Topic 6: Graphs and Networks

MATH7502 Group Project — fbb0bfdc-79d6-44ad-8100-05067b9a0cf9

Chris Lam *41735613*

Anthony North *46139896*

Stuart Norvill *42938019*

Lee Phillips *43908587*

Minh Tram Julien Tran *44536389*

I. Directed graphs and Incidence matrices

A directed graph consists of a set of edges $(1, \ldots, m)$ and vertices, also known as nodes $(1, \ldots, n)$. The graph can be represented using a $m \times n$ matrix called the incidence matrix, defined by the following rule:

$$A_{ij} = \begin{cases} 1 & \text{if edge i flows to node j} \\ -1 & \text{if edge i flows from node j} \\ 0 & \text{if otherwise} \end{cases} \tag{1}$$

For example, this graph contains 5 edges and 4 nodes, thus creating a 5 by 4 incidence matrix.

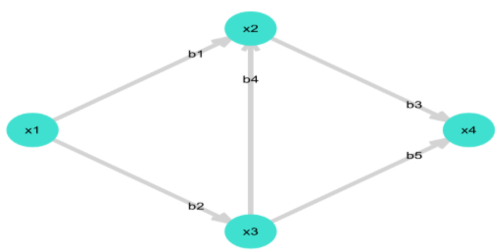

Fig. 1. Directed Graph

$$A = \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \tag{2}$$

II. Adjacency matrix and Degree matrix

A directed graph can also be represented using an adjacency matrix (B) and a degree matrix (D). These two matrices can be linked to the incidence matrix through the use of the operation $A^\top A = D - B$, where $D = \text{diag}(x_1, ..., x_m)$ and D_{ii} is the degree of node i, (i.e. the number of edges interacting with node i); and $Bij = 1$ if there is an edge connecting node i and j.

Using these properties, figure 1 can be represented using the following degree and adjacency matrices:

$$A^\top A = D - B = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \tag{3}$$

It should be noted the concept of direction in the edges is lost by this representation.

III. Flow conservation, Kirchhoffs' Current Law, and Flows with sources

After establishing the incidence matrix, there is a special meaning behind the linear equation $A^\top \vec{x} = \vec{y}$

The vector $\vec{y}$ represents the flow surplus at each node. The vector $\vec{x}$ represents the amount of flow required at each edge to achieve the node surplus.

A more common version of this problem is when $\vec{y} = \vec{0}$. This is known as Kirchhoff's Current Law (KCL). This can be interpreted as making sure the flow into each node is equal to the flow out of each node.

The trick to solving KCL problems is to look for loops within the graph. Below is an explanation for finding three different solutions to the problem $A^\top \vec{x} = \vec{0}$ for figure 1.

There is a loop along edges 1,4, and 2 between nodes 1, 2, and 3. This is denoted by the vector $(1, -1, 0, -1, 0)$. This is a solution to the KCL problem.

There is a second loop along edges 3, 5, and 4 between nodes 2, 4, and 3. This is denoted by the vector $(0, 0, 1, 1, -1)$. This is also a solution to the KCL problem.

The sum of these two solutions produces the vector $(1, -1, 1, 0, -1)$. This is also a solution to the KCL problem. This vector denotes a loop around the outside of the graph.

Finally, you can define a vector $\vec{s} \in \mathbb{R}^n$ and add it to the linear equation.

$$A^\top \vec{x} + \vec{s} = \vec{y} \tag{4}$$

This is useful for many applications where you want to model input into the graph and each node, by assigning values to the vector $\vec{s}$. Each value of s_i within $\vec{s}$ represents flow into node i.

IV. Max Flow and Min Cut Problem

The Max Flow-Min Cut problem is an example of a linear programming problem. The problem is determining the maximum amount of flow able to be sent from a source node to a sink node through a given graph with flow limits applied to each edge. For example, in reference to figure 1, assume the edges each have a flow limit equal to 1. The minimum cut able to be made is a cut partitioning node 1 from the rest of the graph, passing through edges 1 and 2. This minimum cut has a summed flow result of 2, this result is the maximum flow possible for the given graph with the assigned edge limits.