Lab Worksheet 7: Max Flow

The Project Selection Problem

The project selection problem involves choosing a subset of activities to maximize total profit while satisfying prerequisite constraints. We are given a set P of potential projects. Each project $i \in P$ is associated with a real-valued *value* p_i, which may be positive (profit) or negative (cost).

The relationships between projects are defined by a directed acyclic Graph (DAG) $G = (P, E)$, where P is the set of vertices (projects) and E is the set of directed edges (constraints). An edge $(i, j) \in E$ represents a precedence constraint: if project i is selected, then its prerequisite, project j, *must* also be selected. A subset of projects $A \subseteq P$ is considered *feasible* if it satisfies all the constraints. This means that for every constraint $(i, j) \in E$, if $i \in A$, then j must also be in A.

The *total profit*, denoted by $\text{profit}(A)$, resulting from selecting a feasible set A is the sum of the values of all projects included in A:

$$\text{profit}(A) = \sum_{i \in A} p_i$$

The goal is to find a feasible set of projects $A \subseteq P$ that *maximizes the total profit*, $\text{profit}(A)$.

Designing the Flow Network

We solve this problem via a reduction to the max-flow problem. The idea is to construct a network flow from G in such a way that the minimum cut partition the graph G corresponds to an optimal set of projects to pick.

Constructing $G' = (V', E')$. We build a directed network with a new source s and sink t, and:

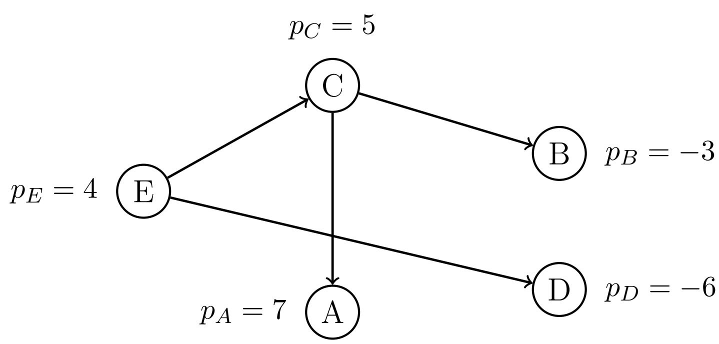
1. For each $i \in P$ with $p_i \geq 0$, add edge (s, i) of capacity p_i.
2. For each $i \in P$ with $p_i < 0$, add edge (i, t) of capacity $-p_i$.

3. For each precedence edge $(i, j) \in E$ ("to take i, you must take j"), add edge (i, j) of capacity $+\infty$[1].

Question 1. Consider the example graph below and the following projects and values:

$$P = \{A, B, C, D, E\}, \quad p_A = 7,\ p_B = -3,\ p_C = 5,\ p_D = -6,\ p_E = 4,$$

For the given graph G below, construct the network G', run the Ford-Fulkerson algorithm on it, and find the minimum $(s - t)$-cut.



Extracting the solution from a minimum cut

Question 2. Let $C \coloneqq \sum_{i \in P : p_i > 0} p_i$. This is the maximum possible profit that one can hope for without considering the precedent constraints. Let (S, T) be a minimum s–t cut in G'. Define $A \coloneqq S \setminus \{s\}$, representing the selected projects.

[1] This can be implemented by setting the capacity to any number that is strictly larger than $C \coloneqq \sum_{i \in P:\, p_i > 0} p_i$, e.g. $C + 1$.

1. Explain why A must satisfy the precedence constraints.

2. Show that the capacity of this cut is $C - \sum_{i \in A} p_i$.

Algorithm and Running Time

Question 3. Using a max-flow algorithm as a subroutine, provide pseudocode to solve the project selection problem. Analyze the running time of your algorithm. You may assume the max-flow subroutine runs in $O(|V|\,|E|^2)$ on any graph $G = (V, E)$.

Proof of Correctness

Question 4 In this part, we prove the correctness of the algorithm by establishing the two key facts: completeness and soundness.

1. (*Completeness: A feasible set implies a cut.*) Let $A \subseteq P$ be any feasible set. Show that placing $A \cup \{s\}$ on the s-side and $(P \setminus A) \cup \{t\}$ on the t-side yields a cut of capacity $C - \sum_{i \in A} p_i$.

2. (*Soundness: A finite capacity cut induces a feasible set.*) Let (S, T) be any $(s-t)$cut with capacity at most C. Prove that $A = S \setminus \{s\}$ satisfies the precedence constraints.

3. Combine the two parts to argue that a minimum cut corresponds to a maximum-profit feasible set.