COL351: Analysis and Design of Algorithms

Ragesh Jaiswal, CSE, IITD

Applications of Network Flow

- 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:

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

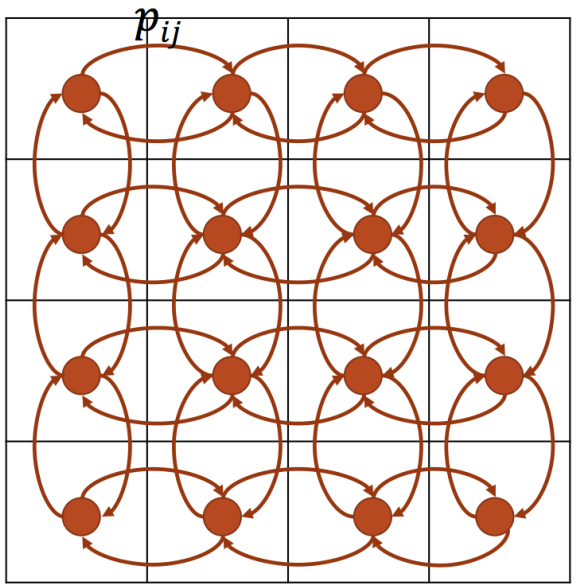
is maximized.

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:

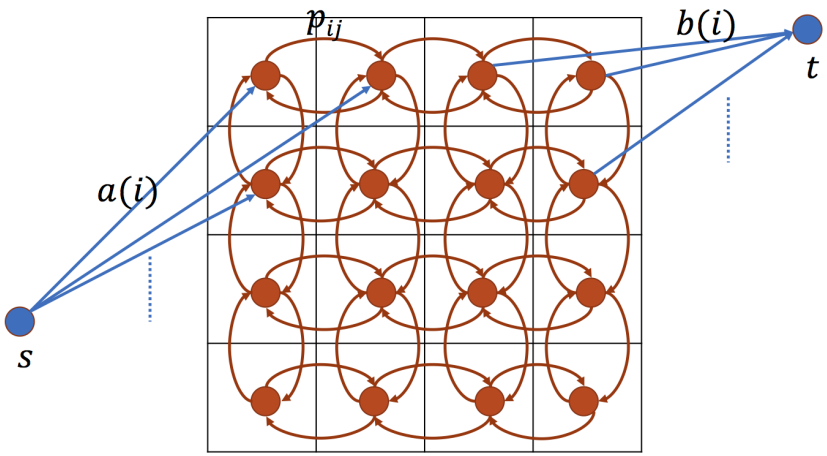


Figure: Idea: The s-t min-cut in the above network gives the optimal partition.

- 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

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 iff 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 iff for any edge (i, j), $i \in A \Rightarrow j \in A$.

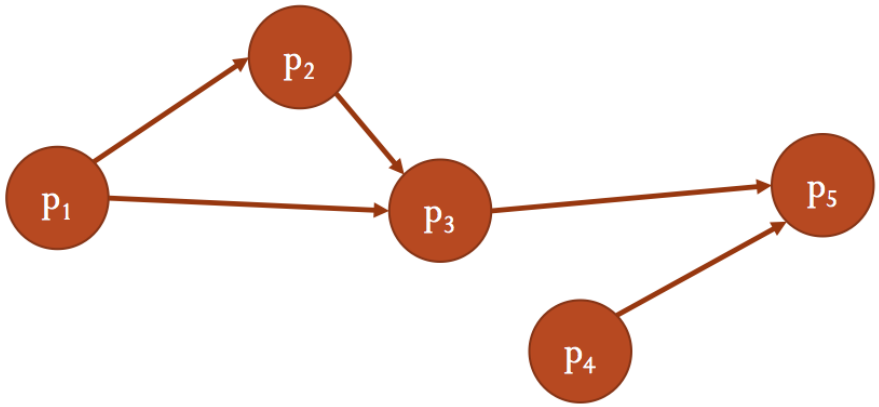


Figure: An example with dependencies.

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 iff 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 iff for any edge (i, j), $i \in A \Rightarrow j \in A$.

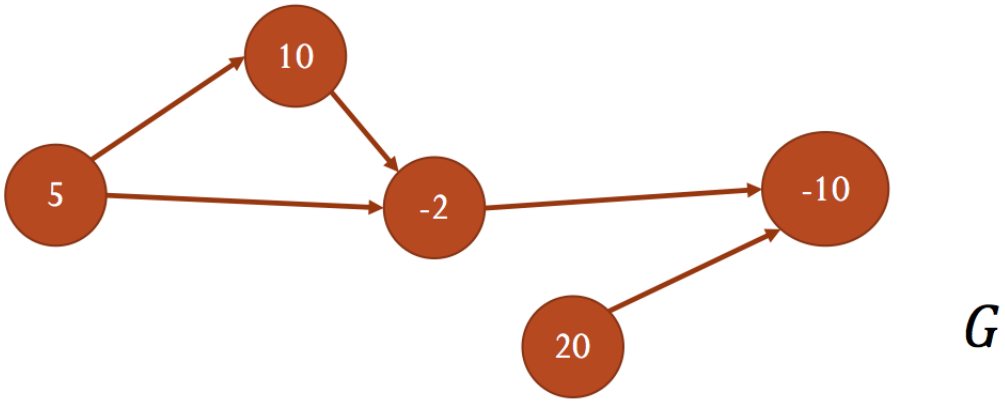


Figure: An example with dependencies and profits.

- Consider the following flow network G':
  - Add a source s and a sink 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 ∞.

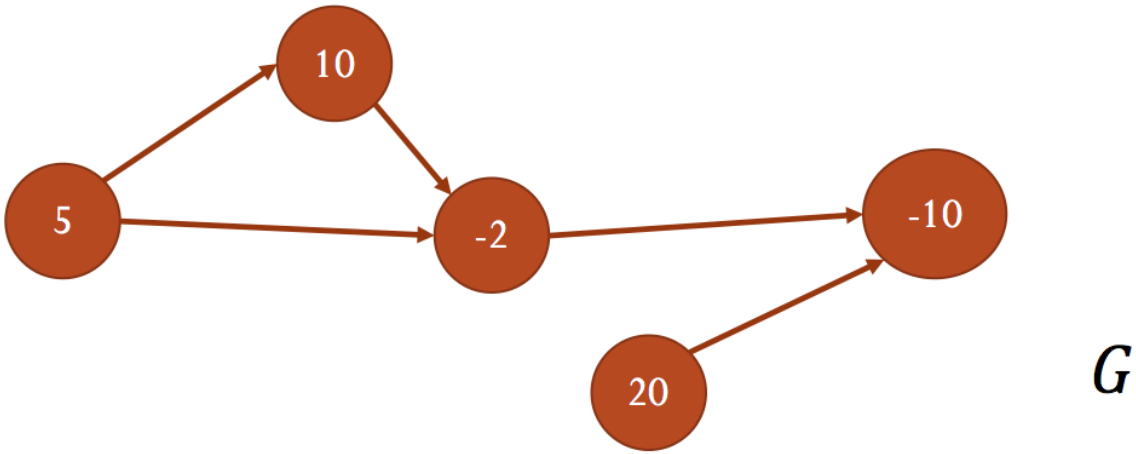
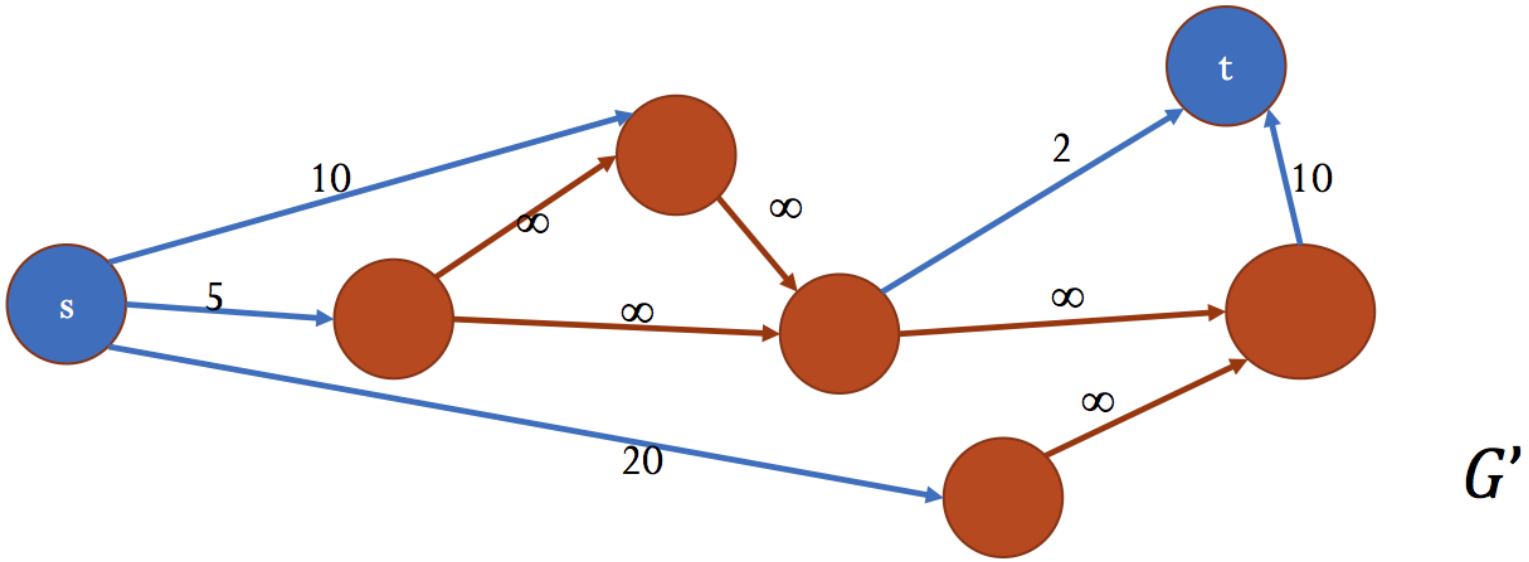
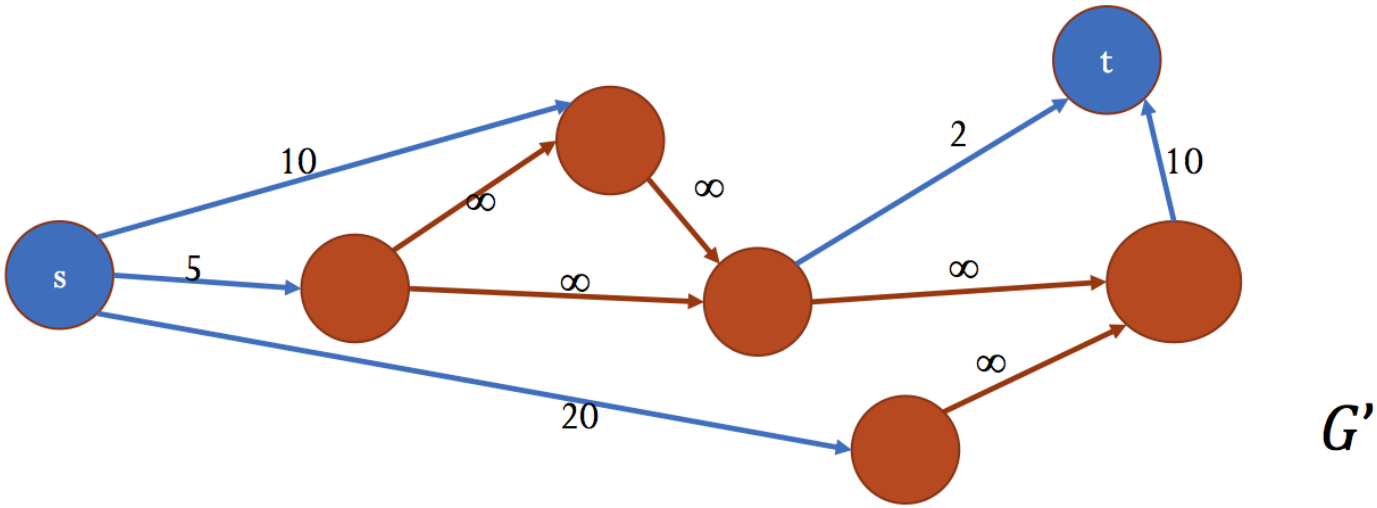


Figure: An example with dependencies and profits.

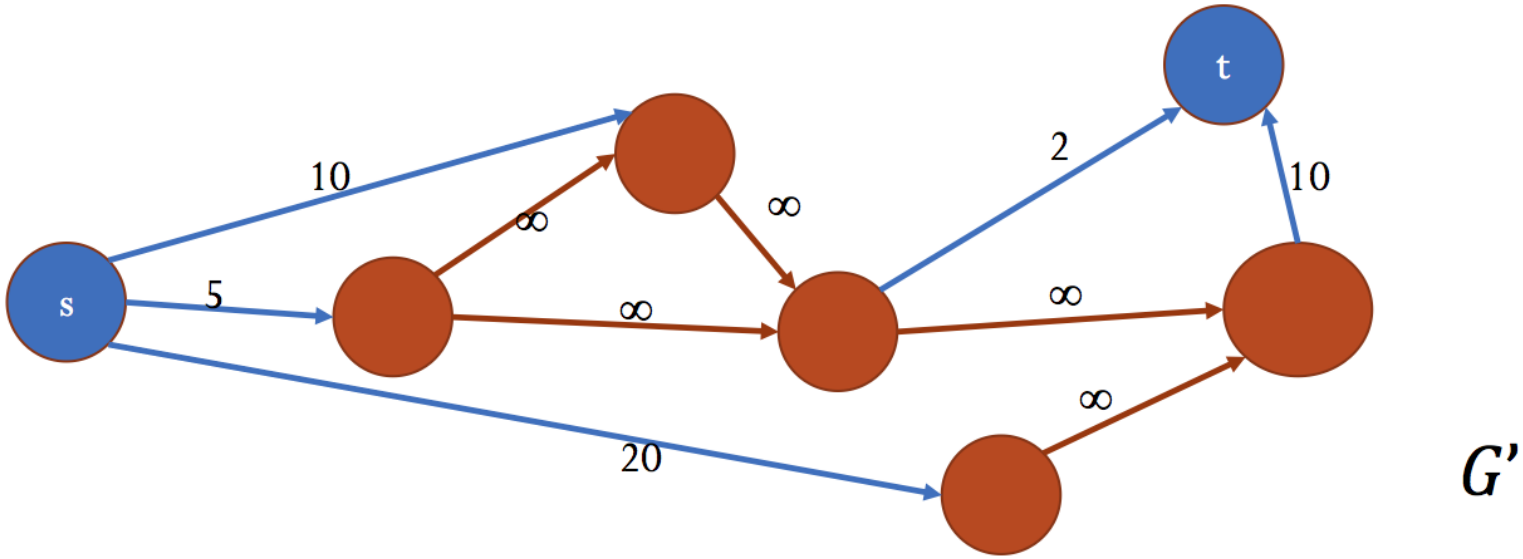
- Consider the following flow network G':
  - Add a source s and a sink t.
  - For all i such that $p(i) > 0$, there is an edge (s, i) in G' with capacity $p(i)$.
  - For all i such that $p(i) \leq 0$, there is an edge (i, t) in G' with capacity $-p(i)$.
  - For all edges $(i, j) \in G$, there is an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



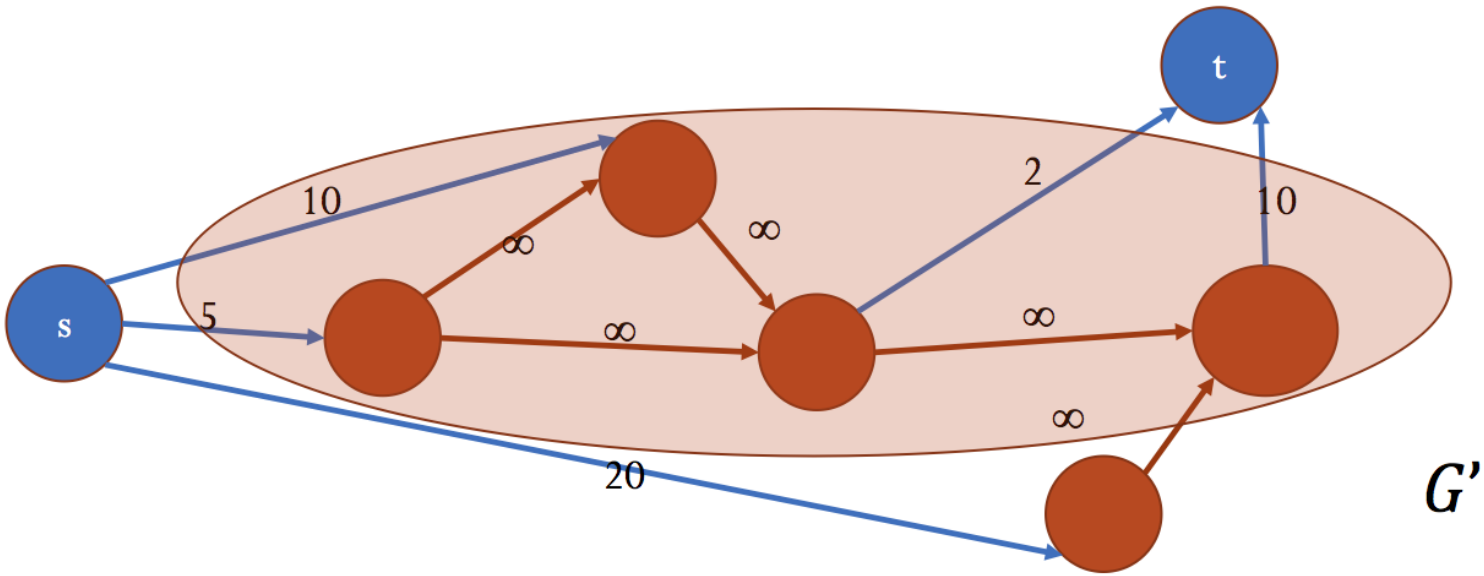
- Let $C = \sum_{i \text{ s.t. } p(i)>0} p(i)$.
- Claim 1: For any feasible subset A, there is an s-t cut (A', B') in G' such that $c(A', B') = C - \sum_{i \in A} p(i)$.



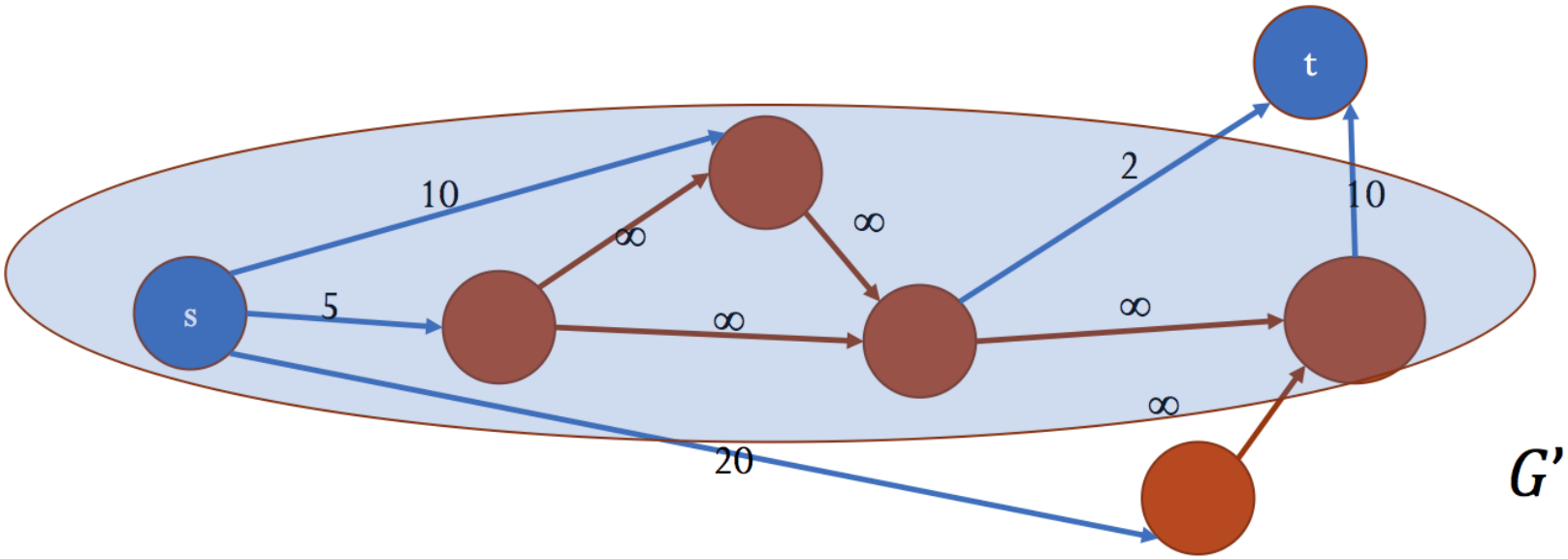
- Let $C = \sum_{i \text{ s.t. } p(i)>0} p(i)$.
- Claim 1: For any feasible subset A, there is an s-t cut (A', B') in G' such that $c(A', B') = C - \sum_{i \in A} p(i)$.
  - Proof sketch: Consider $A' = A \cup \{s\}$.



- Let $C = \sum_{i \text{ s.t. } p(i)>0} p(i)$.
- 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 sketch: Consider $A' = A \cup \{s\}$.

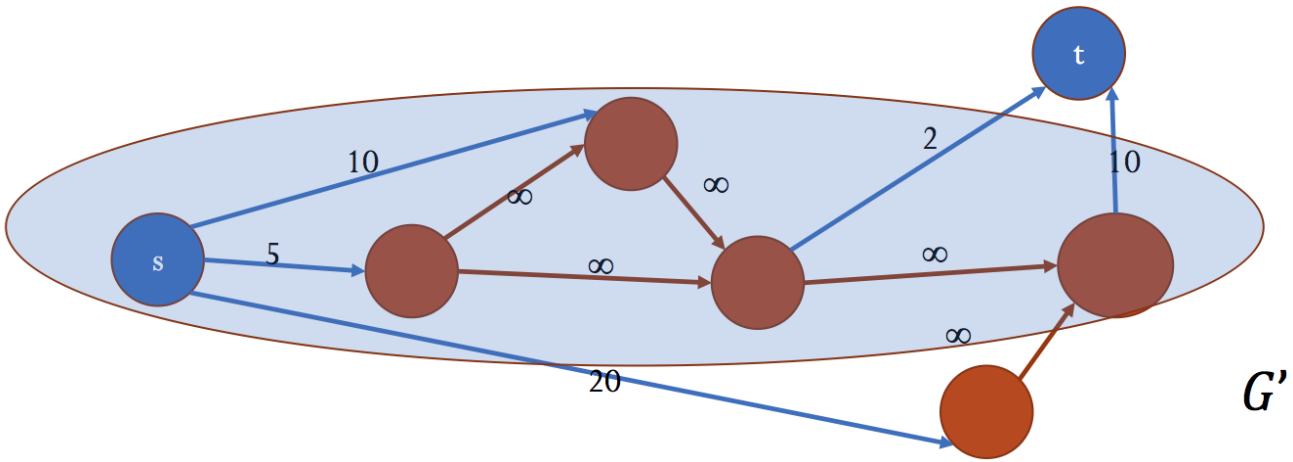


- Let $C = \sum_{i \text{ s.t. } p(i)>0} p(i)$.
- Claim 1: For any feasible subset A, there is an s-t cut (A', B') in G' such that $c(A', B') = C - \sum_{i \in A} p(i)$.
  - Proof sketch: Consider $A' = A \cup \{s\}$.

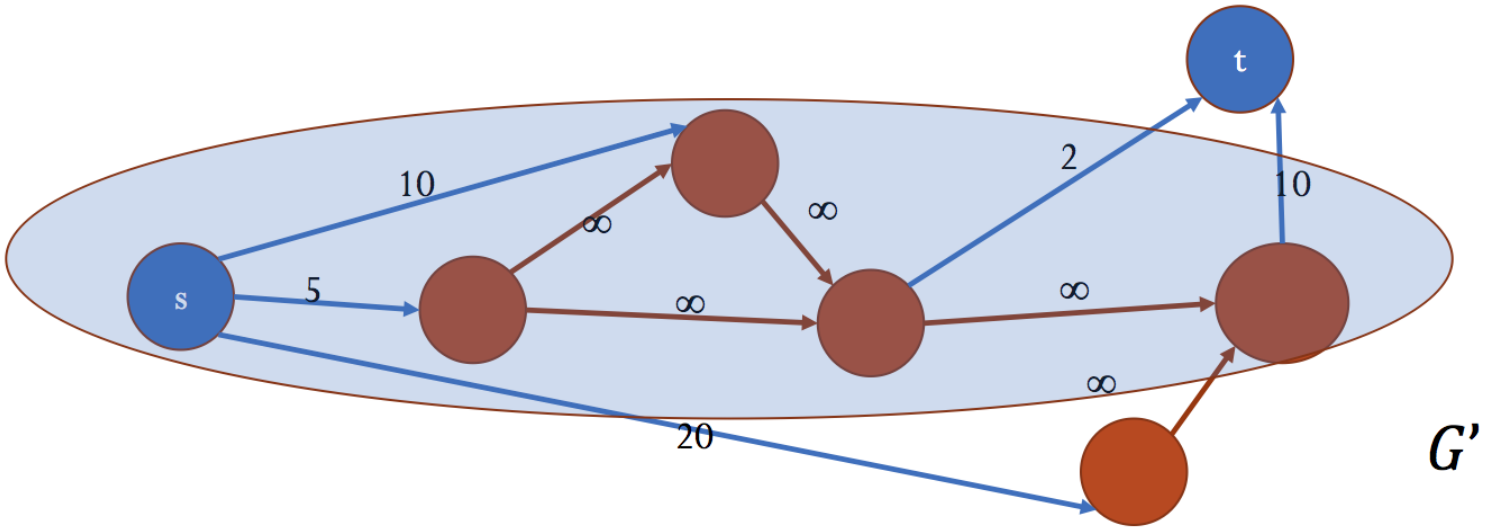


- Let $C = \sum_{i \text{ s.t. } p(i)>0} p(i)$.
- Claim 1: For any feasible subset A, there is an s-t cut (A', B') in G' such that $c(A', B') = C - \sum_{i \in A} p(i)$.
  - Proof sketch: Consider $A' = A \cup \{s\}$. We have:

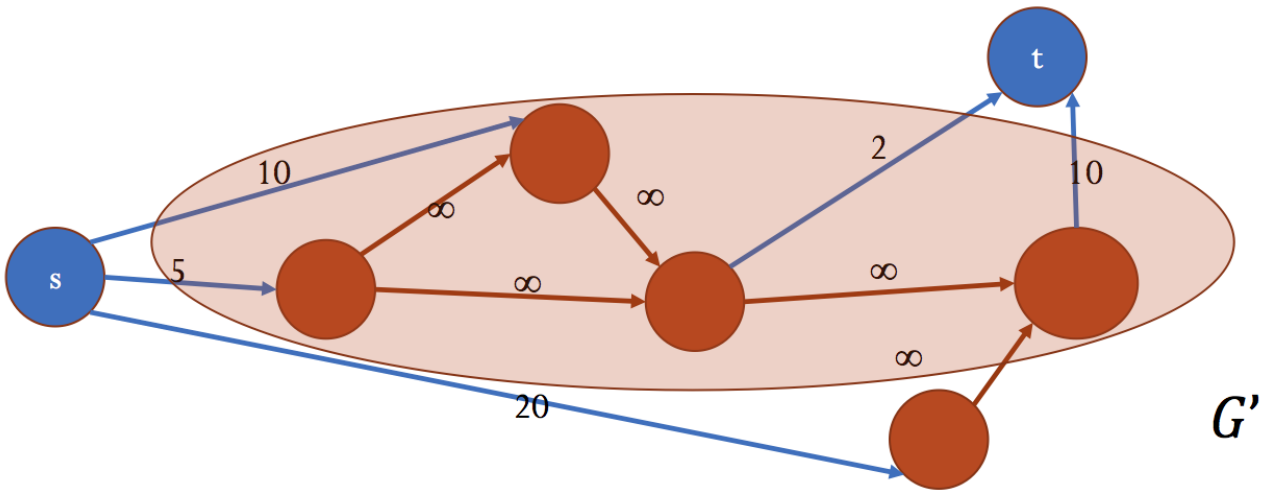
$$\begin{aligned} c(A', B') &= \sum_{i \notin A, p(i)>0} p(i) - \sum_{i \in A, p(i) \leq 0} p(i) \\ &= C - \sum_{i \in A, p(i)>0} p(i) - \sum_{i \in A, p(i) \leq 0} p(i) \\ &= C - \sum_{i \in A} p(i) \end{aligned}$$



- Let $C = \sum_{i \text{ s.t. } p(i)>0} p(i)$.
- Claim 1: For any feasible subset A, there is an s-t cut (A', B') in G' such that $c(A', B') = C - \sum_{i \in A} p(i)$.
- Claim 2: For any s-t cut (A', B') in G' of capacity at most C, $A = A' \setminus \{s\}$ is a feasible subset. Moreover, $c(A', B') = C - \sum_{i \in A} p(i)$.



- Let $C = \sum_{i \text{ s.t. } p(i)>0} p(i)$.
- Claim 1: For any feasible subset A, there is an s-t cut (A', B') in G' such that $c(A', B') = C - \sum_{i \in A} p(i)$.
- Claim 2: For any s-t cut (A', B') in G' of capacity at most C, $A = A' \setminus \{s\}$ is a feasible subset. Moreover, $c(A', B') = C - \sum_{i \in A} p(i)$.
  - Proof sketch: Since all edges (i, j) corresponding to G have capacity ∞.



- Let $C = \sum_{i \text{ s.t. } p(i)>0} p(i)$.
- Claim 1: For any feasible subset A, there is an s-t cut (A', B') in G' such that $c(A', B') = C - \sum_{i \in A} p(i)$.
- Claim 2: For any s-t cut (A', B') in G' of capacity at most C, $A = A' \setminus \{s\}$ is a feasible subset. Moreover, $c(A', B') = C - \sum_{i \in A} p(i)$.
- Claim 3: If (A', B') is the minimum cut in G', then $A' \setminus \{s\}$ is an optimal solution to the project selection problem.

End