84. En sociologia, sovint s'estudia un graf $G = (V, E)$ en el qual els nodes representen les persones, i les arestes representen els que són amics entre si. Suposem que l'amistat és simètrica, per la qual cosa podem considerar G com un graf no dirigit.
Volem estudiar aquest graf G, per a trobar grups de persones, molt unides en el sentit de que tots són amics de tots. Una manera de formalitzar aquesta idea seria la següent. Per a un subconjunt $S \subseteq V$ sigui $e(S)$ el nombre d'arestes dintre d'S, és a dir, el nombre d'arestes que tenen tots dos extrems en S. Definim la *cohesió* d'S com $e(S)/|S|$. en aquest tipus de grafs, un paràmetre natural podia ser el conjunt $S \subseteq V$ amb cohesió màxima.

    (a) Doneu un algorisme polinòmic que pren com a entrada G i un nombre α racional, i determina si hi ha un conjunt S amb cohesió $> \alpha$.
    (b) Doneu un algoritme polinòmic per trobar un conjunt $S \subseteq V$ amb la màxima cohesió.

Una solución

Antes de definir el algoritmo vamos a asociar el problema con un parámetro de una red de flujo. Construimos la siguiente red:

- Por cada $\{u, v\} \in E$ añadimos dos arcos (u, v) y (v, u) con capacidad 1.
- Añadimos vertice s y para cada $u \in V$ un arco (s, u) con capacidad $d(u)$.
- Añadimos vertice t y para cada $u \in V$ un arco (u, t) con capacidad 2α.

Analizemos la capacidad de los $s - t$ cortes (S, T). Supongamos que $S = \{s\} \cup A$ and $T = \{t\} \cup B$.

Notemos que $c(\{s\}, V \cup \{t\}) = 2m$ y $c(V \cup \{s\}, \{t\}) = 2\alpha n$.

Si $\alpha < m/n$, V tiene la cohesión requerida

En caso contrario $2m < 2\alpha n$ y mincut $\leq 2m$.

Para un corte genérico la capacidad del corte es:

$$\begin{aligned} c(S,T) &= \sum_{u \in B} d(u) + \sum_{v \in A} 2\alpha + c(A,B) \\ &= 2\alpha|A| + 2e(B) + 2c(A,B) \\ &= 2\alpha|A| + 2m - 2e(A) \\ &= 2(m + \alpha|A| - e(A)) \end{aligned}$$

$c(S,T) > 2m$ iff $\alpha|A| - e(A) < 0$ iff $\alpha|A| < e(A)$ iff $e(A)/|A| > \alpha$.

Hay un conjunto con cohesion $> \alpha$ iff min-cut $< 2m$

(a) La red se puede construir en tiempo polinómico. Luego calculamos el valor de MaxFlow usando Edmonds Karp en tiempo $O(nm^2)$. Y hacemos la comparación final. Obteniendo un algoritmo con coste polinómico.

(b) Primero calcularemos el valor mas grande de α ($\alpha_{\max}$) para el que existe un conjunto ocn cohesión α. Para ello combinaremos búsqueda binaria con el algoritmo del apartado anterior. Una vez obtenido el valor $\alpha_{\max}$, obtenemos un flujo f con valor máximo usando Edmonds Karp en tiempo $O(nm^2)$ para $\alpha_{\max}$. A partir de f obtenemos el grafo residual y tomamo como A los vertices accesibles desde s en este grafo. De acuerdo con el teorema maxflow-mincut $s \cup A$ define un corte con capacidad máxima y de acuerdo con el resultado anterior A tiene cohesión máxima. Como en el paso anterior el tiempo total es polinómico ya que la cohesión de un conjunto es un valor entre 0 y n.

85. Si utilitzem l'algorisme de Huffmann per a comprimir un text format per n, símbols que apareixen amb freqüències $f_1, f_2, f_3, \ldots, f_n$ quina és la màxima longitud de compressió d'un símbol que podem obtenir? Doneu un algorisme per obtenir freqüències on es compleix aquesta condició per un valor de n donat.

86. (El problema de l'agent de borsa en temps de crisi.) En temps de crisi, un problema complicat per als agents de borsa és el de vendre un gran nombre d'accions de la mateixa entitat, quan el valor d'aquestes accions ha caigut força a la borsa. És difícil trobar el moment més adient per a vendre, però el fet de posar a la venda de cop un nombre gran d'accions de la mateixa companyia, té l'efecte de fer caure el valor d'aquestes accions encara més.

Considereu el següent model simplificat del món borsari. Volem vendre x participacions de l'entitat "Despropòsits Units" (a partir d'ara DU), i suposem que tenim un model de l'evolució del mercat que ens prediu que als propers n dies, el valor de cada acció de DU al mercat serà $p_1, p_2, \ldots, p_n$. A més, el nostre model té un corrector del preu per al cas de venda massiva d'accions de la mateixa entitat: una funció $f(\cdot)$ tal que si venem y accions d'una mateixa entitat el mateix dia, decrementa el preu de l'acció en $f(y)$ (evidentment f depèn de cada entitat), per tant al primer dia, el preu real de venda de cada acció de DU és $p_1 - f(y_1)$, i els "guanys"del primer dia seran $y_1(p_1 - f(y_1))$. Els dies següents també absorbiran el decrement en el preu de venda degut a la venda massiva dels dies previs, és a dir el guany de l'i-èsim dia serà $y_i(p_i - \sum_{j=1}^{i} f(y_i))$.

Dissenyeu un algorisme eficient que, donada com a entrada la previsió de preus $\{p_i\}_{i=1}^{n}$, el nombre total x d'accions a vendre i la funció $f = (f(1), f(2), \ldots, f(x))$, ens determini la millor forma de repartir la venda de les x accions en n dies de manera que obtinguem el màxim guany. És a dir, volem trobar els enters $y_1, y_2, \ldots, y_n$ tals que $x = \sum_{i=1}^{n} y_i$, i venent y_i accions l'i-èsim dia obtinguem el màxim profit total (a la fi dels n dies).

Considereu que $\{p_i\}_{i=1}$ és monòtona decreixent i $\{f(j)\}_{j=1}^{x}$ és monòtona creixent. Evidentment, no cal que utilitzeu tots els n dies (vegeu l'exemple a sota). La complexitat del vostre algorisme ha de ser un polinomi en x i n.

Exemple: Si $n = 3$, $p_1 = 90, p_2 = 80, p_3 = 40$, $x = 100.000$ amb $f(y) = 1$ si $y \leq 40.000$, i altrament $f(y) = 20$.

Si venem totes les accions el primer dia obtenim un preu de 70 per acció i un guany total de 7.000.000. D'altra banda, si venem 40.000 accions el dia 1 i 60.000 el dia 2, tenim un guany total de 7.100.000.

87. Suposem que volem el canvi de n cèntims, utilitzant el mínim de monedes de denominacions 1, 10 i 25 cèntims. Considereu la següent estratègia golafre: si la quantitat que em queda és m; agafem la moneda més gran que no sigui més gran que m; restem el valor d'aquesta moneda de m, i repetim.

Aquesta estratègia resol el problema plantejat?

88. El Rector de la UPC de cara a fomentar la germanor entre el personal de la UPC ha decidit establir activitats mensuals per incentivar la interacció entre els diferents estaments de PDI i PAS. Les festes se celebraran normalment un cop al mes. Suposem que hi han k grups disjunts en el personal, $C_1, \ldots, C_k$. El rectorat produeix una llista L d'activitats que tindran lloc en el curs acadèmic, i per a cada activitat $a \in L$, hi ha un nombre màxim $M(a)$ i un nombre mínim $m(a)$ de gent que pot assistir a a. A més, com que els grups tenen diferents nivells d'influència, el rectorat estima, per a cada j, el nombre mínim $s(j)$ de persones del grup C_j que s'han de convidar a cada activitat. Per una altra banda és ben conegut que alguns membres de la UPC no poden participar sovint en activitats lúdiques. Tenint en compte aquest aspecte el rectorat ha determinat per cada membre i del personal de la UPC un valor $t(i)$ indicant el nombre màxim d'activitats a les què i pot participar al llarg del curs. El problema consisteix en, en cas que sigui possible, decidir a qui convidar a cada activitat de manera que es compleixin les restriccions prèvies, o sigui:

- el nombre d'assistents a l'activitat a és $\leq M(a)$ i $\geq m(a)$,
- per a cada grup j hi ha d'haver com a mínim $s(j)$ persones assistint a cada activitat.
- la persona i mai es convidada a més de $t(i)$ activitats.

Proporcioneu un algorisme de cost polinòmic per aquest problema.

89. Es vol emmagatzemar en un dispositiu de capacitat L un conjunt de n arxius. La talla del arxiu i-èsim ve donada per ℓ_i. Se sap que $\sum_{i=1}^{n} \ell_i > L$. Dissenyeu un algorisme que seleccioni un conjunt d'arxius tot maximitzant el nombre d'arxius que es poden emmagatzemar al dispositiu. Justifiqueu la correctesa del vostre algorisme i analitzeu el seu cost.

90. Suposeu que hi han n països que comercien entre ells. Per a cada país i, coneixem el valor del seu superàvit pressupostari s_i, (aquest nombre pot ser positiu o negatiu, un nombre negatiu indica un dèficit). Per a cada parell de països i, j coneixem $e_{i,j} \geq 0$, el valor total de totes les exportacions d'i cap a j, aquest nombre és sempre no negatiu. Diem que un subconjunt S dels països és *autosuficient* si la suma dels superàvits pressupostaris dels països a S, menys el valor total de tots els exportacions dels països a S cap a els països que no són a S, és més gran que zero.

    Doneu un algorisme polinòmic, que a partir d'aquestes dades per n països, decideix si existeix un subconjunt **no buit**, que és autosuficient.

91. Considereu un examen amb n qüestions, on la qüestió i-èsima té un valor de $v_i > 0$ punts i es necessiten m_i minuts per a resoldre-la. Necessitem un total de V punts per aprovar. Dissenyeu un algorisme que donats $v_1, v_2, \dots, v_n$ i $m_1, m_2, \dots m_n$ compute el mínim temps per a obtenir com a mínim V punts. En particular,

   (a) Si $M(i, v)$ denota el mínim nombre de minuts necessaris per obtenir v punts, quan únicament teniu qüestions $1, \cdots, i$, doneu la recurrència per a $M(i, v)$.

   (b) Dissenyeu un algorisme per a resoldre el problema en temps $O(nV)$. Quina n'és la complexitat? Digueu com a partir del vostre algorisme previ, podeu escriure la llista $S \subseteq \{1, 2, \dots, n\}$ de qüestions a resoldre tal que $V \leq \sum_{i \in S} v_i$ i $\sum_{i \in S} r_i$ es minimitza.

   (c) Suposem que pressionat per l'autoritat acadèmica, el professor decideix incrementar els aprovats donant punts parcials a cada pregunta, on la puntuació és proporcional als minuts que l'alumne ha escrit, de la següent manera, a la pregunta i-èsima per cada minut d'escriptura, l'alumne obté v_j/m_j punts (fins a un màxim de v_i punts). Dissenyeu un algorisme per a determinar en quines qüestions heu d treballar i quan de temps heu de treballar a cadascuna, per a obtenir més de V punts. Demostreu la correctesa i doneu la complexitat.

92. La firma *Doctors on Call* té que resoldre el següent problema. Per a cadascun dels pròxims n dies, la firma ha determinat el nombre de doctors disponibles que requereix. Així al dia i-èsim, necessiten exactament p_i doctors. Hi han k doctors en total, i cadascú d'ells ha donat una llista amb els dies en que està disposat a treballar. Així el doctor j proporciona un conjunt L_j de dies. Doctors on Call vol, a partir d'aquesta informació, un procediment que permeti tornar a cada doctor j una llista definitiva de dies L'_j amb les propietats següents: (1) el conjunto $\Delta_j = L'_j \setminus L_j$ té com a molt c dies; i (2) quan es considera tot el conjunto de listas $L'_1, \ldots, L'_k$, per a cada dia $1 \leq i \leq n$, hi han exactament p_i doctors que tenen el dia i a la seva llista definitiva. El paràmetre c reflecteix la tolerància de l'assignació i pot variar segons las circumstancies. Per suposat, si tal solució no es possible, el sistema ha de (correctament) informar de que aquest és el cas.

   (a) Proporcioneu un algorisme de cost polinòmic en n i k que resolgui el problema per un valor de tolerància c donat.

   (b) Proporcioneu un algorisme de cost polinòmic que obtingui la tolerància mínima per que el problema tingui solució, assumint que $p_i \leq k$ per $1 \leq i \leq n$.

93. Considereu el joc de cartes següent : Es posen una seqüència de cartes, $a_1, \ldots, a_n$ cara amunt (on n és parell). La carta a_i té un valor $v_i \in \mathbb{N}$. La seqüència està desordenada El joc consisteix a que cada jugador escull per torns una de les dues cartes possibles; la primera o la darrera de la seqüència. Guanya el jugador que aconsegueix que el valor de les seves cartes sigui el més gran. Si únicament hi juguen 2 jugadors:

   (a) Doneu una seqüència on l'estratègia del que juga primer no és escollir la carta amb valor més alt.

   (b) Doneu un algorisme amb el millor temps que pugueu, que computi una estratègia òptima per a la persona que juga primer (estratègia que li faci guanyar o empatar si no és pot guanyar) (Ajut: Definiu un sub-joc, com el joc que ens queda per jugar, quan s'han escollit i cartes per l'esquerra i j cartes per la dreta $(i + j \leq n)$).

   (c) Quina és la complexitat del vostre algorisme?

94. Les xarxes ad-hoc, composades per dispositius sense fils de baixa potència, s'han proposat per situacions com els desastres naturals en què els coordinadors dels treballs de rescat podrien controlar les condicions en zones de difícil accés. La idea és que una gran col·lecció d'aquests dispositius sense fils es podria llançar des d'un avió en una regió per a continuació reconfigurar-se com una xarxa operativa.
Estem parlant de: (a) dispositius relativament barats, el quals (b) es llancen des d'un avió a (c) un territori perillós; i per la combinació de (a), (b) i (c), es fa necessari fer front a la fallida d'un nombre raonable dels dispositius. Cada dispositiu té un abast de transmissió limitat, pot comunicar-se amb altres dispositius que es troben com a màxim a r metres d'ell.

    Ens agradaria poder determinar si hi han suficients camins que no comparteixen arcs entre els dispositius de dues zones diferents, A i B, a fi de garantir que un senyal d'emergència pot arribar sense problemes des d'algun dispositiu a A a algun dels dispositius a B. Per tal d'evitar interferències volem, a més, controlar el *nivell de participació* dels dispositius. Aquest nivell és el màxim nombre de camins en què un dispositiu hi participa.

    Suposeu que després del llançament podem obtenir les coordenades $p_i = (x_i, y_i)$ dels n dispositius operatius que formen la xarxa inicial. Suposeu també, que les regions A i B són rectangles i que us donen les coordenades de les seves cantonades. Finalment, us donen també un valor c que determina el nombre de camins mínim que voldríem tenir entre A i B.

    - Dissenyeu un algorisme per determinar si és possible trobar c camins que no comparteixin arestes entre A i B.
    - Donat $d > 0$, dissenyeu un algorisme per determinar si hi han c camins que no comparteixen arestes entre A i B de manera que el nivell de participació de cada node és, com a molt, d.
    - Dissenyeu un algorisme per determinar el mínim nivell de participació que permeti trobar c camins que no comparteixin arestes entre A i B amb aquest nivell màxim de participació dels nodes.

95. Considereu l'enunciat del problema 88 en el cas en que els k grups en el personal, $C_1, \ldots, C_k$, no son necessàriament disjunts. En aquesta situació, per a cada activitat, rectorat envia les invitacions a les seus dels grups i els grups seleccionen els membres que els representin a l'activitat. Sota la hipòtesi que üna persona només pot representar un grup a una activitat"doneu un algorisme per decidir (si es pot) a qui convidar per tal de mantenir les restriccions establertes. És a dir,

- el nombre d'assistents a l'activitat a és $\leq M(a)$ i $\geq m(a)$,
- per a cada grup j hi ha d'haver com a mínim $s(j)$ persones assistint a cada activitat.
- la persona i mai es convidada a més de $t(i)$ activitats.

96. SuperFast és una empresa de transport que està intentant decidir si li interessa participar en una oferta de treball de Amazon a Barcelona. Amazon voldria subcontractar el transport de productes de proximitat a SuperFast. El transport ha de garantir uns compromisos de puntualitat molt estrictes i SuperFast vol una estimació de la quantitat de nous vehicles que hauria d'incorporar a la seva flota i una estimació del cost corresponent al seu ús. Amazon li ha proporcionat els resultats de les seves simulacions de comportament dels clients i, en particular, de les necessitats de transport diàries per uns quants dies. Per un dia es disposa d'una llista de sol·licituds de transport. Cada sol·licitud de transport especifica les coordenades GPS de l'origen i del destí d'un enviament juntament amb l'hora de recollida i d'entrega. SuperFast disposa d'un programa que li proporciona el temps necessari de desplaçament d'un vehicle entre qualsevol parell de posicions de la ciutat coneixent el temps d'inici del trasllat.

    En aquest primer estudi SuperFast assumeix el cas pitjor en el què els vehicles no poden portar més d'un enviament. A més, el vehicle ha de ser a la posició d'origen al temps estipulat i no pot deixar el punt de destí fins el temps estipulat de trasllat. Així, un vehicle que transporti un enviament pot encarregar-se d'un altre sempre que el temps de desplaçament entre el destí i el nou origen li permeti arribar l'hora estipulada. També assumeix que l'estimació del temps necessari de desplaçament és acurada.

    Disenyeu algorismes amb cost polinòmic que:

    (a) Donats un nombre k i les sol·licituds de transport d'un dia, determini si es poden servir amb k vehicles.
    (b) Donades les sol·licituds de transports d'un dia, determini el nombre mínim de vehicles que les poden servir ($k_{\min}$)
    (c) Donades les sol·licituds de transports de un dia, proporcioni una planificació per a cadascun dels $k_{\min}$ vehicles indicant, per a cada vehicle, la seqüència de sol·licituds de transport que ha de servir i el temps total de desplaçament del vehicle.

    Podeu suposar que el càlcul del temps de desplaçament entre dos posicions té un temps constant i que entre l'hora de recollida i la d'entrega hi ha temps suficient per fer-hi el desplaçament amb puntualitat.

    Ajut: Podeu pensar un vehicle com una unitat de flux i una sol·licitud de transport com un arc amb fites inferiors i superiors a la seva capacitat.

    Una solució

    (a) Utilizando la ayuda identificaremos los vehiculos con unidades de flujo y las solicitudes con arcos con capacidad limitada con cota superior e inferior.

        De las restricciones del problema sabemos que un vehículo puede servir otro envío siempre que pueda desplazarse con tiempo suficiente desde el destino al nuevo origen, asumiendo que permanece en el destino hasta la hora estipulada. Esta restricción nos indica cual es la red a considerar.

        Tendremos, para cada solicitud i, dos nodos o_i y d_i y un arco (o_i, d_i) con capacidad $[1, 1]$, reflejando el hecho de que todas las solicitudes tienen que ser servidas. Añadiremos un arco (d_i, o_j) cuando saliendo del destino de la solicitud i en el tiempo estipulado podamos llegar al origen de la solicitud j en el tiempo estipulado, reflejando la observación previa.

        Añadiremos tres vertices adicionales s, x, t y los arcos (s, x) con capacidad k, para forzar que el flujo máximo sea $\leq k$, i, para cada solicitud i, los arcos (x, o_i) i (d_i, t) con capacidad 1.

        Notemos que la existencia de un flujo que satisface las restricciones de capacidad nos garantiza la existencia de $\leq k$ caminos que no comparten aristas de s a t que, además, cubren todos los arcos correspondientes a solicitudes. Esto es equivalente a poder servir las peticiones con k o menos vehículos.

        Una vez obtenida la red de flujo, podemos determinar si hay un flujo que satisface las restricciones en tiempo polinómico utilizando el algoritmo visto en clase para decidir la existencia de flujos con demandas y cotas inferiores.

(b) Siempre hay una solución con n vehículos, por ello $1 \leq k_{\min} \leq n$ podemos implementar una búsqueda binaria utilizando el algoritmo del apartado (a) que nos permite decidir si las solicitudes se pueden servir con $\leq k$. En total tendermos que hacer $O(\log n)$ ejecuciones del algoritmo del apartado (a).

(c) Una vez encontrado el valor de $k_{\min}$ utilizando el algoritmo del apartado (a) para este valor de k tendremos una flujo de s a t con valor del flujo $k_{\min}$ que satisface la restricción de que todas las solicitudes están servidas. Como hemos visto en clase lo único que tenemos que hacer es aplicar el algoritmo de extracción de caminos disjuntos como vimos en clae. Esto nos da $k_{\min}$ ejecuciones de BFS sobre el grafo formado por los arcos con flujo positivo.

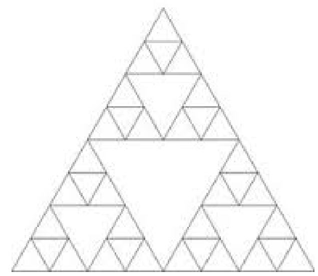
97. Considereu un digraf $G = (V, E)$ en el que cada arc $e \in E$ té associat una capacitat $u_e \in \mathbb{R}^+$ i un cost $c_e \in \mathbb{R}$. El cost s'interpreta com el cost de transportar una unitat al llarg de l'arc. Cada node $v \in V$ té associada una demanda $b_v \in \mathbb{R}$. Si $b_v \geq 0$, v és un sumider, si no és una font. Assumim que la suma de totes les demandes es 0. Considereu el problema de cobrir la demanda de les fonts, a partir de l'oferta dels sumiders mitjançant un flux vàlid amb el mínim cost possible. Doneu una formalització d'aquest problema com programació lineal i calculeu el seu dual.

98. Considereu els enunciats dels problemes 88 y 95 doneu una formalització, com a problema de programació entera del problema de decidir (si es pot) a qui convidar per tal de mantenir les restriccions establertes. És a dir,

- el nombre d'assistents a l'activitat a és $\leq M(a)$ i $\geq m(a)$,
- per a cada grup j hi ha d'haver com a mínim $s(j)$ persones assistint a cada activitat.
- la persona i mai es convidada a més de $t(i)$ activitats.

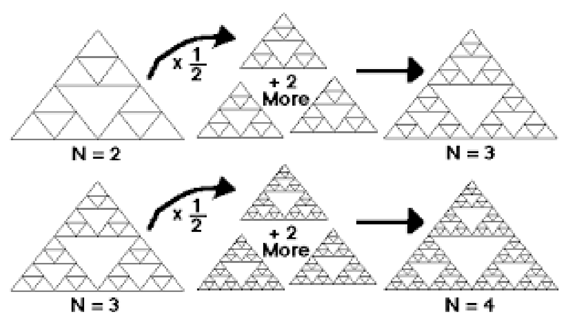
Compareu la complexitat de les dues solucions.

99. El *triangle de Sierpinski* és un graf com el de la figura de sota i és un exemple d'estructura fractal, on cada triangle de costat ℓ consta de 3 triangles de costat $\ell/2$ i en total té n vèrtexs. (A la figura de sota $\ell = 8$ i $n = 42$.)



(a) Demostreu que asimptòticament el nombre de vèrtexs n d'un triangle de Sierpinski de costat ℓ és $n = \Theta(\ell^{\lg 3})$.

(b) Dissenyeu una ED per emmagatzemar triangle de Sierpinski. L'estructura ha de fer servir, per una part, un arbre ternari representant la creació recursiva d'un triangle de Sierpinski de N nivells tal i com es descriu a la següent figura.



A més l'estructura de dades ha de mantenir el graf subjacent com una llista de veïns per node.

Les operacions bàsiques per aquesta ED són: crear un triangle (N=0) i crear un triangle de Sierpinski amb N nivells a partir de 3 triangles de Sierpinski de $N-1$ nivells. Proporcioneu a més el cost de les operacions bàsiques.

(c) Assumint que ens donen com entrada un triangle de sierpinski amb N nivells on cada vertex v té associat un pes $w(v)$, dissenyeu un algorisme eficient per el problema del *Max-weighted independent set*. El vostre algorisme hauria de fer servir l'ED del apartat anterior.

Recordeu que donat un graf $G = (V, E)$ amb pesos $w : V \to \mathbb{Z}^+$, el *Max-weighted independent set* consisteix a trobar un conjunt independent amb pes màxim.

100. Donat un graf $G = (V, E)$, volem obtenir un recobriment de vèrtexs mínim, és a dir un conjunt $S \subseteq V$, amb mínim nombre de vèrtexs, tal que qualsevol aresta a G té al menys un extrem a S.

(a) Doneu una formalització com a problema de programació entera (IP-VC). En aquest cas heu de fer servir les variables $x_1, \dots, x_n$ que han de tenir valors 0 o 1 indicant si el vèrtex i és o no a S. Completeu la resta de restriccions lineals i la funció a minimitzar.

(b) Considereu el problema de programació lineal en la que la condició $x_i \in \{0,1\}$ es substitueix per $x_i \in [0,1]$ (LP-VC). Sigui $y = (y_1, \dots, y_n)$ una solució optima de LP-VC.

Definim $\epsilon = \min_{y_i \notin \{0,1/2,1\}} \{y_i, |y_i - 1/2|, 1 - y_i\}$, considereu les solucions:

$$y'_i = \begin{cases} y_i - \epsilon & 0 < y_i < 1/2 \\ y_i + \epsilon & 1/2 < y_i < 1 \\ y_i & \text{otherwise} \end{cases} \qquad y''_i = \begin{cases} y_i + \epsilon & 0 < y_i < 1/2 \\ y_i - \epsilon & 1/2 < y_i < 1 \\ y_i & \text{otherwise} \end{cases}$$

Demostreu que y' i y'' son també solucions òptimes i que, a més, si y té alguna coordenada fora de $\{0, 1/2, 1\}$, llavors y' o y" tenen menys coordenades fora de $\{0, 1/2, 1\}$ que y.

(c) Considereu el següent algorisme

function RELAX+ROUND VC(G)
    Construir el LP-VC associat a G
    Obtenir y una solució òptima de LP-VC
    Repetint el procediment del apartat previ
        obtenir una solució optima de LP-VC y' amb $y'_i \in \{0, 1, 1/2\}$
    Obtenir x com $x_i = 0$ si $y'_i = 0$, $x_i = 1$ si no.
    return (x)
end function

Demostreu que Relax+Round VC és una 2-aproximació al problema plantejat amb cost polinòmic.

Una solución

(a) Tal y como hemos visto en clase, lo podemos modelizar por:

$$\begin{array}{ll} & \text{VC-ILP} \\ \min & \sum_{i=1}^{n} c_i x_i \\ \text{s.t.} & x_i + x_j \geq 1 \quad \text{for all } (i,j) \in E \\ & x_i \in \{0,1\} \quad \text{for all } i \in V \end{array}$$

(b) La restricción $x_i \leq 1$ la podemos eliminar debido al tipo de función a minimizar. La relajación como LP nos queda

$$\begin{array}{ll} & \text{VC-LP} \\ \min & \sum_{i=1}^{n} c_i x_i \\ \text{s.t.} & x_i + x_j \geq 1 \quad \text{for all} (i,j) \in E \\ & x_i \geq 0 \quad \text{for all } i \in V \end{array}$$

(c) Para ver que ambas soluciones son factibles hacemos un análisis por casos. Cuando $y_i \in (0, 1/2)$, si tenemos una arista (i, j) al ser y factible $y_i + y_j \geq 1$. Por tanto $y_j \in (1/2, 1]$. Si $y_j = 1$, $y'_i + y'_j, y''_i + y''_j \geq 1$. Si Si $y_j \in (1/2, 1)$, $y'_i + y'_j = y_i - \epsilon + y_j + \epsilon = y_i + y_j \geq 1$ y $y''_i + y''_j = y_i + \epsilon + y_j - \epsilon = y_i + y_j \geq 1$. De forma similar obtenemos el mismo resultado cuando $y_i \in (1/2, 1/)$.

Si sumamos todas las componentes de las dos soluciones tenemos, los valores de ϵ cancelan en cada variable y tenemos::

$$\sum_{i=1}^{n} y'_i + \sum_{i=1}^{n} y''_i = 2\sum_{i=1}^{n} y_i = 2\text{opt}.$$

Sabemos que $\sum_{i=1}^{n} y'_i, \sum_{i=1}^{n} y''_i \geq \text{opt}$ ya que las sos son soluciones factibles. Por lo tanto, tiene que cumplirse que $\sum_{i=1}^{n} y'_i = \sum_{i=1}^{n} y''_i = \text{opt}$.

Si consideramos una componente i que nos porporcione el valor de ϵ. Dependiendo de si $y_i \in (0, 1/2)$ o $(1/2, 1)$ y del valor al que esté a distancia ϵ al sumar/restar, un análisis por casos nos garantiza que que y'_i o y''_i está en $\{0, 1/2, 1\}$. Por lo tanto una de las dos asignaciones tiene menos coordenadas que y en este conjunto de valores.

(d) Relax+Round VC tiene coste polinómico. LP se puede resolver en tiempo polinómico en el número de ecuaciones y este tamaño es $O(n+m)$. EL proceso del paso anterior se puede realizar en tiempo $O(n)$ y como mucho $O(n)$ veces. el coste total es polinómico en el tamaño del grafo.

Además, de acuerdo con la construcción, $\sum_{i=1}^{n} x_i \leq 2\sum_{i=1}^{n} y'_i \leq 2\text{opt}$. por lo que eefectivamente tenemos una 2-aproximación.