Preprocessing Graphs for Network Inference Applications

H. R. Sachin Prabhu and Hua-Liang Wei

Department of Automatic Control & Systems Engineering, The University of Sheffield,
Mappin Street, S1 3JD, Sheffield, U.K.

Keywords: Bipartite Graphs, Reduced Graphs, Ordered Matching, Rank, Subnetworks.

Abstract: The problem of network inference can be solved as a constrained matrix factorization problem where some sparsity constraints are imposed on one of the matrix factors. The solution is unique up to a scaling factor when certain rank conditions are imposed on both the matrix factors. Two key issues in factorising a matrix of data from some netwrok are that of establishing simple identifiability conditions and decomposing a network into identifiable subnetworks. This paper solves both the problems by introducing the notion of an ordered matching in a bipartite graphs. Novel and simple graph theoretical conditions are developed which can replace the aforementioned computationally intensive rank conditions. A simple algorithm to reduce a bipartite graph and a graph preprocessing algorithm to decompose a network into a set of identifiable subsystems is proposed.

1 INTRODUCTION

The problem of network inference arises when regulatory pattern of a network is known and its outputs are measured whereas the inputs that drive the network and regulatory strengths are unknown. A regulatory pattern indicates causal relationships between inputs and outputs of a network. In terms of steady-state analysis of systems, input-output relationships can be represented as a system of linear equations where the coefficients of which represent steady-state gains. The challenge is to simultaneously estimate the regulatory strengths and input activities. In other words, a data matrix is to be factorised into a product of two matrices – a regulatory matrix and an input matrix - such that the error in data reconstruction is minimised. Such problems are common in studies on social networks and biological regulatory networks (Newman, 2003; Brugere et al., 2016).

It is hard to simultaneously estimate the matrix factors using conventional techniques such as Principal Component Analysis or Singular Value Decomposition as the structure of the regulatory matrix is constrained (Liao et al., 2003). Network Component Analysis (NCA) (Liao et al., 2003) solves the network inference problem as a bilevel optimisation problem while taking these constraints into consideration. NCA imposes several rank conditions on the regulatory matrix and input matrix in order to ensure uniqueness to a certain degree of the estimates obtained via optimisation. A regulatory network must satisfy all relevant NCA rank conditions imposed on it whereas the input matrix can only be assumed to satisfy the conditions imposed on it.

The patterns found in complex real world networks are not random (Newman, 2003). Therefore, there is a need to characterise such networks whenever possible. Regulatory networks can be formally described using graphs and the benefits of doing so are multifold. Parameter estimation is relatively easier as the solution space is well defined. Graph theoretical descriptions are more comprehensible to a layman than matrix rank conditions. In addition to that, subnetworks can be identified in cases where parameter estimation for the original network is unsolvable. These advantages motivated development of graph theoretical interpretations of regulatory networks in (Boscolo et al., 2005) and (Fritzilas et al., 2013). Identifying subnetworks refers to searching for a part of regulatory pattern that allows application of NCA in the context of this paper. It should not be mistaken for parameter estimation.

Graph theoretical conditions that are based on analysing the structure of regulatory matrix is developed in (Boscolo et al., 2005). These conditions are comprehensible and offer a simple way to test NCA compatibility of relatively smaller networks by inspection. However, a more formal description is possible. More importantly, proposed limit on number of outputs that an input can regulate is inaccurate. Maximal matching property of a graph is used in (Fritzilas et al., 2013) to obtain a formal and computationally simpler conditions to test a network for its NCA compatibility. Though the matching condition

Prabhu, H. and Wei, H-L.
Preprocessing Graphs for Network Inference Applications.
DOI: 10.5220/0006401104060413
In *Proceedings of the 14th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2017) - Volume 1*, pages 406-413
ISBN: 978-989-758-263-9

can greatly reduce the computational burden, there are cases where some networks can be erroneously classified as NCA compatible. Furthermore, both approaches present algorithms to identify largest NCA compatible subnetwork, but they ignore rest of the original network.

Two key problems addressed in this paper are that of deriving an accurate comprehensible formal description of NCA compatible regulatory networks and decomposing NCA incompatible networks into a set of NCA compatible subnetworks without ignoring any part of the original network. In this paper, a new graph property called ordered matching is introduced. A formal description of NCA compatibile networks based on finding a maximal ordered matching in a reduced graph is developed. A new algorithm to reduce a graph is proposed and a method to decompose a given network into a set of NCA compatible subnetworks is proposed.

The paper is organised as follows - section 2 presents a formal description of the regulatory network inference problem followed by graph theoretical interpretations of NCA found in literature. Shortcomings of those interpretations of NCA are demonstrated in section 3 with the help of counter examples. Novel NCA conditions are developed thereafter. Algorithms to reduce a graph, test a network for NCA compatibility, and identify all NCA compatible subnetworks in a given network are presented in the latter part of section 3. The results of application of these algorithms to an example network is presented in section 4.

2 PRELIMINARIES

2.1 Regulatory Network Inference

Consider a regulatory network with M outputs and L inputs, $L \leq M$. Let $D \in \mathbb{R}^{M \times N}$ be the output or data matrix and $S \in \mathbb{R}^{L \times N}$ be the input or source matrix. D_{ik} and $S_{jk}, 1 \leq k \leq N$ respectively denote kth sample of ith output and jth input. At steady-state, the system is represented by a set of linear equations given by

$$D = WS \tag{1}$$

Let $Y = \{v_i, 1 \leq i \leq M\}$ be the set of vertices corresponding to the outputs, $X = \{u_j, 1 \leq j \leq L\}$ be the set of vertices corresponding to the inputs, and $E = \{(u_j, v_i), u_j \in X \text{ regulates } v_i \in Y\}$ be a set of edges. Let $W \in \mathbb{R}^{M \times L}$ encode all the edge strengths. The system in (1) is a bipartite graph $G = \{X \cup Y, E, W\}$ with X and Y as its bipartitions. The regulatory pattern of the system is given by the adjacency matrix of G $B(G) \in [0,1]^{M \times L}$ defined as

$$B(G)_{ij} = \begin{cases} 1, & \text{if } (u_j, v_i) \in E \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

A mapping $\phi : \mathbb{R}^{M \times L} \mapsto [0,1]^{M \times L}$ can be defined such that

$$\phi(W) = B(G)$$

The problem of regulatory network inference is such that $B(G)$ and D are known, whereas W and S are to be estimated simultaneously. Regulatory network inference problem is posed as an optimisation problem below:

$$\begin{array}{rl} \min\limits_{W,S} & \mathcal{J}(W,S) = \|D - WS\|_F^2 \\ \text{subject to} & \phi(W) = B(G) \end{array} \tag{3}$$

where, $\| \, . \, \|_F$ denotes the *Frobenius* norm of a matrix.

Note that the objective is to be minimised simultaneously over W and S. Thus, this cannot be solved as a standard quadratic program in one optimisation variable. Matrix factorization approaches such as Principal Component Analysis and Singular Value Decomposition are not applicable here because of the constraint in (3). Network Component Analysis (NCA) (Liao et al., 2003) proposes a bi-level optimization based approach to factorize matrix D into a product of two matrices $\hat{W}$ and $\hat{S}$.

The following set of conditions is established in (Liao et al., 2003) that W and S must satisfy for NCA to be applicable:

1. W has full column rank, i.e., $rank(W) = L$
2. Reduced submatrices $W_{u_j}, \forall u_j \in X$ have full column rank, i.e., $rank(W_{u_j}) = L - 1$
3. S has full row rank, i.e., $rank(S) = L$

where, the reduced submatrix W_{u_j} of W is obtained by deleting from W the jth column and every ith row such that $\phi(W)_{ij} = 1$. As W is unknown apriori, it is to be replaced by $B(G)$ and W_{u_j} by $B(G)_{u_j} = B(W_{u_j})$. Consider a system with $M = 10, L = 5$, and W such that

$\phi(W) = B(G)$, where $B(G)$ is given by

$$B(G) = \begin{array}{c|ccccc} & u_1 & u_2 & u_3 & u_4 & u_5 \\ \hline v_1 & 1 & 1 & 0 & 1 & 0 \\ v_2 & 0 & 0 & 1 & 1 & 0 \\ v_3 & 0 & 0 & 1 & 1 & 0 \\ v_4 & 1 & 1 & 0 & 1 & 0 \\ v_5 & 0 & 0 & 1 & 1 & 1 \\ v_6 & 0 & 1 & 0 & 1 & 1 \\ v_7 & 1 & 0 & 1 & 0 & 1 \\ v_8 & 1 & 1 & 1 & 0 & 1 \\ v_9 & 1 & 0 & 0 & 0 & 1 \\ v_{10} & 0 & 1 & 1 & 0 & 1 \end{array} \tag{4}$$

$B(G)_{u_1}$ is obtained by deleting rows and columns as shown in Fig. 1.

	~~u_1~~	u_2	u_3	u_4	u_5
~~v_1~~	~~1~~	~~1~~	~~0~~	~~1~~	~~0~~
v_2	~~0~~	0	1	1	0
v_3	~~0~~	0	1	1	0
~~v_4~~	~~1~~	~~1~~	~~0~~	~~1~~	~~0~~
v_5	~~0~~	0	1	1	1
v_6	~~0~~	1	0	1	1
~~v_7~~	~~1~~	~~0~~	~~1~~	~~0~~	~~1~~
~~v_8~~	~~1~~	~~1~~	~~1~~	~~0~~	~~1~~
~~v_9~~	~~1~~	~~0~~	~~0~~	~~0~~	~~1~~
v_{10}	~~0~~	1	1	0	1

Figure 1: Deleting rows and columns to obtain $B(H_{u_1}(G))$.

It can be easily verified that $rank(B(G)) = L$. However, any regulatory network with $\phi(W) = B(G)$ as described in (4) will not be NCA compatible as the submatrices $B(G)_{u_j}$s are such that

$$rank(B(G)_{u_j}) \begin{cases} = L-1, & j = 1,\ 2,\ 4 \\ < L-1, & j = 3,\ 5 \end{cases}$$

Testing for NCA compatibility is computationally intensive as it involves checking ranks of $L+1$ matrices for a system with L inputs. Graph theoretical interpretation of these conditions can reduce the involved complexity to a great extent as explained in the next section.

2.2 Graph Theoretical Interpretations of NCA

Let $\mathcal{N}(u_j) = \{v_i \in Y | (u_j, v_i) \in E\}$ be the set of neighbours of some vertex $u_j \in X$. Trivially, $\mathcal{N}(u_j) = \{v_i | W_{ij} = 1\}$. Given a graph G, $G - V$ denotes an induced subgraph obtained by deleting the set of vertices V and all associated edges from G. A graph theoretical interpretation of submatrix W_{u_j} of W, referred to as reduced matrix in NCA literature, is defines as

Definition 1. *An induced subgraph in the sense of NCA $H_{u_j}(G)$ of G is defined as $H_{u_j}(G) = G - \{u_j, \mathcal{N}(u_j)\}$.*

Definition 2. *Let $W \in \mathbb{R}^{M \times L}$ be such that $\phi(W) = B(G)$. An induced submatrix W_{u_j} is obtained by deleting rows and columns of W such that $\phi(W_{u_j}) = B(H_{u_j}(G))$*

$B(H_{u_j}(G))$ is nothing but the submatrix $B(G)_{u_j}$ described in section 2.1.

Various constraints are imposed on NCA compatible networks in (Boscolo et al., 2005) that are more comprehensible compared to the original NCA rank conditions. However, no computationally simple algorithm is given to implement those. Moreover, the condition imposed on the degree of input vertices $d(u_j) = |\ \mathcal{N}(u_j)| \leq L-1$ is inaccurate as demonstrated in the next section. The interpretation in definition 2 is implicitly introduced in (Fritzilas et al., 2013) where it is argued that NCA conditions are equivalent to finding a maximal matching of size $L-1$ in $H_{u_j}(G)$.

Definition 3. *A matching in a bipartite graph $G = \{X \cup Y, E, W\}$ is a pair $(X_1, Y_1), X_1 \subseteq X, Y_1 \subseteq Y$ such that $(u_j, v_i) \in E$, $|\mathcal{N}(u_j) \cap Y_1| = 1$, and $|\mathcal{N}(v_i) \cap X_1| = 1$, $\forall u_j \in X_1, v_i \in Y_1$.*

In other words, every vertex in a matching has a unique neighbour and $|X_1| = |Y_1|$ is the size of the matching (X_1, Y_1). A maximal matching (X_1^*, Y_1^*) is such that $|X_1^*| \geq |X_1|$, where (X_1, Y_1) is any other matching in G. Let G_1^* denote a subgraph defined by a maximal matching (X_1^*, Y_1^*). Its adjacency matrix $B(G_1^*)$ will have standard basis vectors e_i, i.e., vectors with 1 in ith position and rest of the entries being zero. Thus

$$rank(B(G_1^*)) = |X_1^*|$$

This fact is forms the basis for the argument that matching conditions are equivalent to the original NCA conditions.

This interpretation of NCA reduces the problem of computing rank of an induced submatrix to a computationally less intensive problem of finding a maximal matching in bipartite graph. However, the case of duplicate vertices is not considered which may lead to inaccurate descriptions of NCA compatible networks. Merely finding a matching in G will not suffice as demonstrated in the next section.

3 GRAPH PREPROCESSING FOR NCA

The problem of identifying NCA compatible subnetworks arises when a given regulatory network does not satisfy the original NCA conditions. This problem is addressed in (Boscolo et al., 2005) and (Fritzilas et al., 2013) where they first develop graph theoretical description of compatible regulatory networks and then propose methods to identify the largest NCA compatible subnetwork. However, the set of conditions claimed to be equivalent to the original NCA conditions are rather inaccurate as shown with the help of counter-examples in this section

Consider testing $B(H_{u_1}(G))$ given by

$$B(H_{u_1}(G)) = \begin{array}{c|cccc} & u_2 & u_3 & u_4 & u_5 \\ \hline v_2 & 0 & 1 & 1 & 0 \\ v_3 & 0 & 1 & 1 & 0 \\ v_5 & 0 & 1 & 1 & 1 \\ v_6 & 1 & 0 & 1 & 1 \\ v_{10} & 1 & 1 & 0 & 1 \end{array} \tag{5}$$

against the conditions in (Fritzilas et al., 2013). $(X_1^*, Y_1^*), X_1^* \subset X = \{u_2,\ u_3,\ u_4,\ u_5\}, Y_1^* \subset Y = \{v_2,\ v_3,\ v_5,\ v_6\}$ defines a maximal matching of size $L-1$. $B(H_{u_1,1}^*)$ defined by this matching as shown below

$$B(H_{u_1,1}^*) = \begin{array}{c|cccc} & u_2 & u_3 & u_4 & u_5 \\ \hline v_2 & 0 & 1 & 1 & 0 \\ v_3 & 0 & 1 & 1 & 0 \\ v_5 & 0 & 1 & 1 & 1 \\ v_6 & 1 & 0 & 1 & 1 \end{array} \tag{6}$$

Note that $rankB(H_{u_1,1}^*) = 3 < L-1$ as rows corresponding to v_2 and v_3 are identical. Therefore, it is possible to find a case where an induced submatrix $B(H_{u_j}(G))$ for some u_j has rank smaller than $L-1$, but contains a maximal matching of size $L-1$. This observation is summarised as follows

Remark 1. *Duplicate vertices can lead to wrong classification of a network as NCA compatible*

Consider the following submatrix of $B(H_{u_1}(G))$

$$B_1 = \begin{array}{c|cccc} & u_2 & u_3 & u_4 & u_5 \\ \hline v_2 & 0 & 1 & 1 & 0 \\ v_5 & 0 & 1 & 1 & 1 \\ v_6 & 1 & 0 & 1 & 1 \end{array}$$

$(\{v_2,\ v_5,\ v_6\}, \{u_2,\ u_3,\ u_4\})$ define a matching of size 3. Though $rank(B_1) = 3$, this matching does not represent a set of linearly independent columns - column u_4 is linearly dependent on column u_2 and u_3, $u_4 = u_2 + u_3$.

Remark 2. *A maximal matching may contain linearly dependent vertices*

Both remarks 1 and 2 show that a maximal matching does not necessarily correlate to the rank of a graph as argued in (Fritzilas et al., 2013).

It is evident that for $B(G)$ in (4), $d(u_j) > L-1, \forall u_j \in X$. It can be verified that a subnetwork formed by all rows of $B(G)$ and columns u_1, u_2, and u_4 is NCA compatible. This contradicts condition 2 of Lemma 1 in (Boscolo et al., 2005) (page 292) which limits d_u to $L-1$ for every $u_j \in X$.

Remark 3. *$d(u_j), u_j \in X$ can be greater than $L-1$*

Novel conditions that resolve the issues pointed out in remarks 1, 2 and 3 are presented in the next section.

3.1 Novel NCA Compatibility Conditions

Identifying a maximal matching (X_1^*, Y_1^*) is not necessarily equivalent to calculating rank of the original graph G

$$rank(B(G)) \neq |X_1^*|$$

as demonstrated in the previous section. The issue in remark 1 arises as the vertex v_3 is a duplicate of v_2 in the sense that $\mathcal{N}(v_2) = \mathcal{N}(v_3)$. Another maximal matching (X_2^*, Y_2^*) of $B(H_{u_1}(G))$ can be found where $X_2^* = \{u_2,\ u_3,\ u_4,\ u_5\}$ and $Y_2^* = \{v_2,\ v_5,\ v_6,\ v_{10}\}$. $B(H_{u_1,2}^*(G))$ defined by X_2^* and Y_2^* will contain all rows in (5) except the second row. It can be verified that the rank of this matrix is $L-1 = 4$, which is equal to $|X_2^*|$. This demonstrates the need to reduce a graph before finding a maximal matching. A reduced bipartite graph is defined as follows

Definition 4. *A reduced bipartite graph $\bar{G} = \{\bar{X} \cup \bar{Y}, \bar{E}, \bar{W}\}$ is a subgraph of G such that*

1. *$\bar{X} \subseteq X$ and $\bar{Y} \subseteq Y$*
2. *there are no isolated vertices, $d(v) > 0, \forall v \in \bar{X} \cup \bar{Y}$*

3. *there are no duplicate vertices,* $\mathcal{N}(v_1) \neq \mathcal{N}(v_2), \forall v_1, v_2 \in \bar{X} \cup \bar{Y}$

and, $\bar{E} \subseteq E$ *and* $\bar{W} \subseteq W$ *are submatrices whose rows and columns are indexed by* $\bar{X}$ *and* $\bar{Y}$*.*

A matching $(\{u_2,\ u_5,\ u_3\}, \{v_2,\ v_5,\ v_6\})$ in (6) represents linearly independent columns of $B(H^*_{u_1,1})$. This demonstrates the need to rearrange the rows and columns of the adjacency matrix of a bipartite graph before looking for a maximal matching. It will be shown in this section that imposing a partial order based on degrees of vertices is helpful in simplifying NCA compatibility conditions.

Definition 5. *An ordered matching is a matching* $(\mathcal{X}, \mathcal{Y})$ *obtained from partially ordered sets* $\mathcal{D}(X)$ *and* $\mathcal{D}(Y)$ *where* $\mathcal{D}(.)$ *sorts a set of vertices in increasing order of degrees of its elements.*

The act of imposing such an order is referred to as ordering a graph in the context of this paper. Let $\mathcal{D}(G)$ denote an ordered graph. Ordering and reduction operations are independent of each other and hence, can be executed in any order. The relationship between graph reduction and ordering, and rank of a graph can be derived as shown below.

Lemma 1. *The rank of a bipartite graph* $G = \{X \cup Y, E, W\}$ *such that* $|X| \leq |Y|$ *can be determined as*

$$\text{rank}(G) = |\bar{\mathcal{X}}^*(G)|$$

where, $(\bar{\mathcal{X}}^*(G), \bar{\mathcal{Y}}^*(G))$ *define a maximal matching in ordered reduced bipartite graph* $\mathcal{D}(\bar{G})$*.*

Proof Rank of a graph is equal to the rank of reduced graph $\bar{G}$ (Li et al., 2012). Defining a partial order $\mathcal{D}(.)$ on $\bar{G}$ does not affect its rank. It is sufficient to show that $(\bar{\mathcal{X}}^*, \bar{\mathcal{Y}}^*)$ a maximal matching in ordered reduced graph $\mathcal{D}(\bar{G})$ corresponds to a set of linearly independent columns in $\mathcal{D}(\bar{G})$, and hence, in $\bar{G}$ and G.

The ordered reduced graph $\mathcal{D}(\bar{G})$ has no duplicate or isolated vertices. Assume that there exists a vertex $u_i \in \mathcal{D}(\bar{X})$ the column corresponding to which can be expressed as a linear combination of other columns in the adjacency matrix $B(\mathcal{D}) = B(\mathcal{D}(\bar{G}))$ as

$$B(\mathcal{D})_{:i} = \sum_{j \neq i} \alpha_j B(\mathcal{D})_{:j}$$

Thus, for every $v \in \mathcal{N}(u_i)$, there exists some $u_j \in \mathcal{D}(\bar{X}), j \neq i$ such that $v \in \mathcal{N}(u_j)$. Since every vertex in a matching has exactly one neighbour, u_i will not be a part of any maximal ordered matching as the columns corresponding to u_js will precede the column corresponding to u_i in $B(\mathcal{D})$. Thus, all columns of $B(\bar{G})$ corresponding to $(\bar{\mathcal{X}}^*, \bar{\mathcal{Y}}^*)$ are linearly independent. A similar argument with respect to the rows of $B(\bar{G})$ by replacing u with v and $\mathcal{X}$ by $\mathcal{Y}$ completes the proof. □

A novel and simple set of conditions that is equivalent to the original NCA conditions can now be developed.

Theorem 1. *A regulatory network as described in (1) is NCA-compatible if and only if every input vertex* $u_j \in X$ *is such that*

$$|\bar{\mathcal{X}}^*(H_{u_j}(G))| = L - 1$$

where $G = (X \cup Y, E, W)$ *is a bipartite graph representing the network,* $H_{u_j}(G)$*s are induced subgraphs of G, and* $\bar{\mathcal{X}}^*(.)$ *represents a maximal ordered matching in* $\bar{G}$*.*

Proof If every vertex $u_j \in X$ satisfies the condition in theorem 1, $rank(W_{u_j}) = L - 1$ from Lemma 1. The graph $H_{u_j}(G) + v$ for any $u_j \in X$ and $v \in \mathcal{N}(u_j)$ is reduced and it will have a maximal ordered matching of size L and hence $rank(W) = L$.

Conversely, by definition, maximal ordered matchings of size $L - 1$ can be found for all H_{u_j}s of NCA compatible networks.

We use the argument on rank of S provided in (Liao et al., 2003). Thus, all original NCA conditions are satisfied. □

3.2 Identifying NCA Compatible Subnetworks

As pointed out towards the end of section 2.2 and demonstrated in section 3, the conditions developed in (Boscolo et al., 2005) and (Fritzilas et al., 2013) are inaccurate graph theoretical interpretations of original NCA conditions. Both present their own approaches to identify NCA compatible subnetworks of NCA incompatible networks. Algorithm proposed in the former one starts with an initial guess on the largest NCA compatible subnetwork and continues to add more input vertices such that no NCA conditions are violated. This requires an initial guess on largest NCA compatible subnetwork. The simplest starting point could be one where only one input vertex and its neighbours are considered, but it is trivial to see that randomly adding more vertices thereafter is a tedious task. The latter proposes a more structured approach where *nicely separable* subsets of X and Y are to be identified. Nicely separable NCA compliant subnetworks are those that have no vertices in common and are NCA compatible (Fritzilas et al., 2013). The motivation for our work comes from the fact that both former approaches look for the single largest NCA compatible subnetwork and ignore the rest of the vertices. In addition to that, neither of the two are computationally simple.

Several matching algorithms are available in literature. In depth discussions on such algorithms can be found in (Burkard et al., 2009). It is necessary to define a clear objective in order to find a suitable matching - for example, an algorithm designed to find online a maximum cardinality matching with minimum cost is proposed in (Azad et al., 2015). Our objective is to find matchings such that conditions of theorem 1 are satisfied. All the graphs that we consider for preprocessing phase are unlabelled and undirected bipartite graphs. Thus, a set of relatively simple algorithms proposed in this section are sufficient to meet our objectives.

In this section, we propose simple approaches to reduce a bipartite graph, identify a maximal ordered matching and decompose the whole network into NCA compatible subnetworks. If a system at steady state can be clearly represented as a regulatory network as described in (1), the algorithms presented in this section will not only avoid unnecessary computational overhead, but also provide a decomposition of the network without ignoring any of the subnetworks.

A method to remove duplicate and zero rows and columns is presentedbelow. This algorithm identifies duplicates by looking for zero entries in the inner product over binary field (Gudder and Latrémoliére, 2009) of $B(G)$ and its conjugate $\neg B(G)$.

Algorithm 1: Conjugate reduction.

```
input: $B \in \mathbb{R}^{M\times L}$
STEP 1:
Set $B_r = B(\neg B)^T + (\neg B)B^T$ and $M$ = number of rows in $B$
STEP 2:
for $i = 1$ to $M$ do
    for $j = i+1$ to $M$ do
        if $B_r(i,j) == 0$ then
            remove row $j$ from $B$
        end if
    end for
end for
STEP 3:
remove any zero rows from B
Repeat all steps with $B = B^T$
output: $B^T$
```

Reduced submatrix $B(\bar{G})$ can be obtained by applying Algorithm 1 on $B(G)$. In order to test a network against the conditions in theorem 1, a maximal ordered matching should be found in $B(\bar{G})$. In order to do that, the rows and columns of $B(\bar{G})$ are rearranged in increasing order of number of ones in them. Several matching algorithms available in literature can be used to find a maximal matching. However, a simple linear-search-and-eliminate based approach is sufficient as there is no need to minimise any associated cost. It can be shown that such an algorithm runs in $O(L)$ time for $B(\bar{G})$ with $|\bar{X}| = L$. The algorithm has not been described here for brevity.

We can now establish a simple approach to test NCA-compatibility of a given network as described in Algorithm 2.

Algorithm 2: NCA compatibility test.

```
input: $B(G) \in \mathbb{R}^{M\times L}$
if $\sum_i B_{ij}(G) > M - L + 1$ for any $u_j \in X$ then
    NCA-incompatible
else
    Obtain $B(\bar{G})$ using algorithm 1
    if for any $u_j \in \bar{X}, |\bar{X}^*(H_{u_j}(G))| < L-1$ then
        NCA-incompatible
    else
        NCA-compatible
    end if
end if
```

NCA compatible subnetworks can be identified if a network is found to be incompatible after applying Algorithm 2. This can be done by grouping together vertices $u_p, u_q \in X$ such that $u_p \cup \bar{X}^*(H_{u_p}(G))$ is identical to $u_q \cup \bar{X}^*(H_{u_q}(G))$ as outlined below. All vertices in a set $\mathcal{C}_n$ form maximal ordered matching of size n with other vertices in the same set. In Algorithm 3, $E_{:,\mathcal{C}_n}$ and $W_{:,\mathcal{C}_n}$ respectively represent submatrices of E and W with all rows and columns corresponding to the vertices in set $\mathcal{C}_n$.

Algorithm 3: Graph preprocessing for NCA.

```
input: $B(G) \in \mathbb{R}^{M\times L}$
STEP 1: for every $u_j \in X$ set
            $\mathcal{X}_j^* = \bar{X}^*(H_{u_j}(G)) \cup u_j$ and $n_j = |\mathcal{X}_j^*|$
STEP 2: identify sets $\mathcal{C}_n$ such that
            $u_p, u_q \in \mathcal{C}_n \Rightarrow \mathcal{X}_p^* \equiv \mathcal{X}_q^*, \forall u_p, u_q \in X$
outputs: set
            $G_n = \bigcup_{u_p \in \mathcal{C}_n} (u_p \cup Y), E_{:,\mathcal{C}_n}, W_{:,\mathcal{C}_n})$
```

Algorithm 3 is not only capable of identifying NCA compatible subnetworks, but also of implicitly testing a network for NCA compatibility. If a network is NCA compatible, Algorithm 3 returns one subnetwork G_{L-1} which is the original network G. Thus, Algorithm 3 alone can be used to preprocess a graph to identify NCA compatible subnetworks. The subnetworks can then be inferred individually and recombined to infer the whole network. Such a divide and conquer approach is beyond the scope of this paper.

Part of the results of applying the algorithms in this section to the regulatory network in (4) is presented in the next section.

4 RESULTS

In this section, we apply the algorithms presented in section 3.2 to the example network in (4). We demonstrate each step involved in obtaining ordered matchings $\bar{X}^*(H_{u_1}(G))$ and $\bar{X}^*(H_{u_2}(G))$. Similar steps are to be applied for vertices u_3, u_4 and u_5 (not shown here). The objective of the algorithms in (Boscolo et al., 2005) and (Fritzilas et al., 2013) is to identify largest possible NCA subnetwork while ignoring remainder of the network whereas our goal is to divide a given network into several NCA compatible subnetworks. Thus, a comparison between results of applying those algorithms with the ones presented in this section is unwarranted.

The first step in identifying NCA compatible subnetworks is identifying maximal ordered matchings for every vertex $u_j \in X$. Consider u_1 and u_2. The induced submatrix corresponding to u_1 $B(H_{u_1}(G))$ is given in (5) and that corresponding to u_2 $B(H_{u_2}(G))$ is given below.

$$B(H_{u_2}(G)) = \begin{array}{c|cccc} \hline & u_1 & u_3 & u_4 & u_5 \\ \hline v_2 & 0 & 1 & 1 & 0 \\ v_3 & 0 & 1 & 1 & 0 \\ v_5 & 0 & 1 & 1 & 1 \\ v_7 & 1 & 1 & 0 & 1 \\ v_9 & 1 & 0 & 0 & 1 \\ \hline \end{array} \qquad (7)$$

We reduce $B(H_{u_1}(G))$ and $B(H_{u_2}(G))$, impose the order $\mathcal{D}(.)$ on the reduced versions $B(\bar{H}_{u_.}(G))$ and then look for maximal matchings as demonstrated next. The steps in reducing matrices are illustrated by striking out duplicate rows and columns.

Reduced induced submatrices:

$$B(\bar{H}_{u_1}(G)) = \begin{array}{ccccc} \hline H_{u_1} & u_2 & u_3 & u_4 & u_5 \\ \hline v_2 & 0 & 1 & 1 & 0 \\ \cancel{v_3} & \cancel{0} & \cancel{1} & \cancel{1} & \cancel{0} \\ v_5 & 0 & 1 & 1 & 1 \\ v_6 & 1 & 0 & 1 & 1 \\ v_{10} & 1 & 1 & 0 & 1 \\ \hline \end{array}$$

$$B(\bar{H}_{u_2}(G)) = \begin{array}{ccccc} \hline H_{u_2} & u_1 & u_3 & u_4 & u_5 \\ \hline v_2 & 0 & 1 & 1 & 0 \\ \cancel{v_3} & \cancel{0} & \cancel{1} & \cancel{1} & \cancel{0} \\ v_5 & 0 & 1 & 1 & 1 \\ v_7 & 1 & 1 & 0 & 1 \\ v_9 & 1 & 0 & 0 & 1 \\ \hline \end{array}$$

It can be seen that $B(\mathcal{D}(\bar{H}_{u_1}(G))) = B(\bar{H}_{u_1}(G))$ as the columns and rows in $B(\bar{H}_{u_1}(G))$ are already in increasing order of degrees of vertices. However, the rows and columns of $B(\bar{H}_{u_2}(G))$ illustrated by encircled vertices must be reordered as indicated by the arrows here

Partial ordering:

$$B(\mathcal{D}(\bar{H}_{u_2}(G))) = \begin{array}{ccccc} \hline & u_1 & \overleftarrow{\textcircled{u_4}} & \overrightarrow{\textcircled{u_3}} & u_5 \\ \hline v_2 & 0 & 1 & 1 & 0 \\ \uparrow\textcircled{v_9} & 1 & 0 & 0 & 1 \\ \downarrow\textcircled{v_5} & 0 & 1 & 1 & 1 \\ \downarrow\textcircled{v_7} & 1 & 0 & 1 & 1 \\ \hline \end{array}$$

A linear search is conducted to look for the first 1 entry in every row, the row and column corresponding to the found entry are eliminated. These steps are repeated for all columns until no more 1 entry is found or all columns are exhausted. A set of vertices that correspond to all the columns in which a 1 entry is found defines a maximal matching. Finding first 1 entry in the first row of $B(\mathcal{D}(\bar{H}_{u_1}(G)))$ is illustrated by encircling the entry and eliminating the corresponding row and column by striking them out.

$$\begin{array}{ccccc} \hline \bar{H}_{u_1} & u_2 & \cancel{u_3} & u_4 & u_5 \\ \hline \cancel{v_2} & \cancel{0} & \textcircled{$\cancel{1}$} & \cancel{1} & \cancel{0} \\ v_5 & 0 & \cancel{1} & 1 & 1 \\ v_6 & 1 & \cancel{0} & 1 & 1 \\ v_{10} & 1 & \cancel{1} & 0 & 1 \\ \hline \end{array}$$

A maximal ordered matching in $\mathcal{D}(B(\bar{H}_{u_1}(G)))$ is obtained as

$$\begin{aligned}(\bar{X}^*(H_{u_1}(G)), \bar{\mathcal{Y}}^*(H_{u_1}(G))) \\ = (\{u_3,\ u_4,\ u_2,\ u_5\}, \{v_2,\ v_5,\ v_6,\ v_{10}\})\end{aligned}$$

The encircled entries represent this maximal ordered matching.

$\bar{H}_{u_1}$	u_2	u_3	u_4	u_5
v_2	0	①	1	0
v_5	0	1	①	1
v_6	①	0	1	1
v_{10}	1	1	0	①

Executing all the illustrated steps on all induced submatrices results in the following sets as indicated in Step 1 of Algorithm 3:

$$\begin{array}{lcll}
X_1^* = \{u_3,\, u_4,\, u_2,\, u_5\} & \cup & \{u_1\} & n_1 = 5 \\
X_2^* = \{u_3,\, u_4,\, u_1,\, u_5\} & \cup & \{u_2\} & n_2 = 5 \\
X_3^* = \{u_1,\, u_2,\, u_5\} & \cup & \{u_3\} & n_3 = 4 \\
X_4^* = \{u_1,\, u_3,\, u_2,\, u_5\} & \cup & \{u_4\} & n_4 = 5 \\
X_5^* = \{u_1,\, u_3\} & \cup & \{u_5\} & n_5 = 3
\end{array}$$

The NCA compatible sets obtained as described in Step 2 of Algorithm 3 are $C_5 = \{u_1,\ u_2,\ u_4\}$, $C_4 = \{u_3\}$, and $C_3 = \{u_5\}$. Thus, the given NCA incompatible network can be decomposed into a collection of 3 NCA compatible subnetworks as illustrated in Fig. 2

	u_1	u_2	u_4	u_3	u_5
v_1	1	1	1	0	0
v_2	0	0	1	1	0
v_3	0	0	1	1	0
v_4	1	1	1	0	0
v_5	0	0	1	1	1
v_6	0	1	1	0	1
v_7	1	0	0	1	1
v_8	1	1	0	1	1
v_9	1	0	0	0	1
v_{10}	0	1	0	1	1

Figure 2: Decomposition of original network into subnetworks.

5 CONCLUSIONS

In this paper, the need to consider graph theoretical interpretations of NCA was emphasized. It was demonstrated that merely finding a maximal matching in bipartite graphs may lead to wrong classification of NCA incompatible networks as NCA compatible. In order to overcome this issue, a new property called ordered matching in a bipartite graph was introduced. The rank of a bipartite graph was proven to be equal to the size of an ordered matching in its reduced form. This result was used to develop new conditions for NCA compatibility. A simple algorithm to reduce a bipartite graph was proposed. An algorithm was proposed to identify NCA compatible subnetworks in a given network. The results presented in this paper solve two important preprocessing problems - simplifying NCA compatibility conditions and decomposing a network into identifiable parts.

REFERENCES

Azad, A., Buluç, A., and Pothen, A. (2015). A parallel tree grafting algorithm for maximum cardinality matching in bipartite graphs. In *Proceedings of the 2015 IEEE International Parallel and Distributed Processing Symposium*, pages 1075–1084. IEEE Computer Society.

Boscolo, R., Sabatti, C., Liao, J. C., and Roychowdhury, V. P. (2005). A generalized framework for network component analysis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2(4):289–301.

Brugere, I., Gallagher, B., and Berger-Wolf, T. Y. (2016). Network structure inference, A survey: Motivations, methods, and applications. *CoRR*, abs/1610.00782.

Burkard, R., Dell'Amico, M., and Martello, S. (2009). *Assignment Problems, Revised Reprint*. Other titles in applied mathematics. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104).

Fritzilas, E., MilaniÄ, M., me Monnot, J., and Rios-Solis, Y. A. (2013). Resilience and optimization of identifiable bipartite graphs. *Discrete Applied Mathematics*, 161(4-5):593–603.

Gudder, S. and Latrémoliére, F. (2009). Boolean inner-product spaces and boolean matrices. *Linear Algebra and its Applications*, 431:274–296.

Li, H., Su, L., and Sun, H. (2012). On bipartite graphs which attain minimum rank among bipartite graphs with given diameter. *Electronic Journal of Linear Algebra*, 23:1–14.

Liao, J. C., Boscolo, R., Yang, Y., Tran, L. M., Sabatti, C., and Roychowdhury, V. P. (2003). Network component analysis: Reconstruction of regulatory signals in biological systems. *Proceedings of the National Academy of Sciences*, 100(26):15522–15527.

Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Review*, 45(2):167–256.