19] ——, "Structural controllability analysis via embedding power dominating set approximation in erdhos-rènyi graphs," in *Advanced Information Networking and Applications Workshops (WAINA), 2015 IEEE 29th International Conference on*. IEEE, 2015, pp. 418–423.

[20] C. Alcaraz and S. Wolthusen, "Recovery of structural controllability for control systems," in *International Conference on Critical Infrastructure Protection*. Springer, 2014, pp. 47–63.

[21] S. Pequito, S. Kar, and A. P. Aguiar, "A structured systems approach for optimal actuator-sensor placement in linear time-invariant systems," in *American Control Conference (ACC), 2013*. IEEE, 2013, pp. 6108–6113.

[22] S. D. Pequito, S. Kar, A. P. Aguiar *et al.*, "A framework for structural input/output and control configuration selection in large-scale systems." *IEEE Trans. Automat. Contr.*, vol. 61, no. 2, pp. 303–318, 2016.

[23] G. Chartrand, L. Lesniak, and P. Zhang, *Graphs & digraphs*. CRC Press, 2010.

[24] J. Bang-Jensen and G. Z. Gutin, *Digraphs: theory, algorithms and applications*. Springer Science & Business Media, 2008.

[25] R. Tarjan, "Depth-first search and linear graph algorithms," *SIAM journal on computing*, vol. 1, no. 2, pp. 146–160, 1972.

[26] H. Bast, K. Mehlhorn, G. Schafer, and H. Tamaki, "Matching algorithms are fast in sparse random graphs," *Theory of Computing Systems*, vol. 39, no. 1, pp. 3–14, 2006.