Max Flow - Min Cut: Examples

Ferits i hospitals

Considereu el següent escenari. Ha succeït una catàstrofe a una ciutat gran, el equips de rescat tenen identificats n ferits greus a diferents llocs de la ciutat, que necessiten ser traslladats urgentment a un hospital. Hi han k hospitals disponibles.
Depenent de a on el ferit esta situat necessita anar-hi a un hospital que estigui com a màxim a 30 minuts de distancia per ambulància.
Per a no col·lapsar urgències, voleu seleccionar els hospitals de manera que cada hospital rebi com a màxim $\lceil n/k \rceil$ ferits.

Dissenyeu un algorisme polinòmic per que un cop rebeu la informació sobre el lloc on es cada ferit juntament amb els hospitals a menys de 30 minuts, pugueu determinar si es possible que tots els ferits siguin transportats a un hospital de manera que arriben abans de 30 minuts i que cap hospital rebi més de $\lceil n/k \rceil$ ferits.

Una solución.

Lo pantearemos como un problema de asignación con restricciones en una red de flujo en la que una unidad de flujo de s a t represente la asignación de un herido a un hospital.

La red $\mathcal{N}$ tiene

- Nodos: s, t, F, H, $|F| = n$ representa heridos, $|H| = k$ representa hospitales
- Aristas y capacidades:

$$\begin{aligned} \{(s, f) \mid f \in F\} & \qquad \text{capacidad } 1 \\ \{(h, t) \mid h \in H\} & \qquad \text{capacidad } \lceil n/k \rceil \\ \{(f, h) \mid f \in F, h \in H, d(f, h) \leq 30'\} & \qquad \text{capacidad } 1 \end{aligned}$$

Validez de la reducción

Un camino de s a t tiene la forma $s \to f \to h \to t$, si transporta una unidad de flujo, lo interpretaremos como: el herido f se puede transportar al hospital h.

Las conexiones (f, h) garantizan que el herido y el hospital están a la distancia correcta.

El posible flujo que llega a h, total de heridos en h, se transmite por (h, t), la capacidad garantiza que el flujo no supera el tope por hospital.

Validez de la reducción

Si el problema tiene solución podremos asignar todos los heridos a hospitales dentro del radio permitido. Garantizando además que ningún hospital recibe más de $\lceil n/k \rceil$ heridos.

Si traducimos cada asignación en una unidad de flujo en el camino asociado, obtenemos un flujo válido. Con valor de flujo n que es el máximo posible.

Validez de la reducción

Si el valor del flujo máximo en la red es n, sabemos que hay un flujo f con valores enteros que alcanza ese valor.

Si $f(f, h) = 1$ asignamos al herido f al hospital h, así asignamos a cada herido un hospital, dentro del radio.

El flujo total en la arista (h, t), se corresponde, de acuerdo con la ley de conservación de flujo, con el número total de heridos transportados al hospital h. La capacidad de la arista garantiza que la asignación cumple la condición en los hospitales.

Algoritmo

Construir $\mathcal{N}$

$F = MaxFlow(\mathcal{N})$

if $|F| < n$ **then**

  return NO

else

  return $\{(f, h) \mid F[f, h] = 1\}$

Coste

El coste de la llamada a MaxFlow: El número de vértices en la red es $N = n + k + 2$, el número de aristas es $M = n + k + nk$.

- Con FF el coste es $O(n(N + M)) = O(n^2 k)$
- Con EK el coste es $O(M^2 N) = O(n^2 k^2 (n + k))$

Utilizaremos FF.

Construir $\mathcal{N}$ tiene coste $O(N + M)$ y el último paso $O(nk)$. El coste total del algoritmo es $O(n^2 k)$.

Matchings en grafos regulares

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

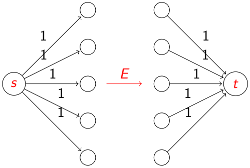
Demuestra que todo grafo bipartido d-regular tiene un matching the tamaño $|L|$. Siempre tiene un matching perfecto?

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.

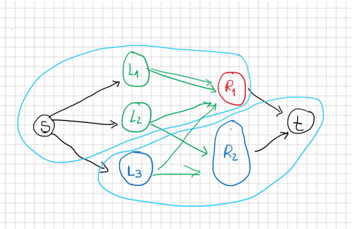
s, t-cortes

Consideremos un s, t-corte $(\{s\} \cup L_1 \cup L_2 \cup R_1, L_3 \cup R_2 \cup \{t\})$

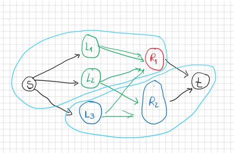
(L_1, L_2, L_3) es una partición de L y

(R_1, R_2) es una partición de R.

si $u \in L_1$, $N(u) \subseteq R_1$ y, si $u \in L_2$, $N(u) \cap R_2 \neq \emptyset$.



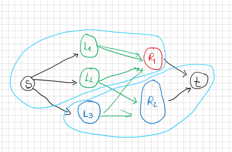
s, t-cortes



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



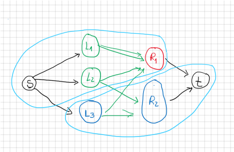
Como todos los vértices en L_2, R_1 tienen al menos un vecino en T y s está conectado con todos los vértices de L_3.

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

- 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

Put drawing

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 $p_s = p_t = 0$.

Add drawing

Project selection: min-cut equivalence

Claim

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

- Associated to a MaxFlow, infinite capacity edges and conservation law 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.