Lecture 7: Feb 12, 2013

1 The Project-Selection Problem

1.1 The problem statement

The project-selection problem is as follows.
Input:

- m projects: $\mathcal{P} = \{1, 2, \ldots, m\}$
- n tools: $\mathcal{T} = \{1, 2, \ldots, n\}$
- For each project $i \in \mathcal{P}$,

$$b_i = \text{benefit gained by selecting project } i \text{ to be completed, and}$$

$$T_i = \text{the set of tools required to complete job } i.$$

- For each tool $j \in \mathcal{T}$,

$$c_j = \text{the cost of selecting to purchase tool } j.$$

Objective: To determine a feasible selection of projects and tools (P', T'), where $P' \subset \mathcal{P}, T' \subset \mathcal{T}$, in order to maximize net profit:

$$\max_{(P',T') \text{ feas}} \sum_{i \in P'} b_i - \sum_{j \in T'} c_j,$$

where (P', T') is a feasible selection if it satisfies the following set of constraints. **Constraints**: if project i is selected (i.e. $i \in P'$), then all tools in T_i must also be selected (i.e., $j \in T_i$).

1.2 Modeling project-selection as a minimum cut problem

Step 1: We construct the following input for the minimum cut problem.

- A directed graph $G = (N, E)$ where

$$N = \{s, t\} \cup \mathcal{P} \cup \mathcal{T},$$

and

$$E = \{(s, i) | \forall i \in \mathcal{P}\} \cup \{(i, j) | \forall i \in \mathcal{P}, j \in T_i\} \cup \{(j, t) | \forall j \in \mathcal{T}\}.$$

- Edge capacities:

$$\begin{aligned} u_{si} &= b_i && \text{forall } i \in \mathcal{P} \\ u_{ij} &= +\infty && \text{forall } i \in \mathcal{P},\ j \in T_i \\ u_{jt} &= c_j && \text{forall } j \in \mathcal{T} \end{aligned}$$

Step 2: We show a correspondence between feasible selections of projects and tools, and feasible cuts with finite capacities.

- Suppose that we are given a feasible selection of projects and tools, call it (P', T'). Construct the corresponding cut as follows:

$$\begin{aligned} S &= \{s\} \cup P' \cup T', \\ T &= \{t\} \cup (\mathcal{P} \backslash P') \cup (\mathcal{T} \backslash T'), \end{aligned}$$

  (that is, T contains the sink, and all project and tool nodes that are not in S).

  Then, we want to show that the cut (S, T) has a finite capacity:

  - The only way for the cut (S, T) to have infinite capacity is if there is an edge (i, j) with capacity $u_{ij} = +\infty$, where $i \in S$, $j \in T$.
  - But the only edges with infinite capacities are edges (i, j) where i is a project and j is a tool needed by the project.
  - So, the only way for the cut (S, T) to have infinite capacity is to have a project node i that is in S, the source-side of the cut, and a tool $j \in T_i$ that is in T, the sink-side of the cut.
  - This means that the only way for the cut (S, T) to have infinite capacity is when i is in P' (because S contains the projects that are selected, in P') but j is not in T' (because if j is in T', then it would have been in S).
  - That is, the only way for the cut (S, T) to have infinite capacity is when the corresponding selection of projects and tools is infeasible.
  - So, (S, T) must have a finite capacity.

- Next, we show the other direction of the correspondence. Suppose that we are given a feasible cut with finite capacity, call it (S, T). Construct the corresponding selection of projects and tools as follows:

$$\begin{aligned} P' &= \mathcal{P} \cap S = \text{ projects that are in } S\text{, the source-side of the cut} \\ T' &= \mathcal{T} \cap S = \text{ tools that are in } S\text{, the source-side of the cut} \end{aligned}$$

  Then, we want to show that (P', T') is a feasible selection of projects and tools. That is, we want to show that if project i is in P', then each $j \in T_i$ must also be selected to be in T'.

  - We know that (S, T) has finite capacity, so for all edges (i, j) with infinite capacity, either both i and j are in S, or i is in T.
  - If (i, j) has infinite capacity and both i and j are in S, then this means that project i is in S, and each tool j needed by i is also in S. So, the project i is in the set P' and all its necessary tools are in T'.
  - If i is in T, then project i is not selected, so we don't have to worry about the tools that project i needs.
  - Hence, the selection (P', T') is a feasible selection of projects and tools.

Step 3: We show a correspondence between the net profit of a feasible selection of projects and tools, and the capacity (finite) of the corresponding cuts. This will then show that a selection with the maximum net profit corresponds to a cut with the minimum capacity.

Consider a feasible selection of projects and tools (P', T') and its corresponding cut (S, T), where the correspondence is as described in step 2 above, namely, $S = \{s\} \cup P' \cup T'$ and T contains all other nodes that are not in S.

We will show that if $PROFIT(P', T')$ denote the net profit of selection (P', T'), then the capacity of the cut (S, T) is

$$CAPACITY(S, T) = -PROFIT(P', T') + B,$$

where $B = \sum_{i \in \mathcal{P}} b_i$ is a constant that does not depend on the choice of (P', T') or (S, T).

To see this, note that

$$PROFIT(P', T') \;=\; \sum_{i \in P'} b_i - \sum_{j \in T'} c_j,$$

so,

$$\begin{aligned}
-PROFIT(P', T') + B &= -\sum_{i \in P'} b_i + \sum_{j \in T'} c_j + \sum_{i \in \mathcal{P}} b_i \\
&= \sum_{i \in \mathcal{P}, i \notin P'} b_i + \sum_{j \in T'} c_j \\
&= \sum_{i \in \mathcal{P}, i \notin P'} b_i + \sum_{j \in T'} c_j \\
&= \sum_{i \in \mathcal{P}, i \notin S} u_{si} + \sum_{j \in \mathcal{T}, j \in S} u_{jt} \\
&= \sum_{s \in S, i \notin S} u_{si} + \sum_{j \in S, t \notin S} u_{jt} \\
&= \sum_{v \in S, w \notin S} u_{vw} \\
&= CAPACITY(S, T).
\end{aligned}$$

This shows the correspondence between the objective values. Hence, the optimal solution must also corresponds, because

$$\begin{aligned}
\max_{(P', T')\text{feas}} PROFIT(P', T') &= \max_{(S,T)\text{feas, finite cap}} (-CAPACITY(S, T) + B) \\
&= \left(\max_{(S,T)\text{feas, finite cap}} -CAPACITY(S, T) \right) + B \\
&= -\left(\min_{(S,T)\text{feas, finite cap}} CAPACITY(S, T) \right) + B.
\end{aligned}$$

So, the problem of maximizing net profit in the project selection problem is equivalent to finding the minimum cut in the constructed maxflow/mincut problem.