# CS6381 : Lecture # 4 8.22.2020



## Robust Flows.

Consider the scenario where we need to send flows from $s$ to $t$ in a directed graph $G=[V;E]$ with edge capacities. However, some edges/nodes may "fail" and we do not know which will fail. So we send flows Knowing that $k$ edges (or nodes) might fail but not which ones. An adversary then selects the edges that fail Knowing the flow that was chosen. We want to maximize surviving flow; adversary Knows This and so selects so as to inflict maximum damage (and we know this about our adversary)

Q1: How should we send flows so as to max the Surviving flow? Can we maximize the Surviving flow and the initial flow Simultaneously?

Q2: Assuming flow is of information. Is there a way to send each message packet on several edge/node disjoint paths to protect arrival of packet?

We will address these two problems using a third variation of max-flow problem Called Parametrized flows. (2)

Q2: Called Multipath flows by W. Kishimoto et.al
q-path flows

Let $G = [V; E]$ be a directed graph.
$s \in V$, $t \in V$ : origin / destination
$u_{ij} \geq 0$: edge capacity $(i,j) \in E$.
and a positive integer $q$.

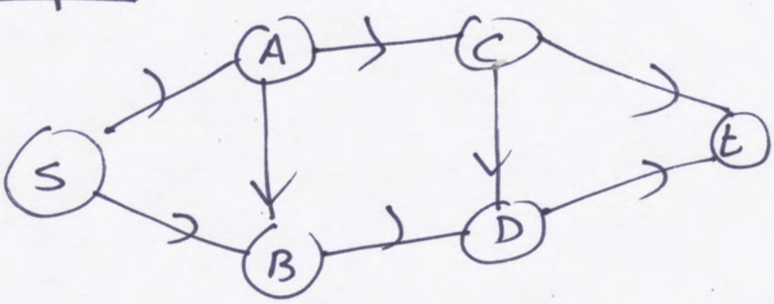
Each "unit" of flow from s-to t must be sent along q disjoint paths (edge disjoints is done here. But node disjointness is also done in W. Kishimoto)
i.e. q copies are being sent from s to t so that even if q-1 edges (nodes) fail, message will reach the destination.
While this is an overkill in many cases, it is useful when extremely important information is sent.
We want to maximize total "distinct" flow.

# Path-Edge (Arc-Chain) Formulation:

We discuss this from now on with $q = 2$; it is <u>Completely</u> illustrative of the general case.

Let $A$ represent the incidence matrix whose rows correspond to edges of $G$ and whose columns correspond to edges of a "$q$-disjoint path" from $s$ to $t$.

## Example



$q = 2$. $\{(s,A), (A,C), (C,t) ; (s,B), (B,D), (D,t)\}$

represents what is called a "double"-path from $s$- to $t$. ($q=2$) One of Column of $A$:

$$
\begin{array}{c c}
 & \text{Path} \\
\begin{bmatrix} sA \\ sB \\ AB \\ AC \\ BD \\ CD \\ Ct \\ Dt \end{bmatrix} &
\begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}
\end{array}
$$

For each such q-path there is a variable representing the q-path flow on that q-path. i.e for each Column of A there is a variable.

The Optimization problem is

$$Ax \leq u$$

$$x \geqq 0 \qquad \underline{e = (1 \cdots \cdots 1)}$$

$$\text{Max} \sum x_j = e^t x$$

[In the "regular" max flow problem, Each Column of A represented one ~~single-path~~ 1-path from s to t]

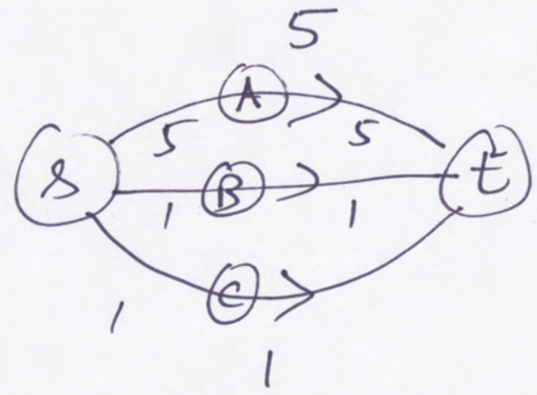
Kishimoto et al gave the "Corresponding" Node edge formulation.

<u>Node-edge-Formulation</u>:

Justification follows.

$$\left\{\begin{array}{l} 0 \leq f_{i,j} \leq u_{i,j} \quad \forall (i,j) \in E \\ \sum_j f_{i,j} - \sum_j f_{j,i} = \begin{cases} F & i = s \\ 0 & i \neq s,t \\ -F & i = t \end{cases} \\ f_{i,j} \leqq \frac{F}{q} \quad \forall (i,j) \in E \\ \text{Max } F \text{ or } \text{Max } F/q \end{array}\right.$$

## Example




$\underline{q = 2}$

Remember each unit of "flow" is <u>along</u> two ($q=2$) disjoint paths (edge disjoint)

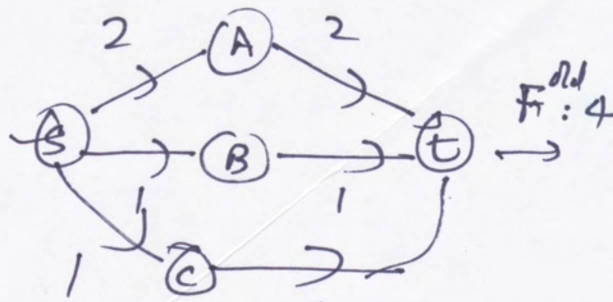
We can <u>not</u> get $7/2$ flow since this can <u>not</u> be decomposed into 2-path flows. The best we can do is

s - A - t
s - B - t } 1

to make up 2 units.

s - A - t
s - C - t } 1

Edge flows for this are




each edge has flow $\leq \dfrac{\text{total (req.) flow}}{2}$.

# Equivalence of two formulations.

Given $x$ in q-Path-edge formulation, define $f$ in Node-edge formulation by the relation

$$f_{i,j} = \sum_{\substack{P: \\ (i,j)\in P}} x_P \quad \forall (i,j)\in E$$

Since $x \geq 0$, $f_{ij} \geq 0 \ \forall (i,j)\in E$.

Since $Ax \leq u$, $f_{ij} \leq u_{ij} \ \forall (i,j)\in E$.

Since $x$ is a path-flow vector, $f_{ij}$ satisfy flow conservation equations.

Let $\frac{F}{q} = \sum_P x_P$

Since $\sum_{P:(i,j)\in P} x_P \leq \sum_P x_P$ (since $x_P \geq 0 \ \forall P$)

$$f_{ij} \leq \frac{F}{q} \quad \forall (i,j)\in E.$$

---

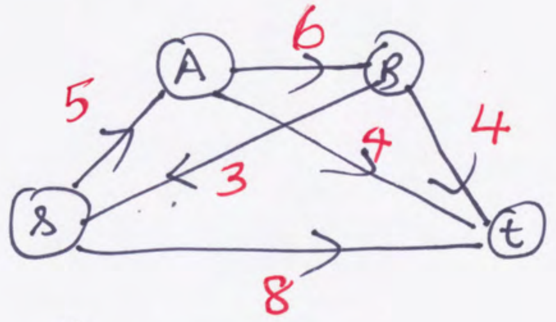
It is the converse that requires some effort.

See next page.

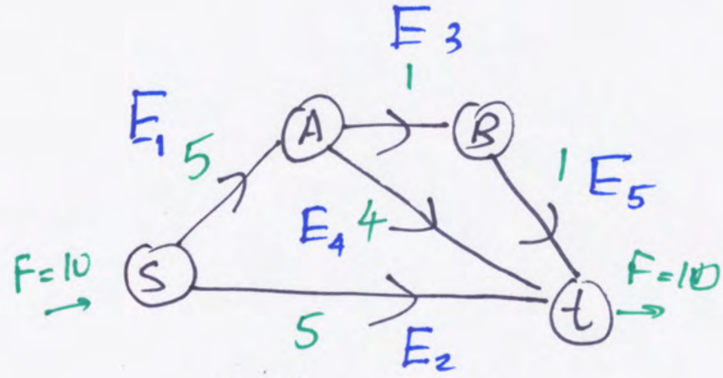
This involves q-path decomposition of flows

It is taken from W. Kishimoto's work. We first illustrate this with an example [This part is the most important aspect of Kishimoto's work.]

Example




$G, u\ ;\ q=2$




$G, f\ :\ f_{ij} > 0$

$$0 \le f_{ij} \le u_{ij} \quad \forall (i,j) \in E$$

$$f_{ij} \le F/2$$

Satisfies conservation

Decomposition into 2-path flows

There are five edges that carry positive flows. & $q=2$. We create a $7 \times 7$ table where there are five rows corresponding to edges $E_1, \ldots E_5$ and similarly 5 columns

These rows: $u_1, u_2, \cdots u_5$ ; $w_1, w_2, \ldots w_5$

2 additional rows: $a_1, a_2$

$q = 2$ additional columns $b_1, b_2$.

(~~13~~ 8)

| | $W_1$ | $W_2$ | $W_3$ | $W_4$ | $W_5$ | $b_1$ | $b_2$ |
|---|---|---|---|---|---|---|---|
| $a_1$ | 5 | | | | | | |
| $a_2$ | | 5 | | | | | |
| $u_1$ | | | 1 | 4 | | | |
| $u_2$ | | | | | | 5 | |
| $u_3$ | | | 4 | | 1 | | |
| $u_4$ | | | | 1 | | | 4 |
| $u_5$ | | | | | 4 | | 1 |

$(a_1, W_1)$ $s \to E_1$ : $5 \leq F/2$

$(a_2, W_2)$ $s \to E_2$ : $5 \leq F/2$

$(u_1, W_3)$ $E_1 \to E_3$ : $1 \leq F/2$

$(u_1, W_4)$ : $E_1 \to E_4$ : $4 \leq F/2$

$(u_2, b_1)$ : $E_2 \to t$ : $5 \leq F/2$

$(u_3, W_3)$ : $E_3 \to E_5$ : $1 \leq F/2$

$(u_4, b_2)$ : $E_4 \to t$ : $4 \leq F/2$

$(u_5, t)$ : $E_5 \to t$ : $1 \leq F/2$

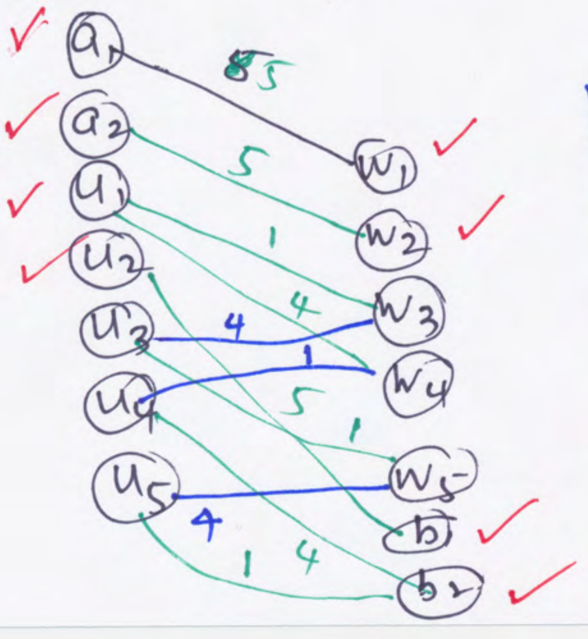
Notice sum of each row and column $\leq 5 = F/2$.

Now we add (fictitious) values at $(u_i, W_i)$ so that total in each row and column equal $F/2 = 5$

These are done in red in the table

(That this always possible is left as an exercise

The above is equivalent to:



In the table after adding "buffers", all totals (of each row & each column) is $5 = F/2$.

There is a result that, in such a table, we can Select a set of positive cells with exactly One in each row & column. One such Set is shown in blue circled entries. We subtract the minimum of these from each of them → this set will correspond to a 2-path flow. (will show this ~~soon~~ next)

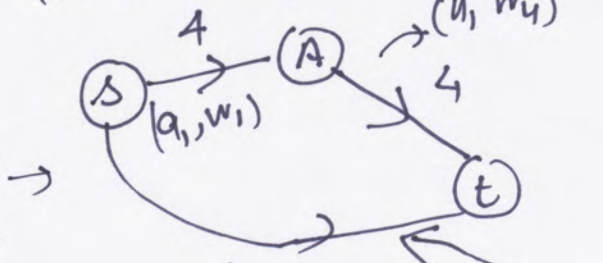
Min Value = 4.

| | $w_1$ | $w_2$ | $w_3$ | $w_4$ | $w_5$ | $b_1$ | $b_2$ |
|---|---|---|---|---|---|---|---|
| $a_1$ | 4 | | | | | | |
| $a_2$ | | 4 | | | | | |
| $u_1$ | | | | 4 | | | |
| $u_2$ | | | | | | 4 | |
| $u_3$ | | | 4 | | | | |
| $u_4$ | | | | | | | 4 |
| $w_5$ | | | | | 4 | | |

$(u_3, w_3), (u_5, w_5)$ : buffers Can be ignored.

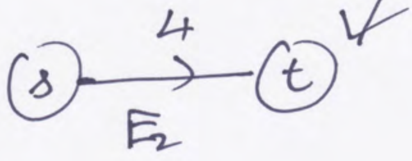
$\overset{s}{\cong} a_1, w_1 ; u_1, w_4 ; u_4\ b_2 \overset{=}{} t$

$(s\text{-}A)$ $A, E_4$, $E_4 \to t$



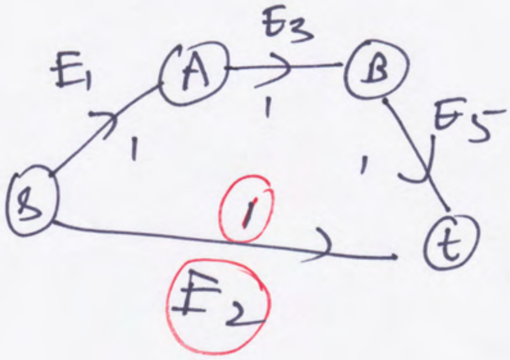

$\overset{s}{\cong}(a_2 \cdot w_2) ; (w_2, b_1) : 4$

$s \to E_2$ $E_2 \to t$



The remaining table.

| | $w_1$ | $w_2$ | $w_3$ | $w_4$ | $w_5$ | $b_1$ | $b_2$ |
|---|---|---|---|---|---|---|---|
| $a_1$ | 1 | | | | | | |
| $a_2$ | | 1 | | | | | |
| $u_1$ | | | 1 | | | | |
| $u_2$ | | | | | | 1 | |
| $u_3$ | | | | | 1 | | |
| $u_4$ | | | | 1 | | | |
| $u_5$ | | | | | | | 1 |




$(u_4, w_4)$ : buffer

---

<u>Theorem</u> : We can find such elements one in each row and one in each column

<u>Pf</u>: Recall row and column totals are same for each row and each column & these two are also same. Hence if we have a ~~b~~ bipartite graph as shown before and each node on the left has an input $=1$, and each node on the right has an output $=1$, we get

$$\sum_{j=1}^{7} x_{i,j} = 1 \qquad i = 1 \cdots 7$$

$$\sum_{i=1}^{7} x_{ij} = 1 \qquad j = 1 \cdots 7$$

$x_{ij} = 0$ if cell has 0 value

---

This system has a solution;

$$x_{ij} = \frac{\text{Cell value}}{5} \rightarrow \text{total value.}$$

Hence it has an integer solution.
(Think max flow = 7 in the network)
∴ There is an integral flow → which is
0/1 → 1's are the cells we want.

---

This completes ~~a~~ q-path decomposition

If $q = 3$, we have $a_1, a_2, a_3, b_1, b_2, b_3$

and so on; $F/q = F/3$. etc.