

Lecture 6

In the previous lecture, we defined the network coding model, showed an example of a transfer function and showed that transmitting information from source to terminal in a network is equivalent to checking whether the determinant of the transfer matrix can be made non-zero. In this lecture, we would discuss the Sparse Zeroes Lemma, introduce the concept of topological ordering and show the use of topological ordering in network coding.

In general, we can write down a transfer matrix in terms of indeterminates and analyze the determinant expression. We have seen in the last lecture that if the maximum flow to a terminal is not high enough, then the determinant cannot be made non-zero since it is a zero polynomial. However, if the maximum flow is high enough then the determinant cannot be made identically zero. e.g., In Figure 1, where the source v_1 has 3 inputs $X(v_1, 1)$, $X(v_1, 2)$ and $X(v_1, 3)$, and a terminal node v_3 has demands z_1 , z_2 and z_3 , the transfer matrix equation can be of the following form:

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = A \begin{bmatrix} X(v_1, 1) \\ X(v_1, 2) \\ X(v_1, 3) \end{bmatrix}$$

where A is the 3x3 transfer matrix. There is at least one set of assignments to the elements in A such that $\det(A)$ is non-zero. That assignment is the 3x3 unit matrix, which essentially implies that we channelize $X(v_1, 1)$, $X(v_1, 2)$ and $X(v_1, 3)$ along three different paths in the network all leading to v_3 without splitting any of $X(v_1, 1)$, $X(v_1, 2)$ and $X(v_1, 3)$, e.g., the paths can be $v_1-v_2-v_3$, $v_1-v_5-v_3$ and v_1-v_3 , and the demands satisfy $z_1 = X(v_1, 1)$, $z_2 = X(v_1, 2)$ and $z_3 = X(v_1, 3)$.

In general, if there is one source and multiple terminals, the paths overlap in a nontrivial manner which may result in network coding being needed.

Let the transfer function from the source to the i^{th} terminal be denoted by M_i . We want to ensure that $\det(M_i) \neq 0 \forall i$. This is equivalent to $\prod_{i=1}^{|T|} \det(M_i) \neq 0$. The left-hand-side expression in this equation is a polynomial expression in the indeterminates.

Note that a quadratic equation $ax^2 + bx + c = 0$ ($x \in \mathfrak{R}$) has at most two distinct roots i.e. there is an infinite number of reals where the polynomial $ax^2 + bx + c$ evaluates to a non-zero value. The same statement holds if x is chosen from finite fields as well. However in finite fields one may have to choose a field large enough such that values of the indeterminates exist such that the expression can be non-zero. Recall that the polynomial $z^2 + z$ evaluates to zero at all points in $GF(2)$. However when we go to $GF(3)$ (modulo-3 addition and multiplication) then at $z = 1$, it evaluates to $2 \bmod 3 = 2$, which is non-zero i.e. if we are willing to operate over a large enough field, then we can find an element over which the polynomial is non-zero.

Lemma 1 Sparse Zeroes Lemma: *Let $f(x_1, x_2, \dots, x_n)$ be a non-zero multivariate polynomial in variables x_1, x_2, \dots, x_n such that the maximum degree of f in any variable is d . Then in every finite field of size $q > d$ there exist values $x_1^*, x_2^*, \dots, x_n^*$ such that $f(x_1^*, x_2^*, \dots, x_n^*) \neq 0$.*

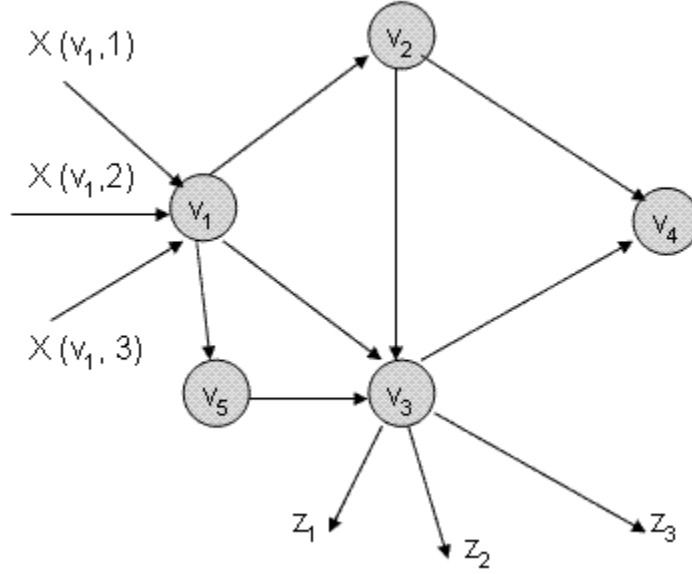


Figure 1: A DAG with sources and a terminal. The input sources to v_1 and the demands of the output processes at v_3 are shown.

Proof: We prove it by induction on the number of variables.

The base case is when $n = 1$. Then f is a polynomial in x_1 of degree at most d . It implies it has at most d roots. If $q > d$ then there exists a choice of x_1 such that $f(x_1) \neq 0$.

We state the induction hypothesis as follows. The claim stated in the lemma holds for all polynomials with at most $(n-1)$ variables. We can write $f(x_1, x_2, \dots, x_n)$ as a polynomial in x_n as follows: $f(x_1, x_2, \dots, x_n) = \sum_{k=0}^d f_k(x_1, x_2, \dots, x_{n-1})x_n^k$

Since f is not a zero polynomial at least one f_k is non-zero. Let us call it f_j . Now, $f_j(x_1, x_2, \dots, x_{n-1})$ is a non-zero polynomial in $(n - 1)$ variables with degree of any variable at most d . So the induction hypothesis applies for f_j , and hence there exist some $x_1^*, x_2^*, \dots, x_{n-1}^*$ such that $f_j(x_1, x_2, \dots, x_{n-1}) \neq 0$. Then $f(x_1^*, x_2^*, \dots, x_{n-1}^*, x_n)$ is a non-zero polynomial in the variable x_n with degree at most d . Here, we apply the induction hypothesis again, and the proof follows. \square

1 Transfer Matrices

When we think about network codes, the adjacency of edges is more important than the adjacency of nodes. Given a directed acyclic graph G , we can create another corresponding G_{line} where the edges in G can be the nodes of G_{line} , and if a node v in G has an incoming edge e_i and an outgoing edge e_j , then we create a pair of nodes e_i and e_j , with the corresponding entry in the adjacency matrix for G_{line} being β_{e_i, e_j} . The adjacency matrix for G_{line} has $|E|$ rows and $|E|$ columns. Figures 2 and 3 show a graph G and a graph G_{line} with nodes corresponding to edges in G , respectively. The adjacency matrix for G_{line} in Figure 3 is as follows:

$$\begin{bmatrix} 0 & 0 & 0 & \beta_{e_1,e_4} & \beta_{e_1,e_5} & 0 & 0 \\ 0 & 0 & 0 & \beta_{e_2,e_4} & \beta_{e_2,e_5} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \beta_{e_3,e_6} & \beta_{e_3,e_7} \\ 0 & 0 & 0 & 0 & 0 & \beta_{e_4,e_6} & \beta_{e_4,e_7} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

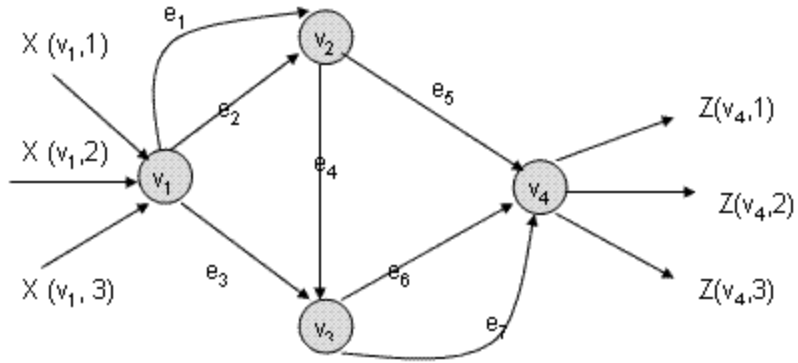


Figure 2: Graph G

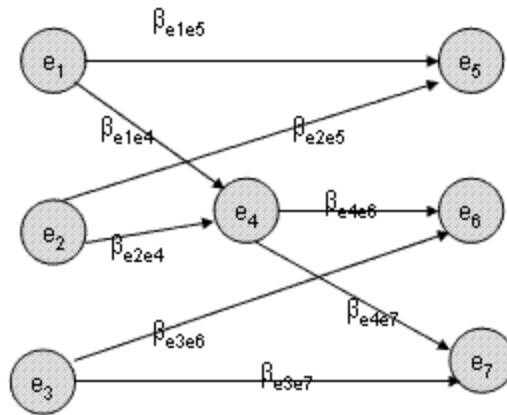


Figure 3: Graph G_{line} , where each edge in G is represented by a node. The label β_{e_i,e_j} on the edge between e_i and e_j in G_{line} shows what fraction of flow on edge e_i goes to edge e_j in G

2 Topological Ordering

For a directed graph G , a topological ordering is an ordering of nodes v_1, v_2, \dots, v_n such that if edge $(v_i, v_j) \in G$ then $i < j$, i.e., in a topologically ordered graph, an edge can only point from a

node with lower index to one with higher index. If a graph is not topologically ordered, we can sort it topologically by renaming the vertices. Figure 5 shows how the graph in Figure 4 can be topologically ordered by renaming v_1 to v_2' and v_2 to v_1' . However, for a graph with three or more nodes, the topological ordering may not be unique.

Suppose the adjacency matrix of G_{line} is denoted by F . If G_{line} is topologically ordered, then F is a strictly upper triangular matrix, i.e., all the entries in and below the diagonal are zero.



Figure 4: A graph not ordered topologically

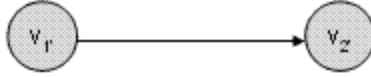


Figure 5: The graph of Figure 4 ordered topologically by renaming the nodes

Lemma 2 *If F is the adjacency matrix of a directed labelled line graph (topologically ordered), then $I - F$ is invertible.*

Proof: I has all 1's along the diagonal, and F has all 0's. Hence $(I - F)$ is an upper triangular matrix with all 1's on the diagonal. It implies $\det(I - F) = 1$ (it is possible to show by induction that any such matrix has a determinant 1), hence $I - F$ is invertible. \square

Consider a network with multiple sources and sinks. At node v , let the number of sources be $\mu(v)$. Note that $\mu(v)$ can be zero if there are no sources.

The input vector for node v can be defined as $\vec{x} = [X(v, 1) \ X(v, 2) \ X(v, 3) \ \dots \ X(v, \mu(v))]$

The total number of sources for the graph is the sum of the number of sources for all the nodes which receive input from some source, i.e., $\mu = \sum_{v_i \in V} \mu(v_i)$

Let A be a matrix of dimension $\mu|E|$ defined as follows:

$$A_{i,j} = \begin{cases} \alpha_{l,e_j} & \text{if } x_i = X(\text{tail}(e_j), l) \\ 0 & \text{otherwise} \end{cases}$$

i.e., if the tail of the edge e_j is the vertex v_i , and if node v_i receives input from a source $X(v_i, l)$, then α_{l,e_j} is the fraction of the input from $X(v_i, l)$ that goes to the edge e_j .

Similarly, let \vec{z} be the row vector of output processes from the terminal nodes. Let $\nu(v_j)$ denote the number of output processes at v_j . The total number of output processes for the graph is the sum of the number of output processes for all the terminal nodes, i.e., $\sum_{v_j \in V} \nu(v_j) = \nu$

For the output side, let us define a similar matrix B , which is a $\nu|E|$ matrix, as follows:

$$B_{i,j} = \begin{cases} \varepsilon_{e_j,l} & \text{if } z_i = Z(\text{head}(e_j), l) \\ 0 & \text{otherwise} \end{cases}$$

i.e., if the head of the edge e_j is the vertex v_i , and if node v_i has a output process $Z(v_i, l)$, then $\varepsilon_{e_j,l}$ is the fraction of the input from e_j that goes to the output $Z(v_i, l)$.

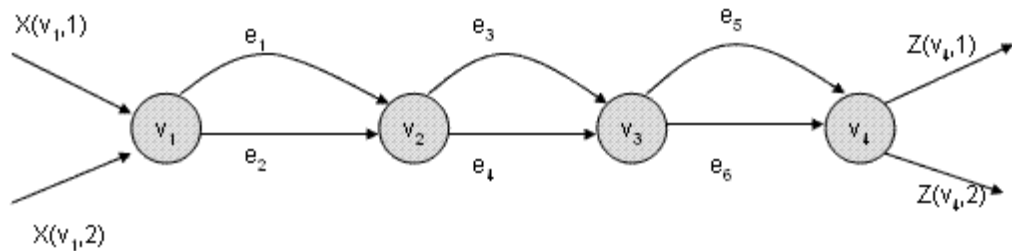


Figure 6: A DAG with sources and a terminal. The input sources to v_1 and the demands of the output processes at v_4 are shown.

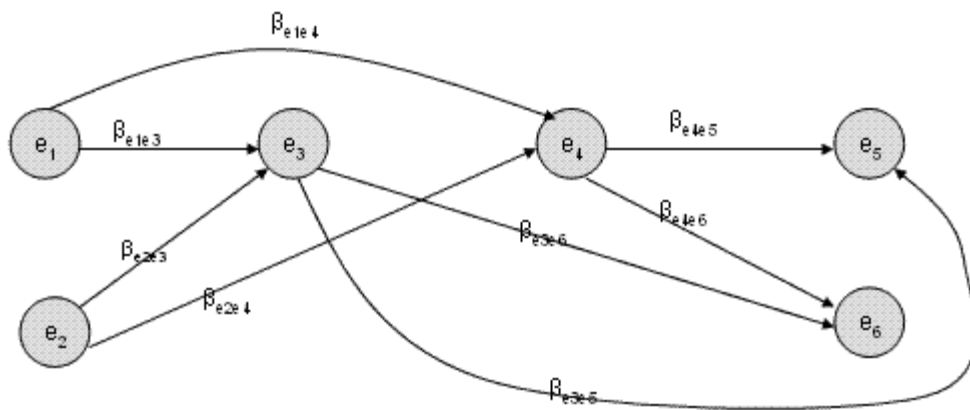


Figure 7: The edge graph corresponding to the graph in Figure 6. Each edge in that graph is represented here by a node. The label β_{e_i,e_j} on the edge between e_i and e_j in this graph shows what fraction of flow on edge e_i goes to edge e_j in the original graph

So, the matrices A and B for the graph in Figure 6 are 2×6 and 2×6 respectively, and the matrix F for the corresponding edge-graph in Figure 7 is 6×6 , and they are as follows:

$$\begin{aligned}
F &= \begin{bmatrix} 0 & 0 & \beta_{e_1,e_3} & \beta_{e_1,e_4} & 0 & 0 \\ 0 & 0 & \beta_{e_2,e_3} & \beta_{e_2,e_4} & 0 & 0 \\ 0 & 0 & 0 & 0 & \beta_{e_3,e_5} & \beta_{e_3,e_6} \\ 0 & 0 & 0 & 0 & \beta_{e_4,e_5} & \beta_{e_4,e_6} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
A &= \begin{bmatrix} \alpha_{1,e_1} & \alpha_{1,e_2} & 0 & 0 & 0 & 0 \\ \alpha_{2,e_1} & \alpha_{2,e_2} & 0 & 0 & 0 & 0 \end{bmatrix} \\
B &= \begin{bmatrix} 0 & 0 & 0 & 0 & \varepsilon_{e_5,1} & \varepsilon_{e_6,1} \\ 0 & 0 & 0 & 0 & \varepsilon_{e_5,2} & \varepsilon_{e_6,2} \end{bmatrix}
\end{aligned}$$