Zero forcing number, constraint matchings and strong structural controllability

Maguy Trefois* Jean-Charles Delvenne* †

January 29, 2014

Abstract

The zero forcing number is a graph invariant introduced in order to study the minimum rank of the graph. In the first part of this paper, we first highlight that the computation of the zero forcing number of any directed graph (allowing loops) is NP-hard. Furthermore, we identify a class of directed trees for which the zero forcing number is computable in linear time. The second part of the paper is an application of the notion of zero forcing set in the study of the strong structural controllability of networked systems. This kind of controllability takes into account only the structure of the interconnection graph, but not the interaction strengths along the edges. We present the first efficient algorithm providing a minimum-size input set for the strong structural controllability of a self-damped or an undamped system with a tree structure.

1 Introduction

The minimum rank problem has been motivated by the Inverse Eigenvalue Problem of a Graph (IEPG) [1, 2], intensively studied during the last fifteen years. In the IEPG, a simple undirected graph $G$ is considered. Such a graph defines a set $Q_{su}(G)$ of real symmetric matrices whose zero-nonzero pattern of the off-diagonal entries is described by the graph: the $(i, j)$-entry (for $i \neq j$) is nonzero if and only if $\{i, j\}$ is an edge in $G$. The zero-nonzero pattern of the diagonal entries is free.

*ICTEAM Institute - Department of Applied Mathematics, Université catholique de Louvain, Avenue Georges Lemaître 4 B-1348 Louvain-La-Neuve, Belgium. maguy.trefois@uclouvain.be, jean-charles.delvenne@uclouvain.be.

†Center for Operations Research and Econometrics (CORE), Université catholique de Louvain, Louvain-La-Neuve, Belgium.
Given a sequence \([\mu_1, \ldots, \mu_n]\) of non increasing real numbers, we ask whether there exists a matrix in the set \(Q_{su}(G)\) whose spectrum is \([\mu_1, \ldots, \mu_n]\). A first step to study this challenging problem is to compute the maximum possible multiplicity of an eigenvalue \(\mu\) for a matrix in \(Q_{su}(G)\). This maximum is obtained thanks to the minimum rank of the graph \(G\). Indeed, for any real number \(\mu\), the maximum possible multiplicity of \(\mu\) as an eigenvalue of a matrix in \(Q_{su}(G)\) is:

\[
|G| - mr(G),
\]

where \(|G|\) denotes the number of vertices in \(G\) (the vertex set of a graph is assumed to be finite) and \(mr(G)\) refers to the minimum rank of the graph, that is:

\[
mr(G) = \min\{\text{rank}(A) : A \in Q_{su}(G)\}.
\]

Since the zero-nonzero pattern of the diagonal entries for a matrix in \(Q_{su}(G)\) is free, this maximum multiplicity does not depend on \(\mu\).

The minimum rank of a graph is also useful in other problems like the singular graphs or the biclique partition of the edges of a graph (see \[2\] for a brief description and interesting references).

Originally given for simple undirected graphs, the definition of the minimum rank of a graph has then been extended to the case of the simple directed graphs and the loop (undirected and directed) graphs, which are graphs allowing loops.

In this paper, the term ‘simple graph’ refers to a graph that prohibits loops whereas the term loop graph’ refers to a graph that allows loops on its vertices.

In order to study the minimum rank of a graph, many graph invariants have been introduced \([1, 2, 3, 4, 5]\). One of them is the zero forcing number, presented below.

The zero forcing number of a graph was originally defined in \([4]\) for simple graphs and extended to loop graphs in \([3]\). The definitions of zero forcing number and zero forcing set depend on a color change rule on the graph. This rule is slightly different in a loop graph or in a simple graph.

Since the zero forcing number of a loop/simple undirected graph is equal to the zero forcing number of its associated symmetric directed graph \([3]\), we will only focus here without loss of generality on the case of the loop/simple directed graphs.

The color change rule in a loop directed graph \([3, 5]\) is the following: suppose that any vertex of \(G\) is either black, or white. If exactly one out-neighbor \(j\) of vertex \(i\) is white (possibly \(j = i\)), then change the color of vertex \(j\) to black.

In a simple directed graph \(G\), the color change rule is defined as follows: suppose any vertex of \(G\) to be black or white. If vertex \(i\) is black and vertex \(j\) is the only white out-neighbor of \(i\), then change the color of \(j\) to black.
Hence, in a simple graph vertex $i$ must be black to be able to change the color of one of its out-neighbors, which is not the case in a loop directed graph.

These rules are repeatedly applied to each vertex of $G$ until no more color change is possible. The **zero forcing number** $Z(G)$ of the graph $G$ is defined as the minimum number of vertices which have to be initially black so that after applying repeatedly the color change rule to $G$ all the vertices of $G$ are black. A vertex subset $Z$ of $G$ with the property that if only the vertices of $Z$ are initially black in $G$, then the whole graph is black after applying repeatedly the CCR is called a **zero forcing set** of $G$. A zero forcing set of size $Z(G)$ is called a **minimum zero forcing set**.

This graph invariant allows to compute a lower bound for the minimum rank of the graph. The definition of the minimum rank of the graph relies on the matrix family defined by the graph. This family of matrices depends on the type of the graph:

- if $G = (V, E)$ is a simple undirected graph,
  $$Q_{su}(G) = \{ A \in \mathbb{R}^{[G] \times [G]} : A^T = A, \text{ for any } i \neq j, a_{ij} \neq 0 \Leftrightarrow \{i, j\} \in E \},$$
- if $G = (V, E)$ is a simple directed graph,
  $$Q_{sd}(G) = \{ A \in \mathbb{R}^{[G] \times [G]} : \text{ for any } i \neq j, a_{ij} \neq 0 \Leftrightarrow (i, j) \in E \},$$
- if $G = (V, E)$ is a loop undirected graph,
  $$Q_{lu}(G) = \{ A \in \mathbb{R}^{[G] \times [G]} : A^T = A, \text{ for any } i, j, a_{ij} \neq 0 \Leftrightarrow \{i, j\} \in E \},$$
- if $G = (V, E)$ is a loop directed graph,
  $$Q_{ld}(G) = \{ A \in \mathbb{R}^{[G] \times [G]} : \text{ for any } i, j, a_{ij} \neq 0 \Leftrightarrow (i, j) \in E \}.$$

The minimum rank of a graph $G$ of type $i$ with $i = su$, $sd$, $lu$ or $ld$ is then the minimum possible rank for a matrix in $Q_i(G)$, that is:

$$mr(G) = \min \{ \text{rank}(A) : A \in Q_i(G) \}.$$  

It was proved in [3] that for any graph $G$,

$$|G| - Z(G) \leq mr(G).$$

Moreover, if $G$ is a tree of any type (simple or allowing loops, undirected or directed (see Section 3)), then equality holds [3]:

$$|G| - Z(G) = mr(G). \quad (1)$$
This paper is based on the equivalence between the zero forcing sets in a loop directed graph $G$ and the constraint matchings in a bipartite graph defined from $G$.

The notion of constraint matching in a bipartite graph was originally defined in the paper of Hershkowitz and Schneider [6]. A $t$-matching is a set of $t$ edges such that no two edges share a vertex. Given a matching, a vertex of the graph that belongs to an edge of the matching is called a matched vertex, otherwise it is an unmatched vertex. A $t$-matching is called constraint if there is no other $t$-matching with the same matched vertices.

Given a loop directed graph $G$, a bipartite graph $B_G = (V, V', E)$ was defined in [6] as follows: the sets $V$ and $V'$ are two copies of the vertex set of $G$. To avoid ambiguity, the vertices in $V$ are denoted $i_1, ..., i_n$ and the vertices in $V'$ are denoted $j'_1, ..., j'_n$. Given any vertex $i \in V$ and any vertex $j' \in V'$, $\{i, j'\}$ is an edge in $B_G$ if and only if there is an edge from vertex $j$ to vertex $i$ in $G$. Below $B_G$ is then called the bipartite graph associated with $G$.

To our best knowledge, the equivalence between a constraint matching in a bipartite graph [6, 7, 8, 9] and a zero forcing set in a loop directed graph [2, 3, 5] has gone unnoticed. In this paper, we emphasize this equivalence and apply it in the study of the strong structural controllability of networked systems. The problem of determining a control strategy for a network of interconnected systems without exact knowledge of the interaction strengths along the edges has seen a surge of activity in the last decade. In particular, regarding weak and strong structural controllability introduced in the 70’s [10, 11]. Structural controllability takes into account only the structure of the interconnection graph, but not the interaction strengths on the edges. A system with a given interconnection graph is weakly structurally controllable from an input set $S$ if we can choose interaction strengths making the system controllable from $S$. The minimum-size input sets from which the system is weakly structurally controllable are provided by the maximum matchings in the interconnection graph [12]. Instead, a system with a given interconnection graph is strongly structurally controllable from an input set $S$ if whatever the interaction strengths, the system is controllable from $S$. The minimum-size input sets from which the system is strongly structurally controllable are provided by some maximum constraint matchings in the bipartite graph associated with the interconnection graph [7].

In this paper, we shed a new light on the strong structural controllability from the notion of zero forcing set. Firstly, we show that testing if a system is strongly structurally controllable from an input set $S$ is equivalent to checking if $S$ is a zero forcing set in the interconnection graph. Secondly, in the case of systems that are self-damped (each vertex’s state is influenced by itself) or that are undamped (each vertex’s state is influenced by the state of some other vertices but not itself),
we show that minimum-size input sets for the strong structural controllability are provided by the minimum zero forcing sets of the simple interconnection graph, which is the interconnection graph without its loops. In particular, we present the first quadratic time algorithm providing a minimum-size input set for the strong structural controllability of a self-damped or an undamped system with a tree structure.

Furthermore, this paper highlights the NP-hardness of the computation of the zero forcing number of a loop directed graph. However, since this graph invariant is mostly interesting (see Equality (1)) in the case of a loop directed tree (defined in Section 3), we identify a class of loop directed trees for which the zero forcing number can be computed in linear time.

The outline of this paper is as follows: in Section 2 we highlight that the computation of the zero forcing number of any loop directed graph is NP-hard. In Section 3 we identify a class of loop directed trees for which the zero forcing number can be computed in linear time. Section 4 is a review of some concepts and results about strong structural controllability. Section 5 highlights the equivalence between the zero forcing sets in a loop directed graph and the constraint matchings in the bipartite graph associated with $G$. From this equivalence, we re-state some results about the strong structural controllability of a system in terms of zero forcing sets. In Section 6, we use the notion of zero forcing sets in a simple graph in order to solve the problem of determining a minimum-size input set for the strong structural controllability of a self-damped or an undamped system with a tree structure. Section 7 contains concluding remarks.

## 2 NP-hardness

In this section, we highlight from known results that the computation of the zero forcing number of any loop directed graph is NP-hard.

**Definition 2.1.** Two matrices $A$ and $B$ are said to be *permutation similar* if there are permutation matrices $P_1, P_2$ such that $A = P_1BP_2$.

A zero-nonzero pattern $A$ (or pattern for short) of dimension $n$ is an $n \times n$ matrix whose each entry is either a star or zero. A star refers to a nonzero entry.

A loop directed graph $G$ with $n$ vertices defines a zero-nonzero pattern $A(G)$ of dimension $n$ as follows: the entry $a_{ij}$ of $A(G)$ is a star if and only if there is a directed edge from vertex $i$ to vertex $j$ in $G$. This pattern is called the zero-nonzero pattern associated with $G$.

**Definition 2.2.** Let $A$ be a zero-nonzero pattern.
- A \( t \)-triangle of \( A \) is a \( t \times t \) subpattern of \( A \) which is permutation similar to an upper triangular pattern whose all diagonal entries are nonzero.

- The **triangle number** of \( A \) is the maximum size of a triangle in \( A \).

- The triangle number \( \text{tri}(G) \) of a loop directed graph \( G \) is the triangle number of its associated zero-nonzero pattern \( A(G) \).

**Theorem 2.3.** \[5\] For any loop directed graph \( G \), \( \text{tri}(G) + Z(G) = |G| \).

The following theorem proved in \[6\] shows the link between the constraint matchings in a bipartite graph and the triangle number.

A zero-nonzero pattern \( A \) of dimension \( n \) defines a bipartite graph \( B_A \): the vertex sets are \( V = \{1, \ldots, n\} \) and \( V' = \{1', \ldots, n'\} \) and \( \{i, j'\} \) is an edge in \( B_A \) if and only if the entry \( a_{ji} \) is a star in \( A \). \( B_A \) is called the **bipartite graph associated with \( A \)**.

Notice that the bipartite graph associated with a loop directed graph \( G \) is equal to the bipartite graph associated with the pattern \( A(G) \).

**Theorem 2.4.** \[6\] Let \( A \) be an \( n \times n \) zero-nonzero pattern and \( B_A \) the bipartite graph associated with \( A \). Then the following statements are equivalent.

- \( B_A \) has a constraint \( n \)-matching

- \( A \) is permutation similar to a triangular pattern with nonzero diagonal elements.

From the previous theorem, we deduce that the size of a maximum constraint matching in \( B_A \) equals the triangle number of \( A \). However, in \[8\] it has been proved that the computation of the size of a maximum constraint matching in a bipartite graph is NP-hard. Therefore, so it is for the triangle number of a loop directed graph. From this result and Theorem 2.3 we have highlighted the NP-hardness of the computation of the zero forcing number of any loop directed graph.

**Theorem 2.5.** The computation of the zero forcing number of any loop directed graph is NP-hard.

### 3 The minimum rank of some loop directed trees

As said in the introduction, the zero forcing number of any loop directed tree provides the minimum rank of the tree (see Equation (1)). Here is the definition of such a graph.
Definition 3.1. If $G$ is a directed graph, its associated undirected graph $\hat{G}$ is the graph having the same vertex set, and $\{i,j\}$ is an edge in $\hat{G}$ when at least one of $(i,j),(j,i)$ is an edge in $G$.

A loop directed tree is a loop directed graph whose associated undirected graph is a tree allowing loops.

Nowadays, there are algorithms for the computation of the zero forcing number of a loop symmetric directed tree [5], which is a loop directed tree such that if $(i,j)$ is an edge, then $(j,i)$ is an edge too. However, as stated in [3], an efficient algorithm for the computation of the minimum rank of any loop directed tree is still needed.

In this section, we define the loop directed trees with almost loop-free symmetric components for which the minimum rank can be computed in linear time.

Definition 3.2. Let $T$ be a loop directed tree. The set $\{i,j\}$ $(i \neq j)$ is said to be a symmetric edge in $T$ if both directed edges $(i,j),(j,i)$ are in $T$. A vertex $v$ of $T$ is said to be connected to a symmetric edge $\{i,j\}$ of $T$ if $v = i$ or $v = j$.

By abuse of language, we consider that a symmetric edge of $T$ is an edge in $T$.

Denote $T_s$ the subgraph induced by the symmetric edges of $T$. The loops in $T$ on the vertices connected to a symmetric edge are included in $T_s$.

Definition 3.3. A loop directed tree $T$ is said to have almost loop-free symmetric components if each connected component of $T_s$ has at most one vertex with a loop.

It is well known [8] that in an undirected tree a maximum constraint matching can be found in linear time. In Theorem 3.7 we show that the bipartite graph associated with any loop directed tree that has almost loop-free symmetric components is a forest. Moreover, in Section 2 we have shown that the size of a maximum constraint matching in the bipartite graph associated with a loop directed graph equals the triangle number of the graph. Hence from these results, Theorem 2.8 and Equation (1), we deduce that the minimum rank of any loop directed tree that has almost loop-free symmetric components can be computed in linear time.

The following definition and lemmas will be needed to prove Theorem 3.7.

Definition 3.4. Let $B = (V,V',E)$ be a bipartite graph associated with a loop directed graph. A path $P$ in $B$ is symmetric if for any edge $\{i,j'\}$ in $P$, the edge $\{j,i'\}$ is also in $P$.

Remind the notation $\hat{T}$ which refers to the undirected tree associated to the loop directed tree $T$ (see Definition 3.1).

Lemma 3.5. Let $T$ be a loop directed tree and $B_T$ its associated bipartite graph. Any shortest path $P$ in $B_T$ between vertices $v,v'$ is symmetric.
Proof.

- If $P$ has length 1, then $P = \{\{v, v'\}\}$ and it is symmetric.

- If $P$ has length greater than or equal to three, suppose $P$ is not symmetric. This means that there is an edge $\{i, j\}$ with $i \neq j$ such that $\{j, i\}$ is not in $P$. From $P$, we deduce a path between $v$ and $i$ in $\hat{T}$ of the form $P_1(v, j) - \{j, i\}$ and since the edge $\{j, i\}$ is not in $P$, we deduce another path $P_2(i, v)$ between $i$ and $v$ in $\hat{T}$. This implies that in $\hat{T}$ there is a cycle containing at least three different vertices, which is a contradiction.

In the context of the previous lemma, a vertex $i$ of $T$ is said to be connected to an edge in $P$ if there is a vertex $j$ such that $\{i, j\}, \{j, i\}$ are in $P$.

**Lemma 3.6.** Let $T$ be a loop directed tree and $B_T$ its associated bipartite graph. Let $P$ be a shortest path between vertices $v, v'$ in $B_T$. Then $P$ is symmetric and among the vertices of $T$ connected to an edge in $P$, at least one has a loop.

**Proof.** From the previous lemma, we know that $P$ is symmetric.

- If exactly one vertex of $T$ is connected to an edge in $P$, then $P = \{\{v, v'\}\}$ and $v$ has a loop.

- Suppose the lemma true if $n$ vertices of $T$ are connected to an edge in $P$ and prove it if $n + 1$ vertices $v_1, \ldots, v_{n+1}$ of $T$ are connected to an edge in $P$. These vertices define a path in $\hat{T}$: $v_1 - v_2 - \ldots - v_{n+1}$ with $v_1 = v$ such that for any $1 \leq i \leq n$, $\{v_i, v'_i\} \in P$ and $\{v_{i+1}, v'_i\} \in P$. Therefore, $P$ contains a symmetric shortest path between $v_2, v'_2$ with exactly $n$ vertices of $T$ connected to an edge of this path. By induction, at least one of these vertices has a loop. Therefore, so it is for the vertices of $T$ connected to an edge in $P$.

**Theorem 3.7.** The bipartite graph $B_T$ associated with a loop directed tree $T$ that has almost loop-free symmetric components is a forest.

**Proof.** By induction on the number of vertices in $T$.

- If $T$ has two vertices and if $B_T$ has a cycle, then $T$ is made up of a symmetric edge whose both vertices have a loop, which is a contradiction.

- Suppose the theorem true if $T$ has $n$ vertices and prove it if it has $n + 1$ vertices. Designate any vertex of $T$ to be the root. Let $l$ be a leaf of $T$. Denote $f$ its father.
If \( \{l, f\} \) is not a symmetric edge, since \( B_{T-1} \) has no cycle, so it is for \( B_T \).

if \( \{l, f\} \) is a symmetric edge and \( l \) has no loop, then since \( B_{T-1} \) has no cycle, so it is for \( B_T \).

If \( \{l, f\} \) is a symmetric edge and \( l \) has a loop, if there is a cycle on \( l \), it means that there must be a path between vertices \( f \) and \( f' \) in \( B_{T-1} \). Hence there is a cycle \( P^* \) of the form:

\[
P^* = \{l, f'\} - P(f', f) - \{f, l'\} - \{l', l\}
\]

where \( P(f, f') \) is a shortest path in \( B_{T-1} \) between \( f \) and \( f' \). By the previous lemma, at least one vertex \( v \) of \( T \) connected to an edge in \( P(f', f) \) has a loop. The symmetric edges of \( T \) whose both vertices are connected to an edge of \( P^* \) are then all in a same connected component of \( T \), and this component has at least two vertices \( v \) and \( l \) with a loop. This contradicts the definition of \( T \).

\[\square\]

**Corollary 3.8.** The minimum rank of any loop directed tree that has almost loop-free symmetric components is computable in linear time.

Notice it is not indispensable to compute the size of a maximum constraint matching in \( B_T \) in order to compute the minimum rank of a loop directed tree with almost loop-free symmetric components. Indeed, the following elimination process defined in [5] could be used.

A **realization** \( A \) of a pattern \( A \) is a real matrix whose an entry is nonzero if and only if the corresponding entry in \( A \) is a star. We write \( A \in A \).

The minimum rank of a pattern \( A \) is the minimum possible rank of one of its realizations.

**Proposition 3.9.** [Elimination process] Let \( A \) be a zero-nonzero pattern and \( a_{st} \) be a star entry of \( A \) such that either row \( s \) or column \( t \) or both have exactly one star entry. Then,

\[
mr(A) = mr(A_0(s|t)) + 1,
\]

where \( A_0(s|t) \) is the pattern obtained from \( A \) by setting row \( s \) and column \( t \) to zero.

The minimum rank of a loop directed graph \( G \) is by definition the minimum rank of the pattern \( A(G) \) associated with \( G \).

**Theorem 3.10.** The minimum rank of any loop directed tree that has almost loop-free symmetric components is computable in linear time thanks to the elimination process defined in Proposition 3.9.

**Proof.** Simply notice that if the elimination process cannot be used to compute the minimum rank of such a graph, then there must be a cycle in \( B_T \).

\[\square\]
4 Background on strong structural controllability

4.1 Definitions and notations

In Section 2 a bipartite graph associated with a pattern of dimension $n$ was defined. This can be extended to rectangular patterns. Let $A$ be an $n \times m$ zero-nonzero pattern. The bipartite graph $B_A$ associated with $A$ has $V = \{1, ..., m\}$ and $V' = \{1', ..., n'\}$ as vertex sets. Besides, $\{i, j'\}$ is an edge in $B_A$ if and only if $(j, i)$-entry of $A$ is a star.

If the bipartite graph associated with a pattern $A$ has a constraint $t$-matching, then we say by abuse of language that $A$ has a constraint $t$-matching.

Let $T \subset V$ be a vertex subset in a loop directed graph $G$. A constraint $T$-less matching in the bipartite graph $B_G$ associated with $G$ is a constraint matching with no edges of the form $\{i, i'\}$ with $i \in T$. In particular, if $T = V$, a constraint $T$-less matching in $B_G$ is called a constraint self-less matching.

Let $A$ be an $n \times m$ pattern and $S \subset \{1, ..., m\}$. Then $A(\cdot | S)$ denotes the pattern obtained from $A$ by deleting the columns indexed by $S$.

Let $A$ be a pattern of dimension $n$. Then $A_\times$ is the pattern obtained from $A$ by putting stars along the diagonal. Similarly, $G_\times$ denotes the graph obtained from the graph $G$ by putting a loop on each vertex of $G$.

4.2 Strong structural controllability

In many real systems, the interaction strengths between the different components of the system are unknown or only partially known. In such a case, determining the controllability of the system from classical Kalman controllability condition is impossible. An alternative is to use only the structure of the system in order to get information about the system control. That is why weak and strong structural controllability have been introduced [10, 11].

Recall that a system with a given interconnection graph is weakly structurally controllable from an input set $S$ if we can choose interaction strengths making the system controllable from $S$. Instead, a system with a given interconnection graph is strongly structurally controllable from an input set $S$ if the system is controllable from $S$ whatever the interaction strengths.

Weak structural controllability is determined by the matchings in the interconnection graph (see [12] for more details) while strong structural controllability is determined by some constraint matchings in the bipartite graph associated with the interconnection graph. Minimum-size input sets for weak and strong structural controllability provide respectively a lower and an upper bound on the minimum
number of vertices to be controlled by the outside controller in order to have full control over the system.

In this paper, only the strong structural controllability of a system is of interest. We present it below. For a detailed description of weak structural controllability, see [12].

Denote $G$ the interconnection graph of the system. In the rest of the paper, $G$ will be referred as the loop directed graph modeling the system, in order to avoid ambiguity when considering a zero forcing set of $G$. Denote $S$ an input set of the system, meaning that only the vertices of $G$ that are in $S$ are directly controlled by the outside controller. Then $G$ and $S$ define respectively zero-nonzero patterns $A := A(G)$ and $B(S)$ whose specific realizations define the dynamics of the system, that is, there are $A \in A$ and $B \in B(S)$ such that $\dot{x}(t) = A^T x(t) + Bu(t)$.

Consider the system described by the graph in Figure 1. Then, $A = \begin{pmatrix} * & * & 0 \\ 0 & 0 & * \\ * & * & 0 \end{pmatrix}$

and if $S = \{1, 3\}$, $B(S) = \begin{pmatrix} * & 0 \\ 0 & 0 \\ 0 & * \end{pmatrix}$.

Strong structural controllability, or strong controllability for short, focuses on the pair $(A, B(S))$.

**Definition 4.1.** The pair $(A, B(S))$ is strongly $S$-controllable if all realizations $(A, B)$ are controllable (in the sense that the controllability matrix is full rank).

The following theorem is a test criterion proved in [7] for strong $S$-controllability.

In the following, $V_s$ is the set of vertices with a loop in the original loop directed graph $G$ modeling the system.
Theorem 4.2. \[ \text{[Test criterion]} \] Let \( G \) be the loop directed graph on \( n \) vertices modeling the system and \( S \) be an input set with cardinality \( m \leq n \). The pair \( (A, B(S)) \) is strongly \( S \)-controllable if and only if \( A(\cdot | S) \) has a constraint \((n - m)\)-matching and \( A_x(\cdot | S) \) has a constraint \( V_x \)-less \((n - m)\)-matching.

In order to carry out such a test for a given input set \( S \), an algorithm was presented in [7]. If the system is not strongly \( S \)-controllable, the algorithm computes a vertex set \( \tilde{S} \) containing \( S \) such that the system is strongly \( \tilde{S} \)-controllable. However, computing a minimum-size input set for strong structural controllability is a challenging problem. In that scope, the following theorem has been proved in [7].

A self-damped system is a system modeled by a graph with a loop on each vertex (each vertex’s state influences itself), whereas an undamped system is a system modeled by a loop-free graph (each vertex’s state is influenced by the state of some other vertices but not itself).

Theorem 4.3. \[ \text{[7]} \] Consider a self-damped or an undamped system modeled by a loop directed graph \( G \) on \( n \) vertices and a maximum constraint self-less \((n - m)\)-matching in the bipartite graph \((V, V', E)\) associated with \( G_x \), with unmatched vertices \( S' \subset V' \) and \( S \subset V \). Then \( S \) is a minimum-size input set for strong structural controllability.

This theorem provides thus a way to obtain a minimum-size input set for strong structural controllability in the case of a self-damped/undamped system. However, computing a maximum constraint matching in a bipartite graph is known to be NP-hard [8].

In the next section, we re-state these results in terms of zero forcing sets. These new statements show on the one hand that testing whether or not a system is strongly \( S \)-controllable is equivalent to testing if \( S \) is a zero forcing set of the loop directed graph modeling the system and on the other hand, that the zero forcing sets in a simple graph solve the problem of finding a minimum-size input set for strong controllability of a self-damped or an undamped system. In Section 6, we present the first efficient algorithm providing a minimum-size input set for the strong structural controllability of a self-damped or an undamped system with a tree structure.

5 Zero forcing sets and constraint matchings

In Section 2 we have shown from known results that the zero forcing number of a loop directed graph \( G \) is equivalent to the size of a maximum constraint matching in the bipartite graph \( B_G \) associated with \( G \). In this section, we prove
that more than having equivalent sizes, the zero forcing sets of $G$ are equivalent to the constraint matchings of $B_G$. From this result, we re-state the results about strong structural controllability presented in the previous section in terms of zero forcing sets.

**Definition 5.1.** [3] Let $G$ be a loop directed graph.

- Suppose that any vertex of $G$ is either black or white. When the color change rule (cf introduction) is applied to vertex $i$ to change the color of vertex $j$, we say that $i$ *forces* $j$ and write $i \rightarrow j$.

- Given a zero forcing set of $G$, we can list the forces in order in which they were performed to color the vertices of $G$ in black. This list is called a chronological list of forces.

Notice that given a zero forcing set, a chronological list of forces is not necessarily unique. However, unicity is not required here.

Theorem 5.4 is a consequence of the following theorem proved in [8].

**Theorem 5.2.** [8] Let $B = (V, V', E)$ be a bipartite graph and $M$ be a matching in $B$. The following assertions are equivalent:

- $M$ is a constraint matching
- We can order the vertices of $V$, $i_1, ..., i_n$ and the vertices of $V'$, $j'_1, ..., j'_n$ such that for any $1 \leq k \leq |M|$, $\{i_k, j'_k\} \in M$ and for any $1 \leq l < k \leq |M|$, $\{i_l, j'_l\} \notin E$.

The following lemma is a direct consequence of the previous theorem.

**Lemma 5.3.** Let $G = (V, E)$ be a loop directed graph, $B_G$ its associated bipartite graph and $\mathcal{M} := \{\{i_1, j'_1\}, ..., \{i_t, j'_t\}\}$ a constraint matching in $B_G$. Then, $V \setminus \{i_1, ..., i_t\}$ is a zero forcing set in $G$ with chronological list of forces $j_1 \rightarrow i_1, ..., j_t \rightarrow i_t$.

Finally, we deduce the following theorem.

**Theorem 5.4.** Let $G$ be a loop directed graph and $B_G$ the bipartite graph associated with $G$. Then, a vertex subset of $G$ is a (minimum) zero forcing set of $G$ with a chronological list of forces $j_1 \rightarrow i_1, j_2 \rightarrow i_2, ..., j_t \rightarrow i_t$ if and only if $\mathcal{M} := \{\{i_1, j'_1\}, \{i_2, j'_2\}, ..., \{i_t, j'_t\}\}$ is a (maximum) constraint matching in $B_G$.

Thanks to the previous result, we can re-state Theorems 4.2 and 4.3 in terms of zero forcing sets.
Theorem 5.5. Let $G$ be the loop directed graph on $n$ vertices modeling the system and $S$ be an input set with cardinality $m \leq n$. The pair $(A, B(S))$ is strongly $S$-controllable if and only if $S$ is a zero forcing set of $G_x$ for which there is a chronological list of forces that does not contain a force of the form $i \rightarrow i$ with $i \in V_s$ and a zero forcing set of $G$.

Notice that no more than the definition of a zero forcing set is needed to test if a system is strongly $S$-controllable. Testing if a vertex subset is a zero forcing set of a graph on $n$ vertices can be done in $O(n^2)$. As an example, consider the system modeled by the loop directed graph in Figure 1 and check that the system is strongly $S$-controllable for $S = \{1\}$. We immediately check that $S$ is a zero forcing set of $G$ and that $S$ is a zero forcing set of $G_x$ with chronological list of forces: $1 \rightarrow 2, 2 \rightarrow 3$. Since in this list vertex 1 does not force itself, the system modeled by $G$ is then strongly controllable from $S$.

Theorem 4.3 is equivalent to the following.

Theorem 5.6. Consider a self-damped or an undamped system modeled by a loop directed graph $G$ and a minimum zero forcing set $S$ of $G_x$ for which there is a chronological list of forces with no forces of the form $i \rightarrow i$. Then, $S$ is a minimum-size input set for the strong structural controllability of the system.

This theorem can then be stated in terms of zero forcing sets in a simple graph. The simple graph $G_s$ is obtained from the loop directed graph $G$ by removing the loops on its vertices.

Recall that in a simple graph, a vertex must be black to be able to force one of its out-neighbors.

Corollary 5.7. Consider a self-damped or an undamped system modeled by a loop directed graph $G$. Let $S$ be a minimum zero forcing set in the simple graph $G_s$ associated with $G$. Then, $S$ is a minimum-size input set for the strong structural controllability of the system.

This shows that the minimum zero forcing sets in the simple graph $G_s$ provide minimum-size input sets for strong structural controllability of self-damped or undamped systems. However, computing a minimum zero forcing set in a simple graph is a challenging problem. In the next section, we provide an algorithm computing a minimum zero forcing set in a simple tree. This solves the problem of finding a minimum-size input set for the strong structural controllability of a self-damped or an undamped system with a tree structure.
6 The case of systems with a tree structure

The term ‘tree’ refers equivalently to an undirected tree or to a symmetric directed tree, which is obtained from an undirected tree by replacing each edge \{i, j\} by both directed edges (i, j), (j, i). Trees allow loops, whereas simple trees do not.

The goal of this section is to present an algorithm computing a minimum zero forcing set in a simple tree. This solves the problem of finding a minimum-size input set for the strong structural controllability of a self-damped or an undamped system with a tree structure.

Consider a simple tree \( T \) and a vertex \( v \) of \( T \). \( T - v \) denotes the forest obtained from \( T \) by removing vertex \( v \) as well as the edges incident to \( v \). Denote \( T_v^1, ..., T_v^k \) the connected components of \( T - v \). If at least two of them are paths connected to \( v \) from one of their endpoints, then \( v \) is called an 

\textbf{appropriate vertex}.

The algorithm computing a minimum zero forcing set in a simple tree is presented in Algorithm 1.

In order to prove Algorithm 1, we need the following results.

\textbf{Lemma 6.1.} The zero forcing number of a simple path is 1 and any minimum zero forcing set is made up of one of its endpoints.

\textbf{Proposition 6.2.} [13] Any simple tree with at least three vertices has an appropriate vertex.

\textbf{Theorem 6.3.} [13] Let \( T \) be a simple tree on \( n \geq 3 \) vertices and an appropriate vertex \( v \) of \( T \). \( T_v^1, ..., T_v^k \) denote the connected components of \( T - v \). Then,

\[ \text{mr}(T) = \text{mr}(T_v^1) + ... + \text{mr}(T_v^k) + 2. \]

Recall that for any simple tree \( T \), \( \text{mr}(T) = |T| - Z(T) \).

\textbf{Theorem 6.4.} Algorithm 1 is correct.

\textit{Proof.} Denote \( n \) the number of vertices in \( T \). The proof is an induction on \( n \).

- If \( n = 1 \), Algorithm 1 returns \( S \) containing the unique vertex of \( T \). Therefore, \( S \) is trivially a minimum zero forcing set of \( T \).

- Suppose the theorem true if \( T \) has less than \( n \) vertices and prove it for \( T \) having \( n \) vertices. If \( T \) has no appropriate vertex, then \( n = 2 \) and \( T \) is a path. In that case, Algorithm 1 returns \( S \) containing a vertex of \( T \). From Lemma \[6.1\], \( S \) is a minimum zero forcing set of \( T \). If \( n > 2 \), Proposition \[6.2\] claims that \( T \) has an appropriate vertex \( v \). Denote \( T_v^1, ..., T_v^k \) the connected components of \( T - v \).
Suppose that $T^1_v, ..., T^l_v$ ($2 \leq l \leq k$) are the paths connected to $v$ from one of their endpoints. The vertex set $S$ computed by Algorithm 1 is of the form

$$S = S' \cup \bigcup_{j=l+1}^{k} S_j,$$

where $S_j$ ($l + 1 \leq j \leq k$) is by induction a minimum zero forcing set of $T^j_v$ and $S'$ has been built by considering $l - 1$ paths among $T^1_v, ..., T^l_v$, say for example $T^1_v, ..., T^{l-1}_v$, and for each of these paths the endpoint which is a leaf in $T$ has been put in $S'$. When applying the color change rule to $T$ with the vertices in $S$ initially black, the $l - 1$ components $T^1_v, ..., T^{l-1}_v$ will be colored in black as well as vertex $v$. Once $v$ is black, by induction, the other components $T^j_v$ ($l + 1 \leq j \leq k$) will be colored in black. Finally, $v$ will force the endpoint of $T^l_v$ connected to $v$ and $T^l_v$ will be entirely colored in black as well. So, $S$ is a zero forcing set of $T$.

We have to prove now that $S$ has a minimum size, i.e $mr(T) = n - |S|$.

From Theorem 6.3 we know that

$$mr(T) = 2 + \sum_{i=1}^{l} mr(T^i_v) + \sum_{j=l+1}^{k} mr(T^j_v).$$

From Equation (1) and Lemma 6.1 we deduce

$$mr(T) = 2 + \sum_{i=1}^{l} (|T^i_v| - 1) + \sum_{j=l+1}^{k} (|T^j_v| - |S_j|).$$

Finally, since $\sum_{i=1}^{k} |T^i_v| = n - 1$,

$$mr(T) = n - (l - 1 + \sum_{j=l+1}^{k} |S_j|).$$

Hence, since $|S| = l - 1 + \sum_{j=l+1}^{k} |S_j|$,

$$mr(T) = n - |S|.$$

Notice that since finding an appropriate vertex can be done in $O(n)$ time from a depth-first search and since at each iteration at least three vertices are removed from $G$, our algorithm has a $O(n^2)$ complexity.
7 Conclusions

In the first part of this paper, we have first highlighted the NP-hardness of the computation of the zero forcing number of any loop directed graph. Then we have identified a class of loop directed trees, namely the loop directed trees with almost loop-free symmetric components, for which the zero forcing number/minimum rank can be computed in linear time.

Currently, there are thus algorithms for the computation of the minimum rank of any loop symmetric directed tree [5] and any loop directed tree with almost loop-free symmetric components. As said in [3], an efficient algorithm for the computation of the minimum rank of any loop directed tree is still needed.

In the second part of the paper, we have shown the link between the zero forcing sets in the loop directed graph modeling a system and the strong structural controllability of the system. More specifically, we have re-stated some known results [7] about strong structural controllability in terms of zero forcing sets. These new statements have shown two results. Firstly, testing whether or not a system is strongly $S$-controllable from an input set $S$ is equivalent to checking if $S$ is a zero forcing set in the interconnection graph. Secondly, the zero forcing sets in the simple interconnection graph $G_s$, which is the interconnection graph $G$ without the loops, provide minimum-size input sets for the strong structural controllability of a self-damped or an undamped system. Finally, we have presented the first quadratic time algorithm determining a minimum-size input set for the strong structural controllability of a self-damped or an undamped system with a tree structure.

A similar work was done in [14] where the link between the zero forcing sets of a simple undirected graph and the controllability of a quantum system has been proved.

All these results show the role of the zero forcing sets in the study of the dynamics of networked systems.
Algorithm 1:

Input: a simple tree $T$;
Output: a minimum zero forcing set $S$ of $T$;
Set $S = \emptyset$;
$G = T$;
While $G$ is non empty

- if $G$ is a forest whose each connected component has less than 3 vertices, then put a vertex of each component in $S$ and set $G = \emptyset$;

- else consider an appropriate vertex $v$ of $G$ in a connected component $T_i$ of $G$;
  - denote $k(\geq 2)$ the number of connected components of $T_i - v$ which are paths connected to $v$ from an endpoint;
  - for $k - 1$ among them, put in $S$ the endpoint which is a leaf in $T_i$;
  - remove $v$ from $G$ as well as the $k$ paths connected to $v$ from an endpoint;

end

References

[1] L. Hogben, Spectral Graph Theory and the Inverse Eigenvalue Problem of a Graph, Chamchuri Journal of Mathematics 1 (2009) 51-72.

[2] S. Fallat, L. Hogben, The Minimum Rank of Symmetric Matrices Described by a Graph : A Survey, Linear Algebra and Its Applications 426 (2007) 558-582.

[3] L. Hogben, Minimum rank problems, Linear Algebra and Its Applications 432(8) (2010) 1961-1974.

[4] AIM minimum rank - special graphs work group, Zero forcing sets and the minimum rank of graphs, Linear Algebra and Its Applications 428:1628-1648 (2008).
[5] F. Barioli, S. Fallat, D. Hershkowitz, H.T. Hall, L. Hogben, H. van der Holst, B. Shader, On the minimum rank of not necessarily symmetric matrices: a preliminary study, Electron. J. Linear Algebra, 18(2009), pp. 126-145.

[6] D. Hershkowitz, H. Schneider, Ranks of zero patterns and sign patterns, Linear Multilinear Algebra, 34(1993), pp. 3-19.

[7] A. Chapman, M. Mesbahi, Strong structural controllability of networked dynamics, 6141-6146, in Proc. American Control Conference (2013).

[8] M.C. Golumbic, T. Hirst, M. Lewenstein, Uniquely Restricted Matchings, Algorithmica, 31(2001), no. 2, pp. 139-154.

[9] S. Mishra, On the Maximum Uniquely Restricted Matching for Bipartite Graphs, Electron. Notes Discrete Math., 37(2011), pp. 345-350.

[10] C.-T. Lin, Structural controllability, IEEE Trans. on Automatic Control, vol. AC-19, no. 3, pp. 201-208, Aug. 1974.

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

[12] Y.-Y. Liu, J.-J. Slotine, A.-L. Barabasi, Controllability of complex networks, Nature, vol. 473, no. 7346, pp. 167-173, May 2011.

[13] I.-J. Kim, Algorithmic proof of Maxmult($T$) = $p(T)$, Commun. Korean Math. Soc. 27(2012), No.4, pp. 665-668.

[14] D. Burgarth, D’Alessandro, L. Hogben, S. Severini, M. Young, Zero forcing, linear and quantum controllability for systems evolving on networks, IEEE Trans. on Automatic Control 58, 2349 (2013).