# COL702: Advanced Data Structures and Algorithms

Ragesh Jaiswal, CSE, IITD

# Applications of Network Flow

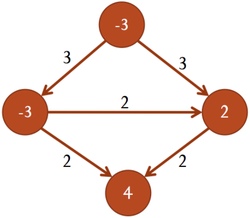
- Given a weighted directed graph representing a transportation network.
- There are multiple supply nodes in the graph denoting the places that has a factory for some product.
- There are multiple demand nodes denoting the consumption points.
- Each supply node $v$ has an associated supply value $s(v)$ denoting the amount the product it can supply.
- Each demand node $v$ has a similar demand value $d(v)$.
- Question: Is there a way to ship product such that all demand and supply goals are met?

### Problem

Given a directed graph $G$ with integer edge capacities. For each node $v$, there is an associated demand value $t(v)$ denoting the demand at the node (*for supply nodes this is* $-s(v)$, *for demand nodes* $d(v)$, *for other nodes* 0). Find whether there exists a flow $f$ such that for all nodes $v$:

$$f^{in}(v) - f^{out}(v) = t(v)$$

and the capacity constraints are met. Such a flow is called a feasible circulation.



## Problem

Given a directed graph $G$ with integer edge capacities. For each node $v$, there is an associated demand value $t(v)$ denoting the demand at the node (*for supply nodes this is* $-s(v)$*, for demand nodes* $d(v)$*, for other nodes* 0). Find whether there exists a flow $f$ such that for all nodes $v$:

$$f^{in}(v) - f^{out}(v) = t(v)$$

and the capacity constraints are met. Such a flow is called a feasible circulation.

- Claim 1: For a feasible circulation to exist, $\sum_v t(v) = 0$. (*That means supply equals the demand*)

- Claim 1: For a feasible circulation to exist, $\sum_v t(v) = 0$. (*That means supply equals the demand*)
- Consider the flow network as shown in the diagram below and let $D = \sum_{\text{demand node } v} d(v)$.

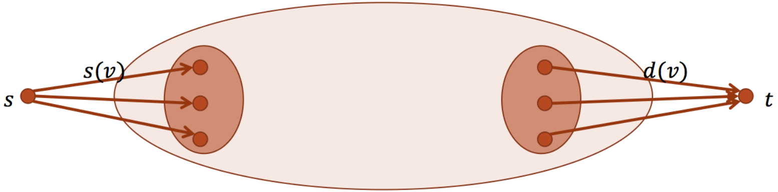



Figure: Connect source to supply nodes and demand nodes to sink.

- Claim 1: For a feasible circulation to exist, $\sum_v t(v) = 0$. (*That means supply equals the demand*)
- Consider the flow network $G'$ as shown in the diagram below and let $D = \sum_{\text{demand node } v} d(v)$.
- Claim 2: There is a feasible circulation in $G$ iff the maximum flow in the network $G'$ is $D$.

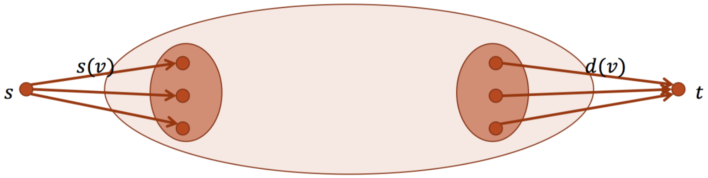



Figure: Connect source to supply nodes and demand nodes to sink.

- Claim 1: For a feasible circulation to exist, $\sum_v t(v) = 0$. (*That means supply equals the demand*)
- Consider the flow network $G'$ as shown in the diagram below and let $D = \sum_{\text{demand node } v} d(v)$.
- Claim 2: There is a feasible circulation in $G$ iff the maximum flow in the network $G'$ is $D$.
  - (if) Consider the max-flow and remove $s, t$.
  - (only if) Extend the feasible circulation in the network.

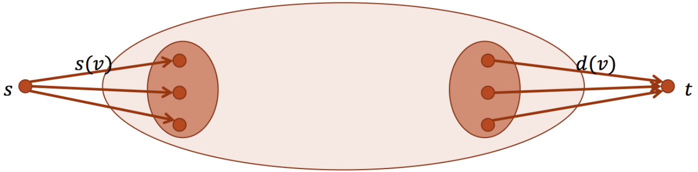



Figure: Connect source to supply nodes and demand nodes to sink.

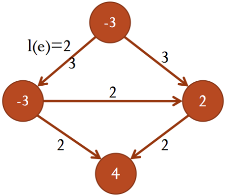
## Problem

Given a directed graph $G$ with integer edge capacities $c(e)$ and lower bounds $l(e)$. For each node $v$, there is an associated demand value $t(v)$ denoting the demand at the node (*for supply nodes this is* $-s(v)$, *for demand nodes* $d(v)$, *for other nodes* 0). Find whether there exists a flow $f$ such that for all nodes $v$:

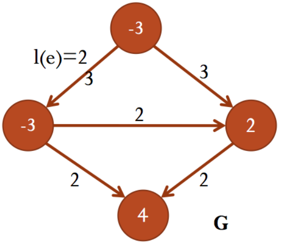
$$f^{in}(v) - f^{out}(v) = t(v)$$

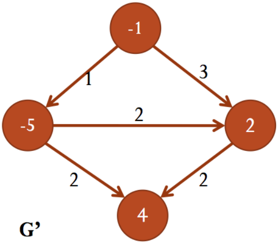
and the following capacity constraints are met. For every edge $e$:

$$l(e) \leq f(e) \leq c(e)$$



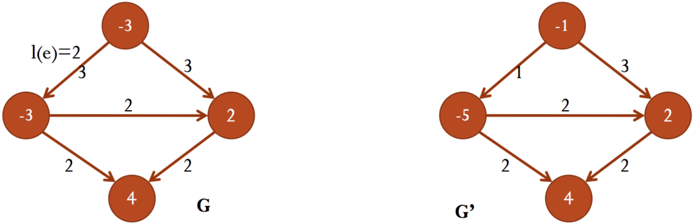
- Consider a flow $f$ such that for all edge $e$, $f(e) = l(e)$.
- For each vertex $v$, let $r(v) = f^{in}(v) - f^{out}(v)$.
- Construct a new graph $G'$:
  - Each edge $e$ in $G'$ has capacity $c(e) - l(e)$.
  - Each vertex $v$ in $G'$ has a demand $t(v) - r(v)$.
- Idea: Solve the feasible circulation problem without lower bounds on $G'$.



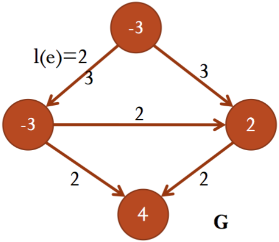



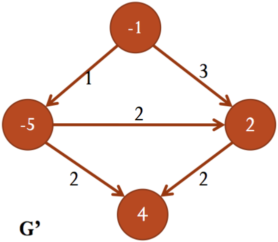
# Network Flow
## Feasible Circulation with Lower Bounds

- Consider a flow $f$ such that for all edge $e$, $f(e) = l(e)$.
- For each vertex $v$, let $r(v) = f^{in}(v) - f^{out}(v)$.
- Construct a new graph $G'$:
  - Each edge $e$ in $G'$ has capacity $c(e) - l(e)$.
  - Each vertex $v$ in $G'$ has a demand $t(v) - r(v)$.
- Idea: Solve the feasible circulation problem without lower bounds on $G'$.
- Claim: There is a feasible circulation (with lower bounds) in $G$ iff there is a feasible circulation in $G'$.



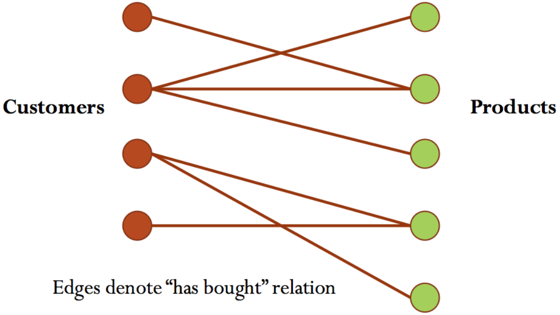
- Claim: There is a feasible circulation (with lower bounds) in $G$ iff there is a feasible circulation in $G'$.
  - (**if**) Let $f'$ be a feasible circulation in $G'$. Consider $f$ where $f(e) = f'(e) + l(e)$. Is $f$ a feasible circulation in $G$?
  - (**only if**) Let $f$ be a feasible circulation in $G$. Consider $f'$ where $f'(e) = f(e) - l(e)$. Is $f'$ a feasible circulation in $G'$?



## Problem

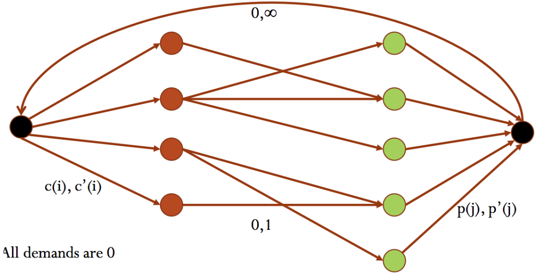
There are $n$ customers and $m$ products. Each customer $i$ is supposed to review between $c(i)$ and $c'(i)$ products that he has bought in the past and each product $j$ should be reviewed by between $p(j)$ and $p'(j)$ customers. Find a way to do the survey.



**Problem**

There are $n$ customers and $m$ products. Each customer $i$ is supposed to review between $c(i)$ and $c'(i)$ products that he has bought in the past and each product $j$ should be reviewed by between $p(j)$ and $p'(j)$ customers. Find a way to do the survey.

- Consider the flow network set up below.
- <u>Claim</u>: The survey is feasible iff there is a feasible circulation (with lower bounds) in the network.



### Definition (Edge-disjoint path)

Two paths $P_1$ and $P_2$ between from vertex $s$ to vertex $t$ are called edge-disjoint if $P_1$ and $P_2$ do not share any edges.

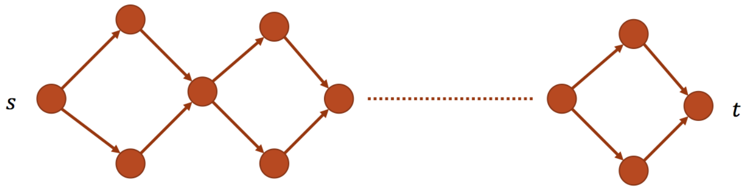
### Problem

Given an unweighted directed graph $G$, find the maximum number of edge-disjoint paths between $s$ and $t$ in $G$.

**Problem**

Given an unweighted directed graph $G$, find the maximum number of edge-disjoint paths between $s$ and $t$ in $G$.

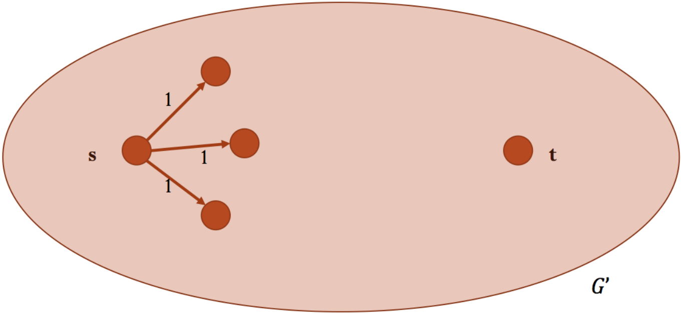
- How many paths from $s$ to $t$ are present in this graph?
- How many edge-disjoint paths from $s$ to $t$ are present in this graph?



## Problem

Given an unweighted directed graph $G$, find the maximum number of edge-disjoint paths between $s$ and $t$ in $G$.

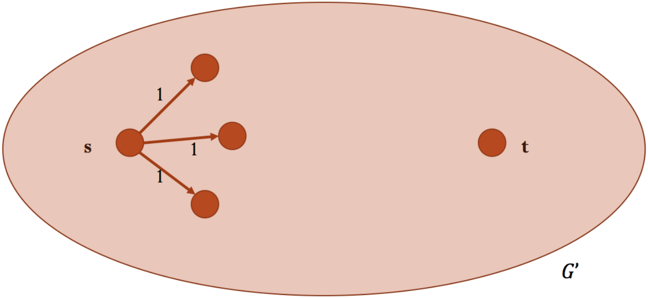
- Consider the network $G'$ below:



## Problem

Given an unweighted directed graph $G$, find the maximum number of edge-disjoint paths between $s$ and $t$ in $G$.

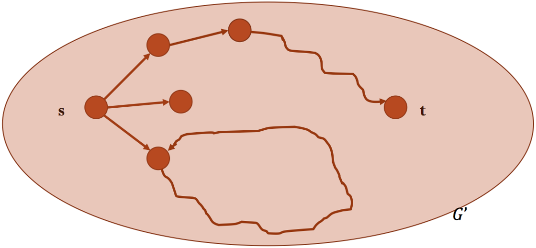
- Claim 1: If there are $k$ edge-disjoint paths in $G$, then there is an $s$-$t$ flow in the network with value at least $k$.



**Problem**

Given an unweighted directed graph $G$, find the maximum number of edge-disjoint paths between $s$ and $t$ in $G$.

- Claim 1: If there are $k$ edge-disjoint paths in $G$, then there is an $s$-$t$ flow in the network with value at least $k$.
- Claim 2: If there is an $s$-$t$ flow in $G'$ of value $k$, then there are at least $k$ edge-disjoint paths in $G$.
  - *Idea*: Use induction on the number of edges with non-zero flow value.



- You are given an image as a 2-D matrix of pixels.
- We want to determine the foreground and the background pixels.
- Each pixel $i$, has an integer $a(i)$ associated with it denoting how likely it is to be a foreground pixel.
- Similarly, each pixel $i$, has an integer $b(i)$ associated with it denoting how likely it is to be a foreground pixel.
- For neighboring pixels, $i$ and $j$, there is an associated penalty $p(i,j)$ with putting $i$ and $j$ in different sets.

**Problem**

Find a partition of the pixels into $F$ and $B$ such that:

$$\sum_{i \in F} a(i) + \sum_{i \in B} b(i) - \sum_{i \text{ and } j \text{ are neighbors but in different sets}} p(i,j)$$

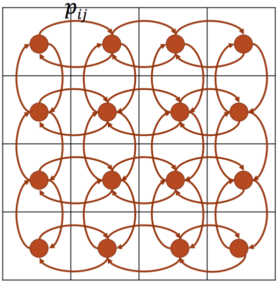
is maximized.

### Problem

Find a partition of the pixels into $F$ and $B$ such that:

$$\sum_{i \in F} a(i) + \sum_{i \in B} b(i) - \sum_{i \text{ and } j \text{ are neighbors but in different sets}} p(i,j)$$

is maximized.



**Problem**

Find a partition of the pixels into $F$ and $B$ such that:

$$\sum_{i \in F} a(i) + \sum_{i \in B} b(i) - \sum_{i \text{ and } j \text{ are neighbors but in different sets}} p(i,j)$$

is maximized.

- Consider the network below:

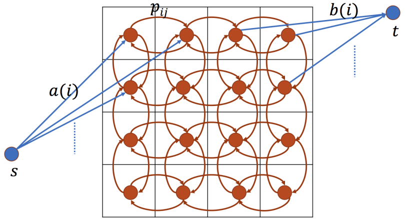



Figure: Idea: The $s$-$t$ min-cut in the above network gives the optimal partition.

- Let $C = \sum_i a(i) + \sum_i b(i)$.
- Claim 1: Consider a partition $(F, B)$ of the set of pixels. Let $S = F \cup \{s\}$, $T = B \cup \{t\}$. Then the capacity of the $s$-$t$ cut $(S, T)$ in the network is given by

$$C(S,T) = C - \left( \sum_{i \in F} a(i) + \sum_{i \in B} b(j) - \sum_{i \text{ and } j \text{ are neighbors but in different sets}} p(i,j) \right)$$

- Let $C = \sum_i a(i) + \sum_i b(i)$.
- Claim 1: Consider a partition $(F, B)$ of the set of pixels. Let $S = F \cup \{s\}$, $T = B \cup \{t\}$. Then the capacity of the $s$-$t$ cut $(S, T)$ in the network is given by

$$C(S,T) = C - \left( \sum_{i \in F} a(i) + \sum_{i \in B} b(j) - \sum_{i \text{ and } j \text{ are neighbors but in different sets}} p(i,j) \right)$$

- Claim 2: Consider an $s$-$t$ cut $(S, T)$ in the network. Let $F = A \setminus \{s\}$, $B = T \setminus \{t\}$. Then

$$C(S,T) = C - \left( \sum_{i \in F} a(i) + \sum_{i \in B} b(j) - \sum_{i \text{ and } j \text{ are neighbors but in different sets}} p(i,j) \right)$$

- Let $C = \sum_i a(i) + \sum_i b(i)$.
- Claim 1: Consider a partition $(F, B)$ of the set of pixels. Let $S = F \cup \{s\}$, $T = B \cup \{t\}$. Then the capacity of the $s$-$t$ cut $(S, T)$ in the network is given by

$$C(S,T) = C - \left( \sum_{i \in F} a(i) + \sum_{i \in B} b(j) - \sum_{i \text{ and } j \text{ are neighbors but in different sets}} p(i,j) \right)$$

- Claim 2: Consider an $s$-$t$ cut $(S, T)$ in the network. Let $F = A \setminus \{s\}$, $B = T \setminus \{t\}$. Then

$$C(S,T) = C - \left( \sum_{i \in F} a(i) + \sum_{i \in B} b(j) - \sum_{i \text{ and } j \text{ are neighbors but in different sets}} p(i,j) \right)$$

- Form Claims 1 and 2, we get that if $(S, T)$ is a $s$-$t$ min-cut in the network, then $F = S \setminus \{s\}, B = T \setminus \{t\}$ is an optimal solution to the Image Segmentation problem

End