CSL 356: Analysis and Design of Algorithms

Ragesh Jaiswal

CSE, IIT Delhi

Network Flow: Applications

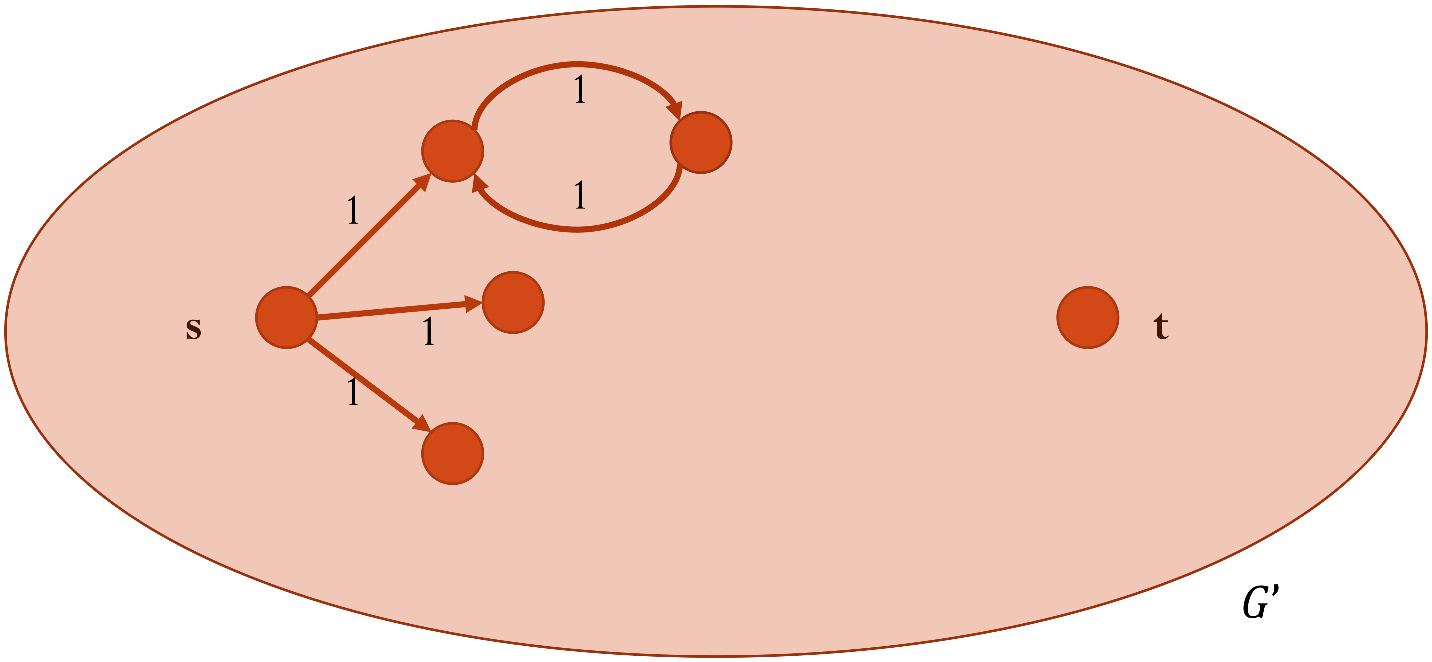
Edge disjoint paths in undirected graphs

Network Flow: Applications

- Problem: Given an *undirected* graph G find the maximum number of *edge disjoint paths* between s and t in G.

Network Flow: Applications

- Problem: Given an *undirected* graph G find the maximum number of *edge disjoint paths* between s and t in G.



- Idea: Add bi-directional edges and consider the max flow that has at least one flow value 0 between any pair of nodes.

Network Flow: Applications

Image Segmentation

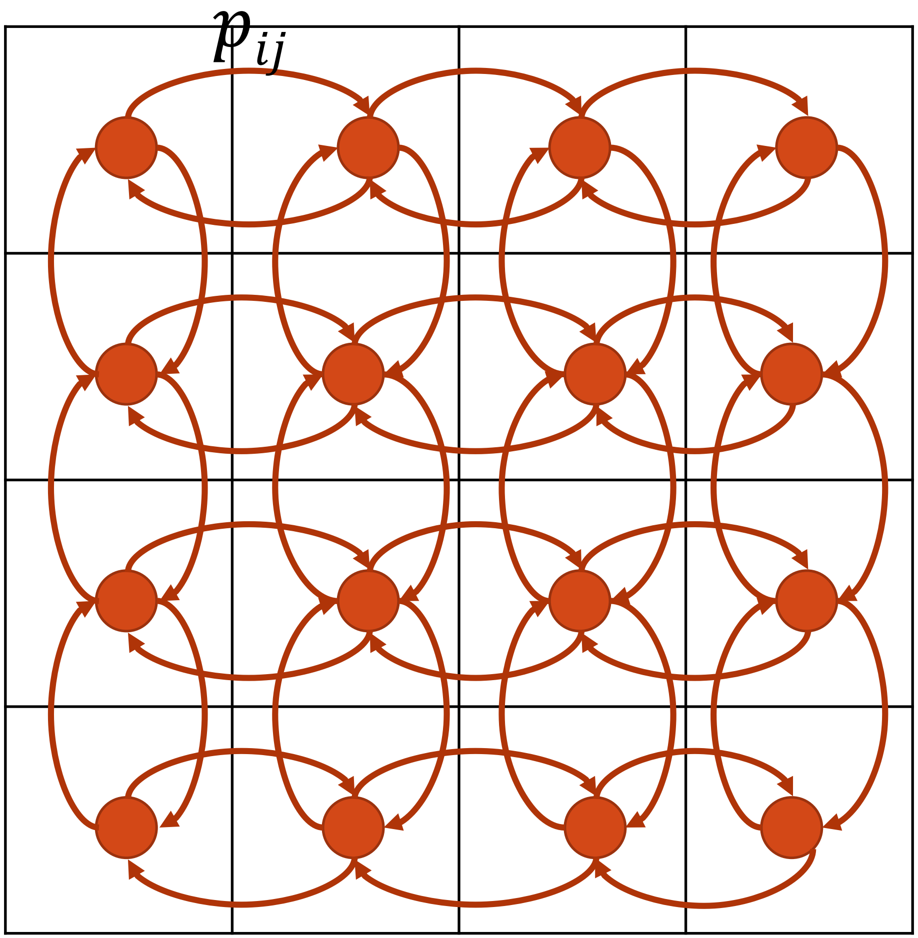
Network Flow: Applications

- 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.
- Each pixel i, has an integer $b(i)$ associated with it denoting how likely it is to be a background pixel.
- For neighboring pixels, i and j, there is an associated penalty $p(i,j)$ with putting i and j in different sets.
- Find a partition of the pixels into F and B such that:

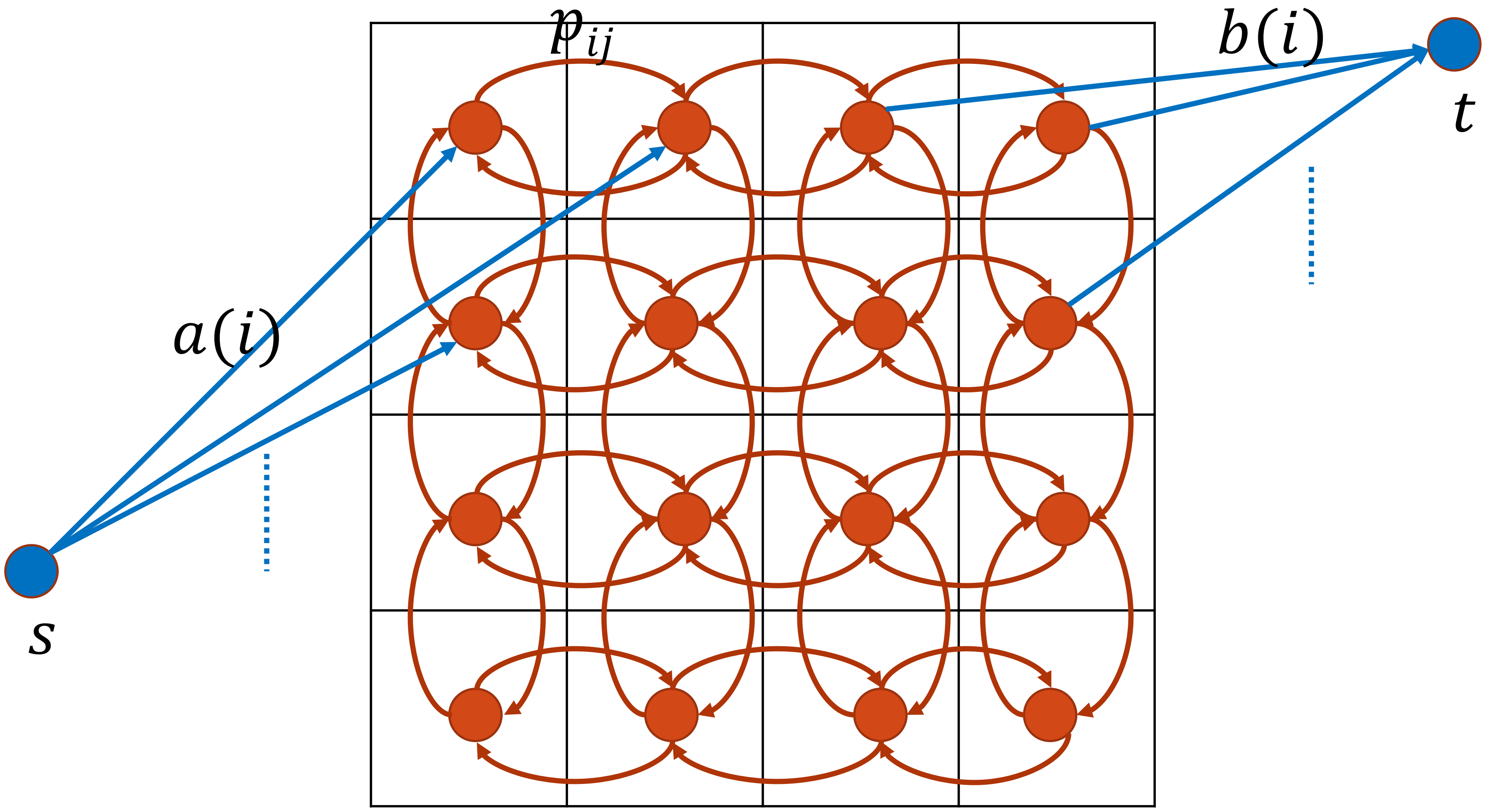
$$\sum_{i \in F} a(i) + \sum_{i \in B} b(i) - \sum_{i,j \text{ are neighbors and in different sets}} p(i,j)$$

is maximized.

Network Flow: Applications



Network Flow: Applications



Idea: Find an s-t min-cut in the above flow network. This gives an optimal partition of pixels.

Network Flow: Applications

- Let $C = \sum a(i) + \sum b(i)$.
- 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 flow network is given by

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

Network Flow: Applications

- Let $C = \sum a(i) + \sum b(i)$.
- 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 flow network is given by

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

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

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

Network Flow: Applications

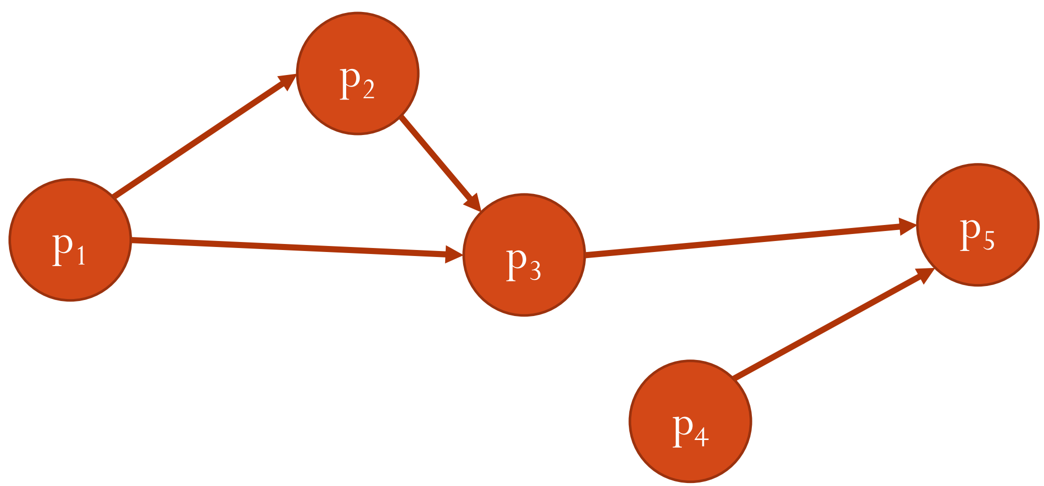
- Let $C = \sum a(i) + \sum b(i)$.
- Claim 1: Consider a partition (F, B) of the set of pixels. Let $S = F \cup \{s\}, T = B \cup \{t\}$. Then the capacity of the $s - t$ cut (S, T) in the flow network is given by
$$c(S,T) = C - \left(\sum_{i \in F} a(i) + \sum_{j \in B} b(j) - \sum_{i,j \text{ are neighbors and in different sets}} p(i,j) \right)$$
- Claim 2: Consider an $s - t$ cut (S, T) in the flow network. Let $F = A \setminus \{s\}, B = T \setminus \{t\}$. Then
$$c(S,T) = C - \left(\sum_{i \in F} a(i) + \sum_{i \in B} b(i) - \sum_{i,j \text{ are neighbors in different sets}} p(i,j) \right)$$
- Theorem: If (S, T) be an $s - t$ min-cut in the flow network, then $F = S \setminus \{s\}, B = T \setminus \{t\}$ is an optimal solution to the Image Segmentation problem.

Network Flow: Applications

Project Selection

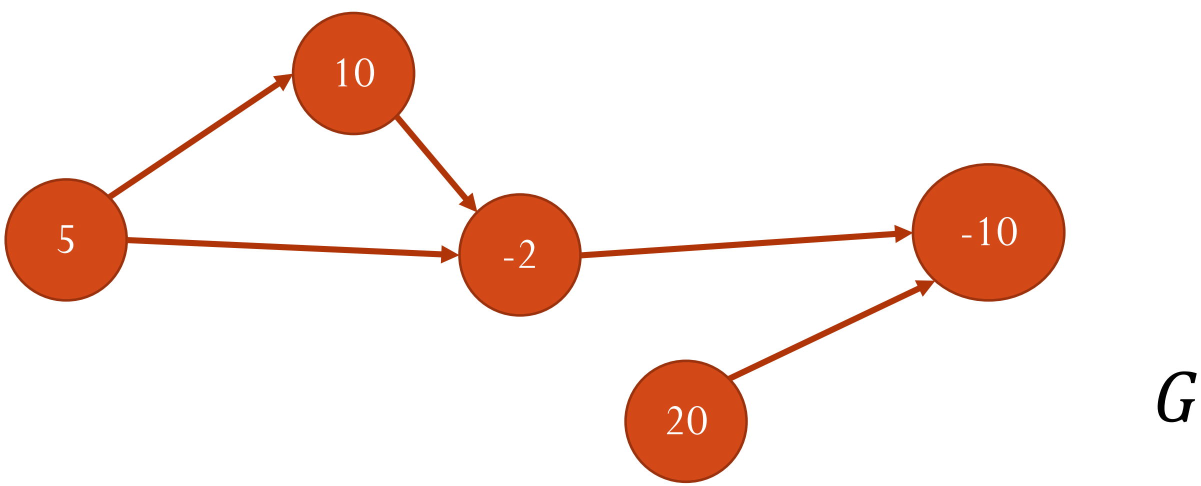
Network Flow: Applications

- Problem: There are n projects, each associated with a profit $p(i)$ (this could be positive or negative integer). There are dependencies between projects. These dependencies are stored using a dependency graph G where there is a directed edge from project i to project j if project i depends on project j. Find a *feasible subset* A of projects such that $\sum_{i \in A} p(i)$ is maximized.
  - *Feasible subset*: A subset A is called feasible if for any edge (i, j), if i is in A, then j is in A.



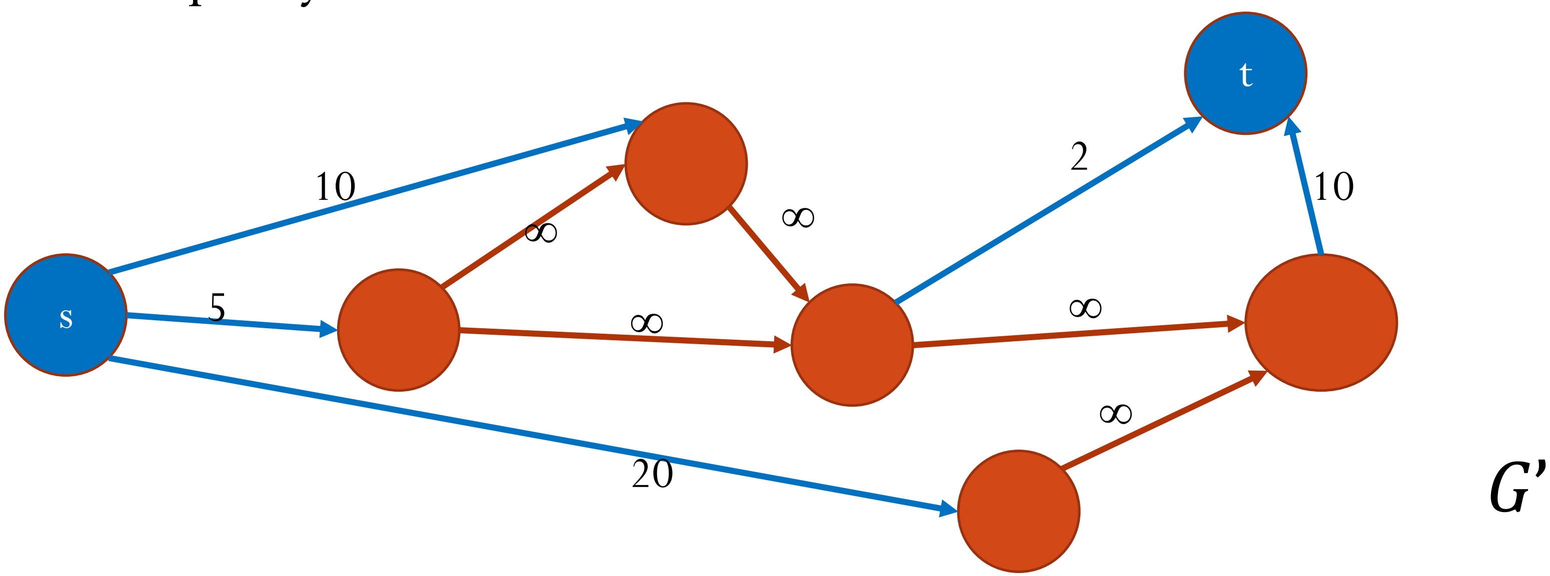
Network Flow: Applications

- Consider the following network flow G'
  - Add a source s and a sink node t.
  - For all i such that $p(i) > 0$, there is an edge (s, i) in G' with capacity $p(i)$.
  - For all i such that $p(i) \leq 0$, there is an edge (i, t) in G' with capacity $-p(i)$.
  - For all edges (i, j) in G, there is an edge (i, j) in G' with capacity ∞.



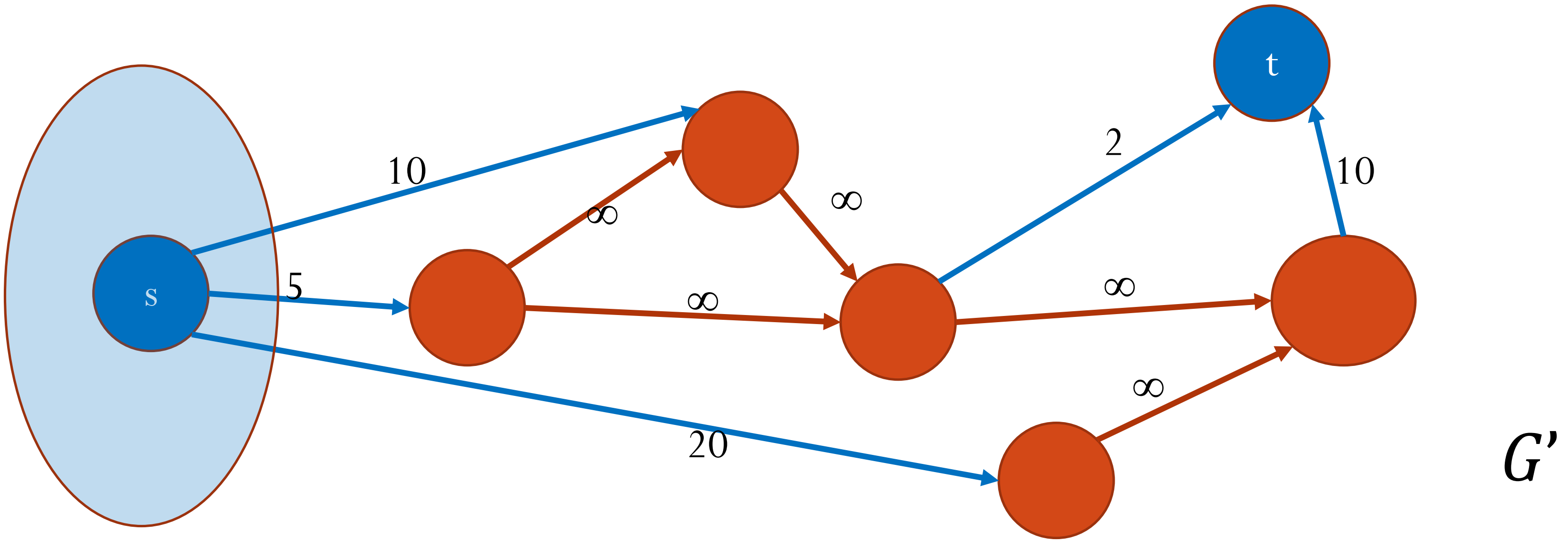
Network Flow: Applications

- Consider the following network flow G'
  - Add a source s and a sink node t.
  - For all i such that $p(i) > 0$, there is an edge (s, i) in G' with capacity $p(i)$.
  - For all i such that $p(i) \leq 0$, there is an edge (i, t) in G' with capacity $-p(i)$.
  - For all edges (i, j) in G, there is an edge (i, j) in G' with capacity ∞.



Network Flow: Applications

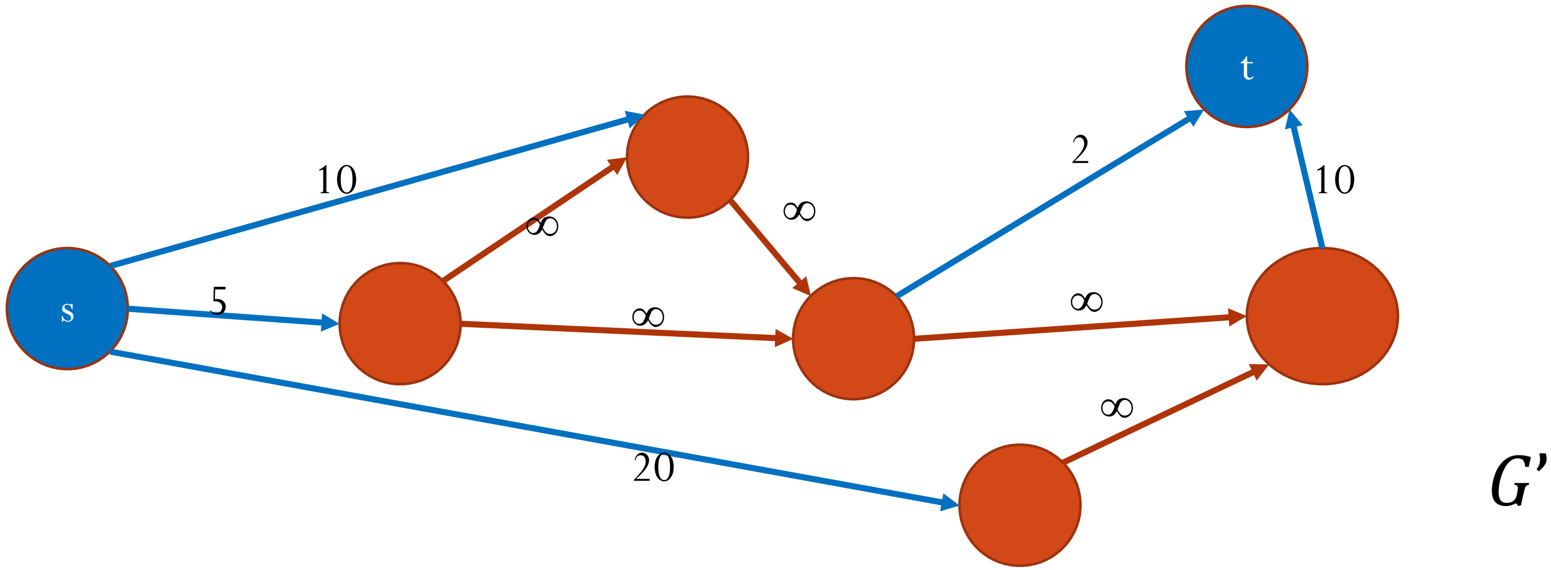
- Let $C = \sum_{i, p(i)>0} p(i)$



Network Flow: Applications

- Let $C = \sum_{i, p(i)>0} p(i)$

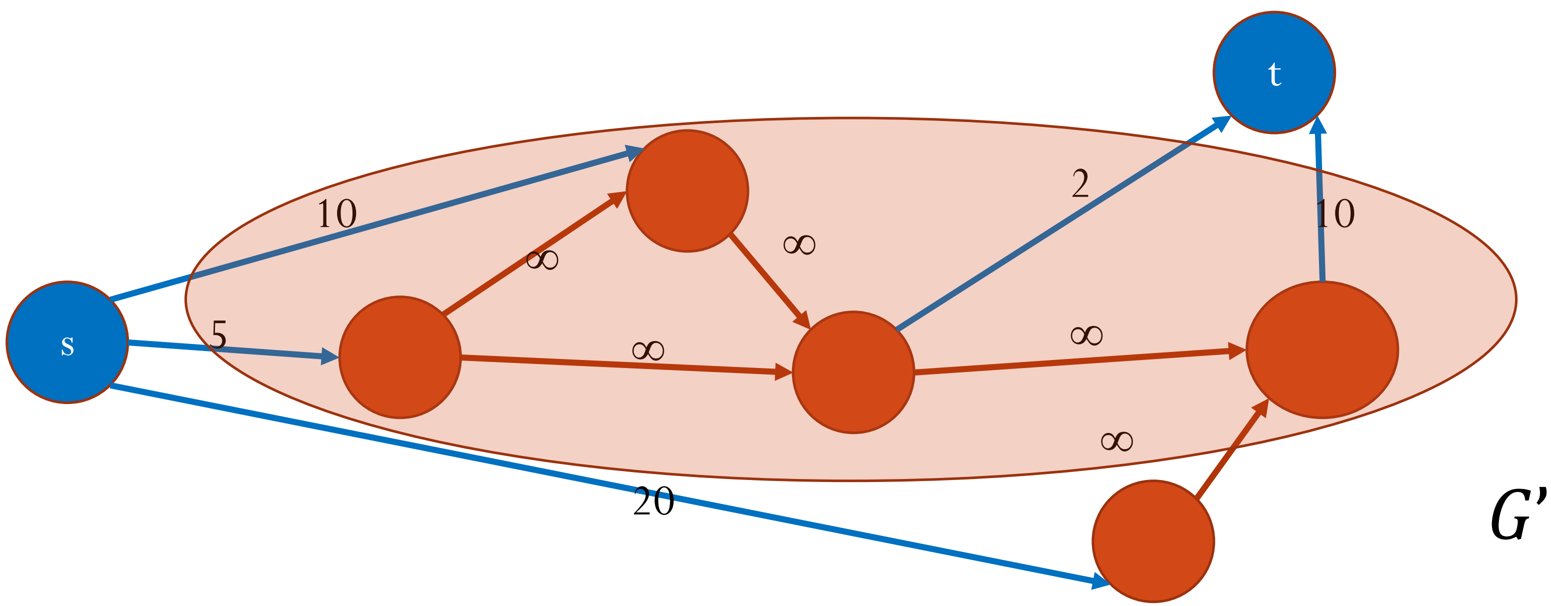
- <u>Claim 1</u>: For any feasible subset A, there is an $s - t$ cut (A', B') in G' such that $c(A', B') = C - \sum_{i \in A} p(i)$.



Network Flow: Applications

- Let $C = \sum_{i, p(i)>0} p(i)$

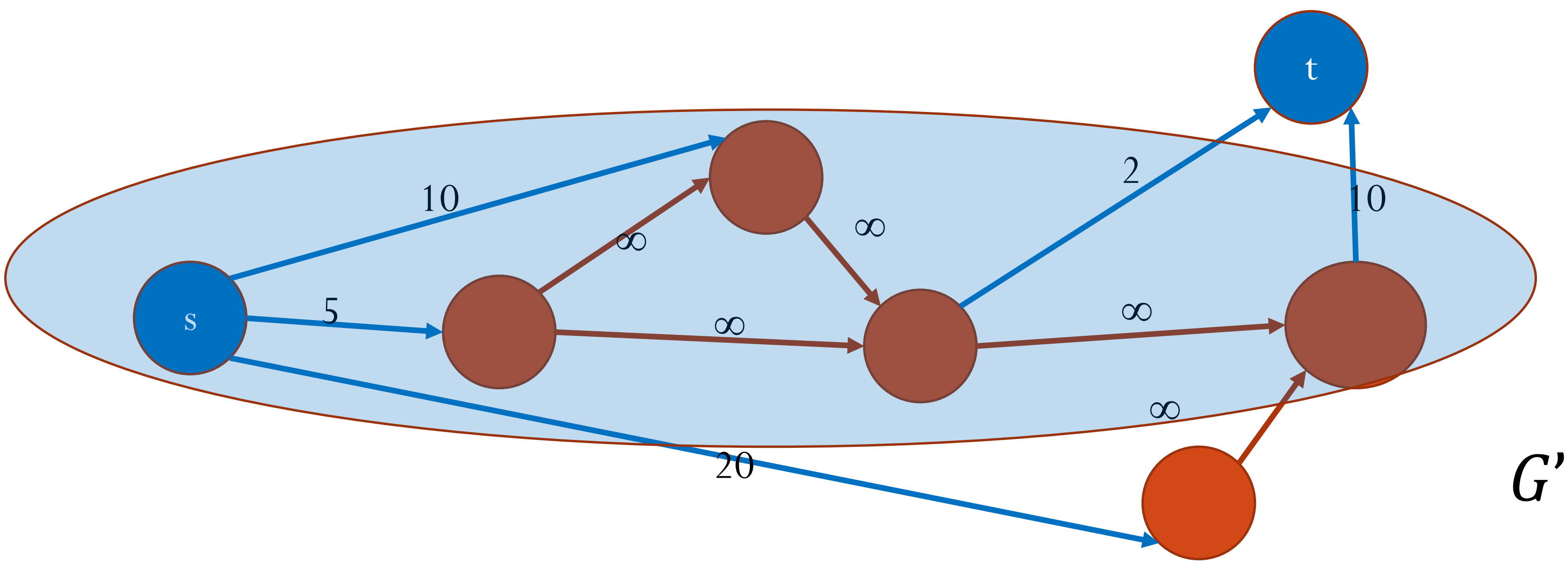
- <u>Claim 1</u>: For any feasible subset A, there is an $s - t$ cut (A', B') in G' such that $c(A', B') = C - \sum_{i \in A} p(i)$.
  - Proof: Consider $A' = A \cup \{s\}$.



Network Flow: Applications

- Let $C = \sum_{i, p(i)>0} p(i)$

- <u>Claim 1</u>: For any feasible subset A, there is an $s - t$ cut (A', B') in G' such that $c(A', B') = C - \sum_{i \in A} p(i)$.
  - Proof: Consider $A' = A \cup \{s\}$.



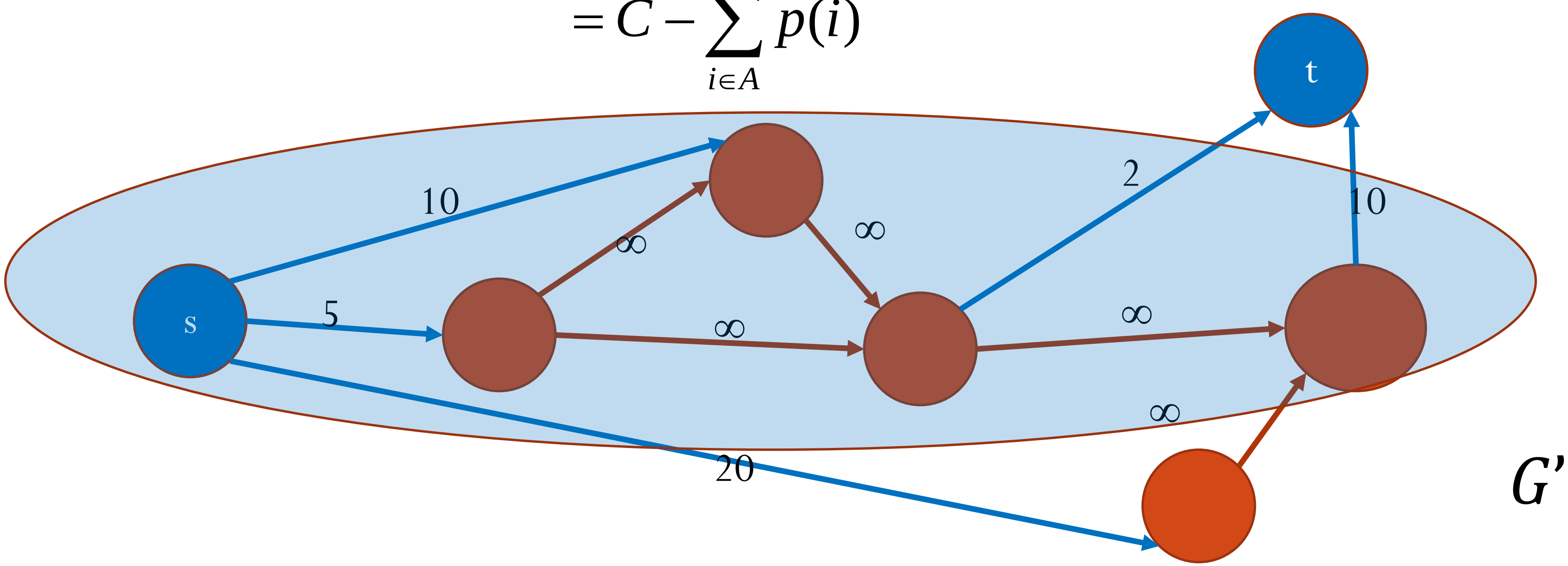
Network Flow: Applications

- Claim 1: For any feasible subset A, there is an $s - t$ cut (A', B') in G' such that $c(A', B') = C - \sum_{i \in A} p(i)$.
  - Proof: Consider $A' = A \cup \{s\}$. We have:

$$c(A', B') = \sum_{i \notin A, p(i)>0} p(i) - \sum_{i \in A, p(i) \leq 0} p(i)$$

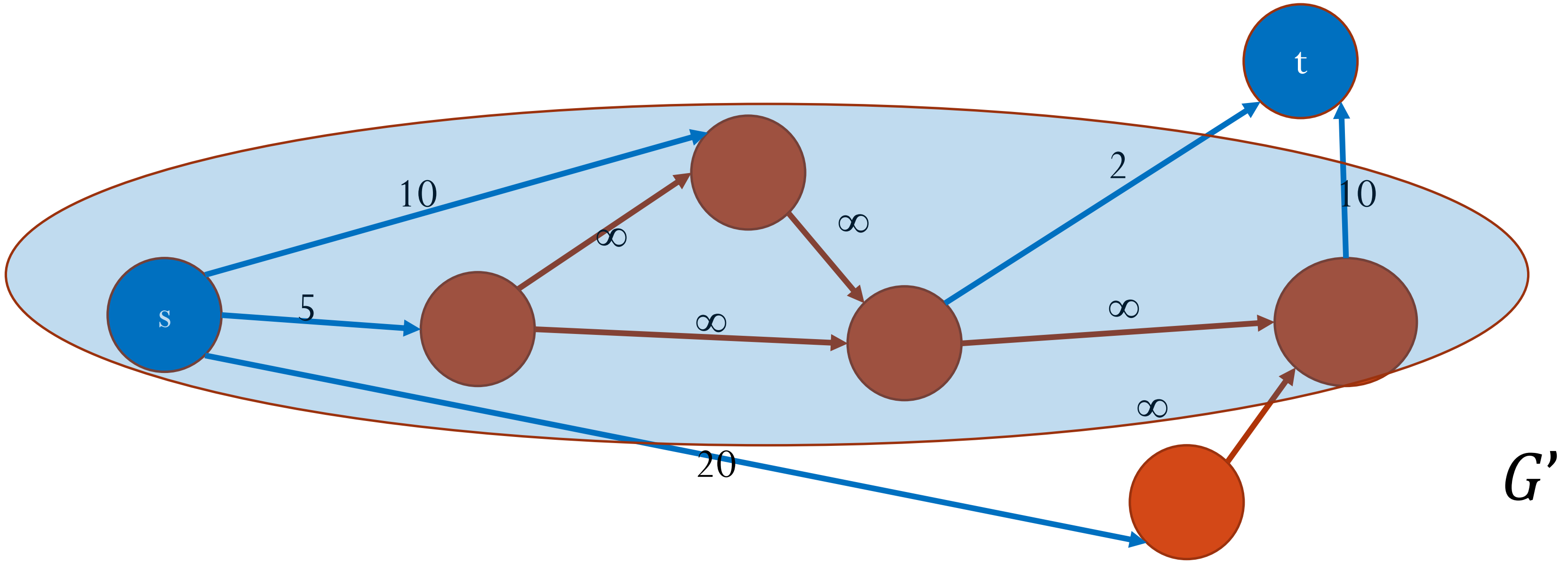
$$= (C - \sum_{i \in A, p(i)>0} p(i)) - \sum_{i \in A, p(i) \leq 0} p(i)$$

$$= C - \sum_{i \in A} p(i)$$



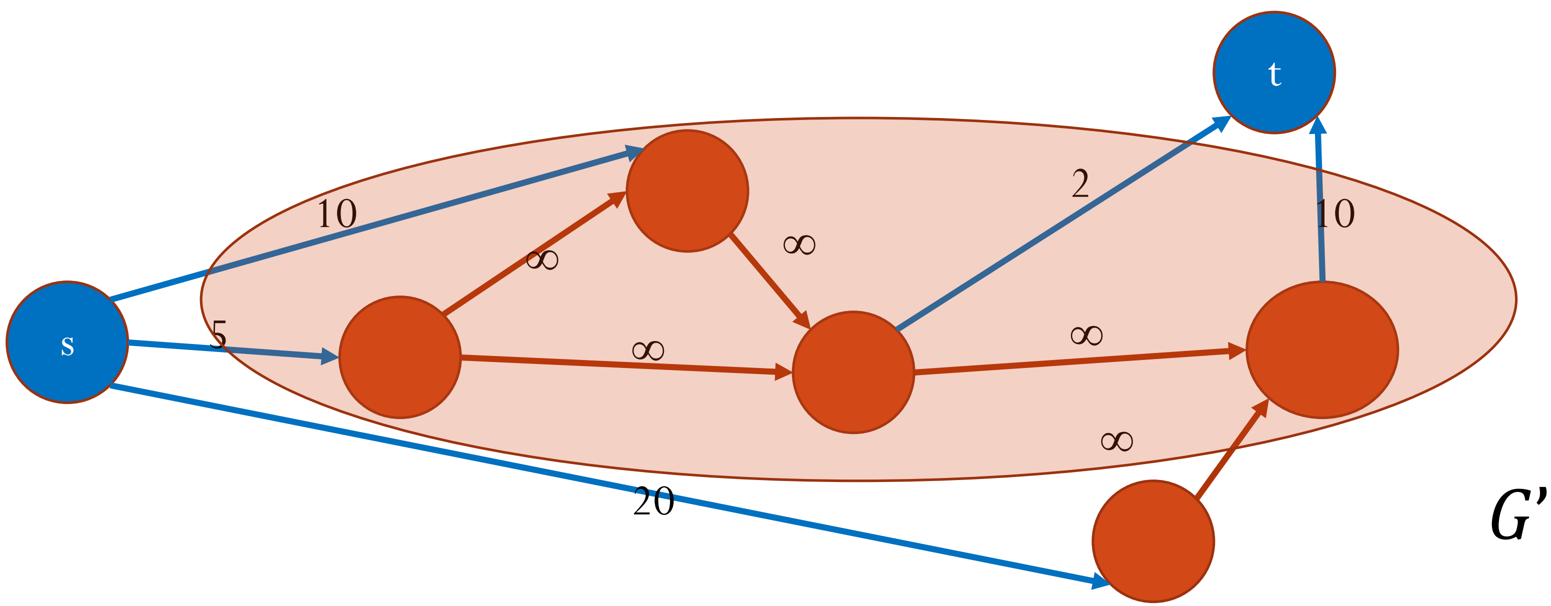
Network Flow: Applications

- <u>Claim 2</u>: For any $s - t$ cut (A', B') in G' of capacity at most C, $A = A' \setminus \{s\}$ is a feasible subset.



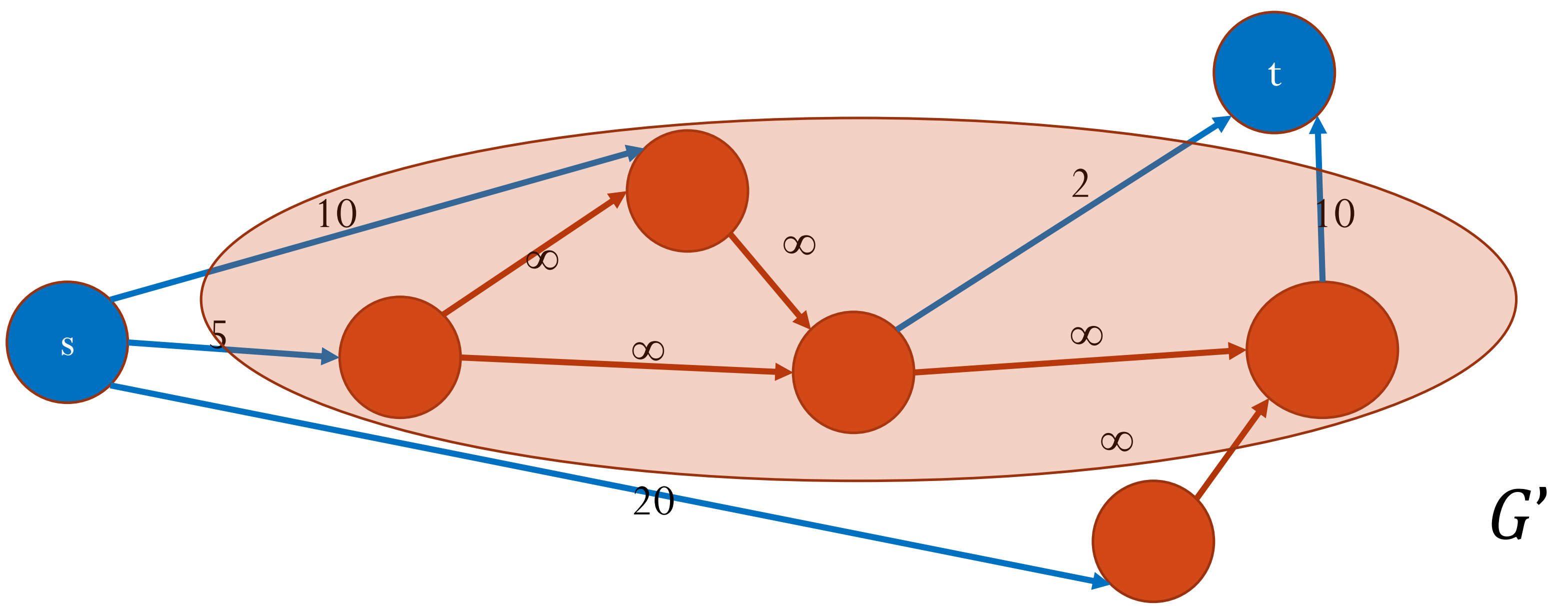
Network Flow: Applications

- Claim 2: For any $s - t$ cut (A', B') in G' of capacity at most C, $A = A' \setminus \{s\}$ is a feasible subset.
  - Proof: Since all edges (i, j) corresponding to G are have capacity ∞.



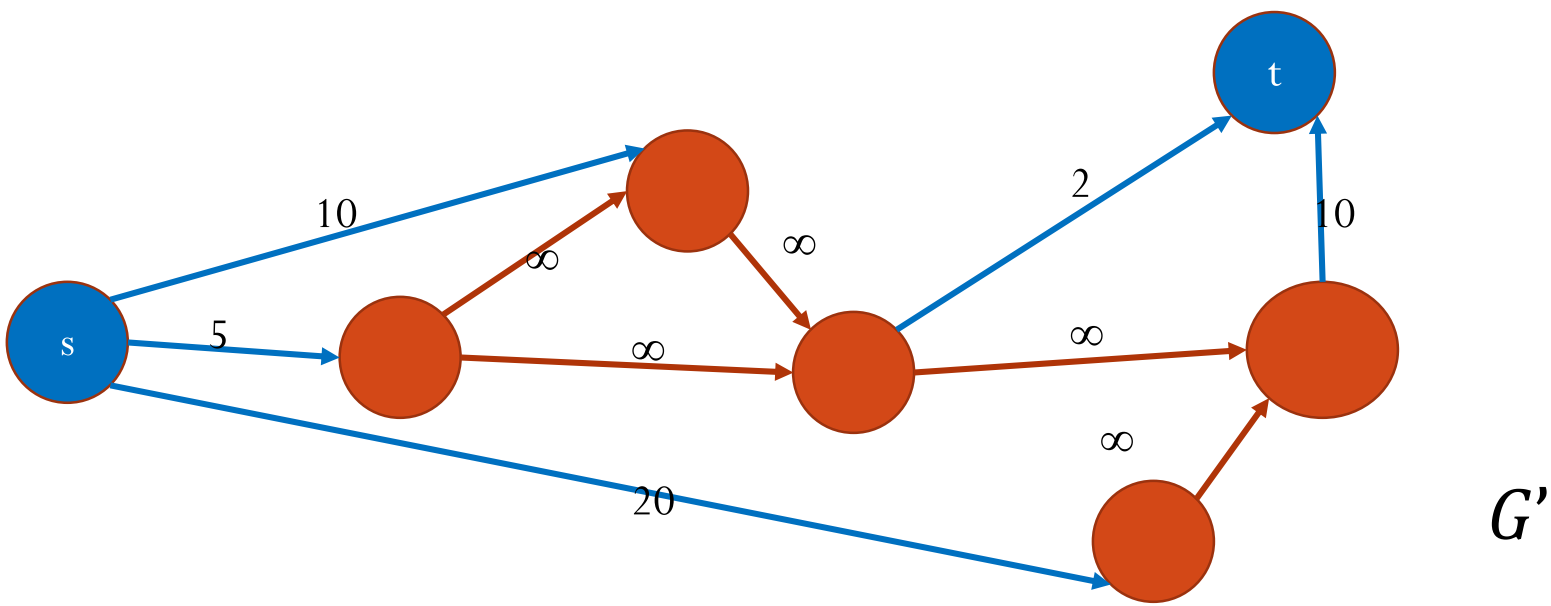
Network Flow: Applications

- <u>Claim 2</u>: For any $s - t$ cut (A', B') in G' of capacity at most C, $A = A' \setminus \{s\}$ is a feasible subset. Moreover $c(A', B') = C - \sum_{i \in A} p(i)$.



Network Flow: Applications

- Theorem: If (A', B') is the minimum cut in G', then $A' - \{s\}$ is an optimal solution to the project selection problem.



End