AA-Entrega-5

Oriol Miró López-Feliu

April 2023

1 Problema 7

Donat com a entrada un graf no dirigit $G = (V, E)$, definim el problema de GST (graf sense triangles) com el problema de seleccionar el màxim nombre d'arestes E, de manera que el graf $G_0 = (V, E_0)$ no contingui cap triangle (és a dir, no hi ha 3 vèrtexs x, y, z tal que (x, y), (x, z) i (y, z) siguin arestes a G_0). Aquest problema és NP-hard.

Donat $G = (V, E)$, considereu el següent algorisme golafre (greedy):

1. Ordeneu els vèrtexs en ordre decreixent respecte al seu grau (nombre veïns). Considereu el resultat de l'ordenació que s'escriu com a $(v_1, v_2, ..., v_n)$.
2. Inicialment tenim $S = \emptyset = S^c$. Al primer pas, $v_1 \in S$.
3. Al pas i-èssim, col·loquem v_i a S o S^c de manera que maximitzi $|C(S, S^c)|$.
4. Prenem com $E' = (u, v)|u \in S, v \notin S$.

Aquest algorisme és una 2-aproximació polinòmica al problema?

Correcte i Polonòmic

L'algorisme és correcte ja que el resultat és el conjunt d'arestes entre dos conjunts de vèrtexs. És per tant un graf bipartit, el qual mai té cicles de grau tres.

Sigui $n = |V$, s'executa en temps polinòmic ja que tots els passos requereixen temps polinòmic. Ordenar els vèrtexs en ordre creixent té un cost diferent depenent de la manera en què se'ns doni el graf però ho podem fitar per $O(n^2)$. Seguidament, inicialitzar els conjunts requereix temps constant, i calcular per a cada v_i si va a S o S^c implica recórrer tots els vèrtexs d'ambdós conjunts per a comptar la connectivitat amb v_i, el qual es polinòmic en n. Finalment, prendre les arestes entre els dos conjunts és un recorregut de cada vèrtex d'un conjunt a tots els altres del segon conjunt, amb cost depenent de l'entrada però ho podem fitar amb $O(n^2)$.

Amb això podem concloure que l'algorisme és correcte i polinòmic.

És una 2-Aproximació

Per demostrar que el nostre algorisme es una 2-aproximació, ens cal demostrar que la solució que produeix sempre es major o igual que la meitat del òptim.

Si el graf no tinguis ningun triangle, es a dir, fos perfectament bipartible, al maximitzar en cada pas la connectivitat entre S i S^c, l'algoritme produiria l'optim. En cambi, si tenim triangles això ens porta a decisions erroneas.

Per ilustrar això, sigui G' un graf només format per triangles. Donats dos triangles en aquest graf, formats per les arestes (x, y), (y, z), (z, x) i (x, y), (y, p), (p, x), es a dir, dos triangles que comparteixen l'aresta (x, y), la solució optima sería treure aquesta aresta, deixant-nos 4 arestes. En cambi, el nostre algoritme sí que agafara aquesta aresta, fent el màxim nombre d'arestes obtenibles en aquest cas de 3.

Anem doncs a argumentar en base a quines decisions prenem en cada punt.

Quan considerem el vertex v_i, l'afegirem al conjunt S o S^c en funció de en quin maximitzem la connectivitat entre conjunts. Per tant es donen els seguents dos casos:

1. Si el vertex té grau parell, al pendre la decisió de afegir-lo a un conjunt sempre garantim que tindrá ¿= 1/2 de les seves arestes cap al altre.

2. Si el vertex té grau imparell, per la mateixa idea el posarem on maximitzi la connectivitat, i per tant tindrá ¿1/2 de les seves arestes cap a l'altre conjunt.

Argumentant en vers a les arestes que aporta cada vertex a la solució, podem veure que, sigui OPT la solució òptima, i per tant $|OPT|$ el cost d'aquesta solució:

Al ser $|E| \geq |OPT|$,

$\sum_{i=0}^{n}$ Arestes aportades per vertex $\geq \frac{|E|}{2} \geq \frac{|OPT|}{2}$

Per tant hem concluit que és una 2-aproximació del óptim.