Max Flow - Min Cut: Examples

Matchings en grafos regulares

Un grafo bipartido $G = (V, E)$, dónde $V = L \cup R$, es d-regular si todo vértice tiene grado d.

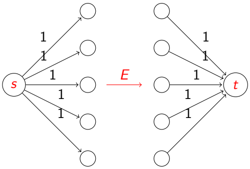
Demuestra que todo grafo bipartido d-regular tiene un matching the tamaño $|L|$. Siempre tiene un matching perfecto?

Ayuda: considerad la red de flujo correspondiente y analizar la capacidad del min cut.

Una solución

Si el grafo es bipartido y d-regular, $d|L| = |E|$ y $d|M| = |E|$. Por lo tanto $|L| = |R|$.

Como el grafo es bipartido podemos utilizar la formulación de maximum matching como problema de flujo y analizar la capacidad de los cortes.



En la red de flujo conectamos s con todas los vértices de L y todos los vértices de R con t. Luego orientamos todas las aristas de L hacia R. Asignamos capacidad 1 a todas las aristas.

s, t-cortes

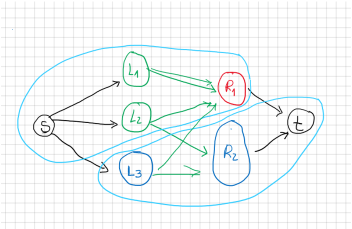
Como $cut(\{s\}, V \cup \{t\}) = |L|$. Nos basta con demostrar que cualquier otro s, t-corte tiene capacidad $\geq |L|$.

- $cut(\{s\} \cup V, \{t\}) = |R| = |L|$
- $cut(\{s\} \cup L, R \cup \{t\}) = d|L| \geq |L|$

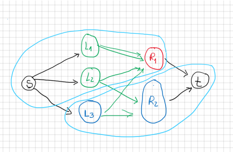
Nos queda analizar cortes que dividan L y R.

s, t-cortes

Consideremos un s, t-corte $(\{s\} \cup L_1 \cup L_2 \cup R_1, L_3 \cup R_2 \cup \{t\})$
(L_1, L_2, L_3) es una partición de L y
(R_1, R_2) es una partición de R.
si $u \in L_1$, $N(u) \subseteq R_1$ y, si $u \in L_2$, $N(u) \cap R_2 \neq \emptyset$.



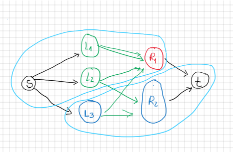
s, t-cortes



Como todos los vértices en L_2, R_1 tienen al menos un vecino en T y s está conectado con todos los vértices de L_3.

$$cut(S, T) = \geq |L_3| + |L_2| + |R_1|$$

s, t-cortes



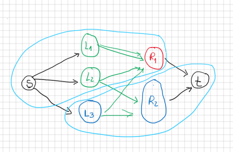
Como todos los vértices en L_2, R_1 tienen al menos un vecino en T y s está conectado con todos los vértices de L_3.

$$cut(S,T) = \geq |L_3| + |L_2| + |R_1|$$

- Todos los vecinos de vértices en L_1 están en R_1.
- El grafo es d-regular, $d|R_1| \geq d|L_1|$, y $|R_1| \geq |L_1|$.

$$cut(S,T) \geq |L_3| + |L_2| + |L_1| \geq |L|.$$

s, t-cortes



Como todos los vértices en L_2, R_1 tienen al menos un vecino en T y s está conectado con todos los vértices de L_3.

$$cut(S, T) =\geq |L_3| + |L_2| + |R_1|$$

- Todos los vecinos de vértices en L_1 están en R_1.
- El grafo es d-regular, $d|R_1| \geq d|L_1|$, y $|R_1| \geq |L_1|$.

$$cut(S, T) \geq |L_3| + |L_2| + |L_1| \geq |L|.$$

Siempre hay un matching de tamaño $|L|$ en un grafo bipartido d-regular y como $|L| = |R|$ este matching es perfecto.

Project selection

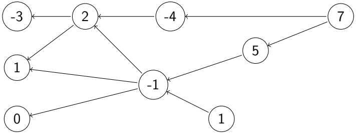
- We have a set of projects with prerequisites among them.
  - Set of possible projects P: project v has associated revenue p_v (positive or negative).
  - Set of prerequisites E: $(v, w) \in E$ means w is a prerequisite for v.
  - A subset of projects $A \subseteq P$ is feasible if the prerequisite of every project in A also belongs to A.

Project selection

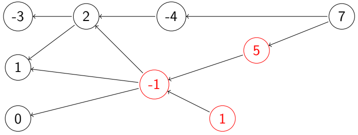
- We have a set of projects with prerequisites among them.
  - Set of possible projects P: project v has associated revenue p_v (positive or negative).
  - Set of prerequisites E: $(v, w) \in E$ means w is a prerequisite for v.
  - A subset of projects $A \subseteq P$ is feasible if the prerequisite of every project in A also belongs to A.
- Project selection problem. Given a set of projects P and prerequisites E, choose a feasible subset of projects to maximize revenue.

We can see (P, E) as a directed graph.

Project selection: feasibility

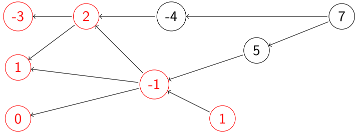


Project selection: feasibility



A is unfeasible

Project selection: feasibility

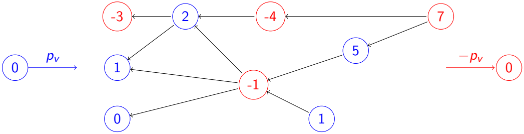


A is feasible and has revenue 4.

Project selection: flow network

- Add vertices s and t to (P, E)
  - Assign a capacity of ∞ to each prerequisite edge.
  - Add edge (s, v) with capacity p_v if $p_v > 0$.
  - Add edge (v, t) with capacity $-p_v$ if $p_v < 0$.
  - For convenience, define $p_s = p_t = 0$.

Project selection: flow network



Project selection: min-cut equivalence

Claim

(A, B) is a min cut iff $A - \{s\}$ is an optimal set of projects

- In a min cut, infinite capacity edges ensure $A - \{s\}$ is feasible.
- To see that it is an optimal solution let us analyze a generic (s, t)-cut

$$
\begin{aligned}
cut(A, B) &= \sum_{v \in B : p_v > 0} p_v + \sum_{v \in A : p_v < 0} (-p_v) \\
&= \sum_{v \in B : p_v > 0} p_v + \sum_{v \in A : p_v > 0} p_v - \sum_{v \in A : p_v > 0} p_v - \sum_{v \in A : p_v < 0} p_v \\
&= \sum_{v \in P : p_v > 0} p_v - \sum_{v \in A} p_v
\end{aligned}
$$

Project selection min-cut equivalence

$$
\begin{aligned}
cut(A,B) &= \sum_{v\in B: p_v>0} p_v + \sum_{v\in A: p_v<0} (-p_v) \\
&= \sum_{v\in B: p_v>0} p_v + \sum_{v\in A: p_v>0} p_v - \sum_{v\in A: p_v>0} p_v - \sum_{v\in A: p_v<0} p_v \\
&= \sum_{v\in P: p_v>0} p_v - \sum_{v\in A} p_v
\end{aligned}
$$

- The first term does not depend on the cut, only of the input.
- Minimizing $cut(A,B)$ is the same as maximizing $\sum_{v\in A} p_v$, i.e, maximizing the revenue.