Applications of Flow Networks (cont.)

Finding Minimum Vertex Cover

**Definition 1** Let \( G(V, E) \) be an undirected connected finite graph. Vertex cover \( U \) of \( G \) is a subset of vertices \( U \subseteq V \) such that for each edge \( \{u, v\} \in E \) at least one of the vertices \( \{u, v\} \) is in \( U \).

**Task:** find a vertex cover of the minimum size (number of vertices) for a given bipartite graph \( G(V = A \cup B, E) \).

**Example**

![Graph Example](image)

**Idea of solution:** reduction to a problem of finding of a minimum cut.

**Solution.** Define a flow network \( N(G'(V', E'), s, t, c) \) such that:

\[
\begin{align*}
V' &= V \cup \{s, t\} ; \\
E' &= \{(s, u) \mid u \in A\} \cup \{(v, t) \mid v \in B\} \cup \{(u, v) \mid \{u, v\} \in E, u \in A, v \in B\} ; \\
\forall u \in A : c((s, u)) &= 1 ; \\
\forall v \in B : c((v, t)) &= 1 ; \\
\forall \{u, v\} \in E, u \in A, v \in B : c((u, v)) &= +\infty .
\end{align*}
\]
Example

Claim 1. A minimum cut in $\mathcal{N}$ has a finite capacity.
Proof. For example, a cut between $\{s\}$ and $\mathcal{V}\setminus\{s\}$ has a finite capacity, and its capacity is not necessarily minimal.

Claim 2. A cut of a finite capacity cannot contain an edge of an infinite capacity.
Proof. Trivial.

Let $(S : \bar{S})$ be a cut such that $s \in S$, $t \notin S$. Define the following set of vertices:

$$U = (A\setminus S) \cup (B \cap S).$$

Lemma 1 $(S : \bar{S})$ has a finite capacity if and only if $U$ is a vertex cover.

Proof.
1. Assume that \((S : \bar{S})\) has a finite capacity. As it can be seen from the picture, the only edges going from \(A\) to \(B\) that are uncovered by the selected vertex set are the edges that have one endpoint in \(A \cap S\) and the other endpoint in \(B \setminus S\). Such edges are in the cut \((S : \bar{S})\). The capacity of this cut is finite. Therefore, there are no edges of infinite capacity in this cut, while the above edge has infinite capacity. We conclude that there are no uncovered by \(U\) edges going from \(A\) to \(B\).

2. Assume that \(U\) is a vertex cover. Consider the cut \((S : \bar{S})\). This cut does not contain edges from \(A \cap S\) to \(B \setminus S\) (otherwise, such an edge is not covered by \(U\)). Therefore, the only edges that such cut can contain (in a forward direction) are edges from \(s\) to \(A\) and edges from \(B\) to \(t\). Those are all edges of capacity 1, and therefore the corresponding cut has a finite capacity.

\[\square\]

**Lemma 2** Capacity of the cut \((S : \bar{S})\) is equal to the size of the cover defined by that cut.

**Proof.**

\[c(S) = |A \setminus S| \cdot 1 + |B \cap S| \cdot 1 = |U| .\]

\[\square\]

These observations yield the following algorithm.

**Algorithm for Finding a Minimum Vertex Cover**

The following algorithm finds the maximum matching.

**Input**: undirected bipartite graph \(G = (V, E)\)

**Output**: minimum vertex cover \(U \subseteq V\)

1. Build \(N(G'(V', E'), s, t, c)\);
2. Run Ford-Fulkerson to find a maximum flow;
3. Find a minimum cut \((S : \bar{S})\);
4. \(U \leftarrow (A \setminus S) \cup (B \cap S)\);

**Algorithm 1**: Algorithm for Minimum Vertex Cover in a Bipartite Graph

**Time complexity**

**Step 1**: \(O(|V| + |E|)\). Since the graph is connected, this reduces to \(O(|E|)\).

**Step 2**: finding an augmenting path requires \(O(|E|)\). Each augmenting path improves the total flow by at least one unit. The total flow is bounded from above by \(|V|\) (a cut between \(\{s\}\) and \(V \setminus \{s\}\) has capacity \(\leq |V|\)). Therefore, the time complexity of Step 2 is \(O(|E||V|)\).

**Step 3**: \(O(|E|)\) (for example, run a BFS from the vertex \(s\) by using only useful edges).

Therefore, the total complexity of the proposed algorithm is \(O(|E||V|)\).
Networks with Upper and Lower Bounds

Previously, the edge rule required that $0 \leq f(e) \leq c(e)$ for all edges $e$. In the following discussion we assume that there are lower bounds on the flow at all edges, in addition to upper bounds. More formally, given a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, consider two functions $b, c : \mathcal{E} \to \mathbb{R}^+$, such that for each edge $b(e) \leq c(e)$.

**New edge rule**

For each $e \in \mathcal{E}$: $b(e) \leq f(e) \leq c(e)$.

The vertex rule remains unchanged, as well as the definition of the total flow. The goal is to find a maximum flow in such a network $\mathcal{N}(\mathcal{G}(\mathcal{V}, \mathcal{E}), s, t, b, c)$ with the upper and lower limits.

**Example** As it is illustrated by the following example, in some networks there might be no legal flow at all.

![Diagram](image)

The flow entering the middle vertex should be at most 1, while the flow leaving it should be at least 2. This is a contradiction to the vertex rule, and therefore there is no legal flow in this network. □

The problem of finding a maximum flow in a given network $\mathcal{N}(\mathcal{G}(\mathcal{V}, \mathcal{E}), s, t, b, c)$ is divided into two taks.

1. Find a legal flow in $\mathcal{N}$ or decide that such flow does not exists.

2. If a legal flow exists, iteratively improve it until a maximum flow is found.

**Method of Ford-Fulkerson**

In the proposed method, we construct a new network $\tilde{\mathcal{N}}$ (with upper limits only) with the underlying graph $\tilde{\mathcal{G}}(\tilde{\mathcal{V}}, \tilde{\mathcal{E}})$ as follows.

1. **Vertex set**: $\tilde{\mathcal{V}} = \mathcal{V} \cup \{\tilde{s}, \tilde{t}\}$, where $\tilde{s}$ and $\tilde{t}$ are a new source and a new sink.

2. **Edge set**:

   $$R = \{(\tilde{s}, v) \mid v \in \mathcal{V}\};$$
   $$T = \{(u, \tilde{t}) \mid u \in \mathcal{V}\};$$
   $$\tilde{\mathcal{E}} = \mathcal{E} \cup R \cup T \cup \{e', e''\};$$
   $$e' = (s, t);$$
   $$e'' = (t, s);$$
3. Capacity function $\tilde{c} : \tilde{E} \to E \to \mathbb{R}^+$:

For $e \in E$ :
\[ \tilde{c}(e) = c(e) - b(e) ; \]

For $e = (\tilde{s}, v) \in R$ :
\[ \tilde{c}(e) = \sum_{e \in \text{In}(v)} b(e) ; \]

For $e = (u, \tilde{t}) \in T$ :
\[ \tilde{c}(e) = \sum_{e \in \text{Out}(u)} b(e) ; \]
\[ \tilde{c}(e') = +\infty ; \]
\[ \tilde{c}(e'') = +\infty . \]

Example

Consider a network $\mathcal{N}$ with upper and lower bounds as shown in the following figure.
By applying the aforementioned construction, we obtain the following network $\tilde{N}$.

**Theorem 3** The network $\mathcal{N}$ has a legal flow if and only if the maximum flow in the network $\tilde{N}$ saturates all the edges leaving $\tilde{s}$ (i.e., the edges in the set $R$).