Algoritmické riešenia ťažkých problémov
Domáca úloha 1

Autor: Marián Kravec

Úloha 1 - Cool game

a)

Do ILP programu prepíšeme tento problém takto:
Budeme mať tri skupiny binárnych premenných (s možnými hodnotami 0 a 1).

- $\forall 1 \leq i \leq k, g_i$ -> tieto premenné budú označovať či k-ta dvojica vrcholov za ktoré dostávame body je vybraná
- $\forall 1 \leq i \leq n, x_i$ -> tieto premenné hovoria či naša cesta obsahuje i-ty vrchol
- $\forall (i,j) \in E, e_{i,j}$ -> tieto premenné hovoria či hrana medzi i-tym a j-tym vrchol patrí do našej cesty (E je množina hrán grafu)

Náš ILP program bude riešiť takýto maximalizačný problém:
$maximize \sum_{i=1}^{k} 100c_i$

Čiže sa snaží aby bolo čo najviac dvojíc za ktoré dostávame body boli na našej ceste.
Teraz potrebujeme spraviť tú najťažšiu časť a to napísať podmienky pre naše premenné.

Najskôr podmienky prepájajúce premenné pre bonusové dvojice a ich príslušné vrcholy:
$\forall 1 \leq i \leq k, 2c_i - x_{u_i} - x_{v_i} \leq 0$
Tieto podmienky hovoria, že na to aby $c_i = 1$ tak sa musí $x_{u_i} = 1$ aj $x_{v_i} = 1$ inak by výsledok bol menší ako 0, ak $c_i = 0$ tak nám na hodnotách x_{u_i} a x_{v_i} nezáleží, keďže program je maximalizačný (a koeficienty pri c_i sú v maximalizovanej funkcii vždy kladné konkrétne 100) tak ak nastane $x_{u_i} = 1$ a $x_{v_i} = 1$ tak určite nastaví $c_i = 1$.

Teraz potrebujeme prepojiť vrcholy a hrany, na to budeme potrebovať viac typov podmienok tak začnime jednoducho:
$\forall (i,j) \in E, 2e_{i,j} - x_i - x_j \leq 0$
Tieto podmienky hovoria, že aby na cesta mohla byť hrana (i, j), čiže $e_{i,j}$ tak musí platiť, že na ceste sú obe vrcholy, čiže $x_i = 1$ a $x_j = 1$, podobne ani tu nás netrápi prípad keď $e_{i,j} = 0$.

Zatiaľ by náš algoritmus vybral všetky vrcholy (aby splnil všetky bonusové dvojice) ale žiadne hrany preto musí prepojiť hrany a vrcholy viac. To spravíme takýmito podmienkami:
$\forall 2 \leq j \leq n-1, U = \{i|(i,j) \in E\}, x_j - \sum_{i \in U} e_{i,j} = 0$
$V = \{i|(j,i) \in E\}, x_j - \sum_{i \in V} e_{j,i} = 0$
Ide o dvojicu podmienok pre druhý až predposledný vrchol, že na to aby mohlo platiť $x_j = 1$ musí existovať práve jedna hrana idúca do tohto vrcholu (U je množina hrán idúcich do vrchola j) a práve jedna hrana vychádzajúca z tohto vrcholu (V je množina hrán idúcich do vrchola j). Týmto zabezpečíme, že na to aby mohol povedať, že vrchol patrí do cesty musí povedať ktorou cestou do tohto vrcholu vošiel a ktorou vyšiel.

Tieto podmienky sú však iba pre druhý a predposledný vrchol, ešte potrebujeme pridať podmienky pre prvý a posledný. Začnime s prvým:
$x_1 = 1$
$V = \{i|(1,i) \in E\}, \sum_{i \in V} e_{1,i} = 1$
Tieto podmienky hovoria, že vrchol 1 je na ceste (keďže je počiatočný) a vychádza z neho práve jedna hrana. Teraz pre posledný:

$x_n = 1$
$U = \{i|(i,n) \in E\}, \sum_{i \in V} e_{i,n} = 1$
Podobne ako pre prvý tak aj pre posledný je podmienka, že leží na ceste (keďže je cieľ) a vchádza do neho práve jedna hrana.

Keď skombinujeme obe typy podmienok prepájajúce hrany a vrcholy dostaneme výsledok, že na to aby hrana mohla patriť do cesty musia tam patriť obe vrcholy na jej koncoch a na to aby vrchol mohol patriť do ceste tak musí tam patriť práve jedna hrana ktorá do vrcholu vchádza a ktorá vychádza. Keďže je graf acyklický tak tieto podmienky neumožňujú aby žiaden z druhého až predposleného vrcholu nemôže byť na začiatku ani konci cesty (keďže potrebuje vstupnú a výstupnú hranu) preto jediný možný počiatočný vrchol je vrchol 1 a jediný možný konečný vrchol je vrchol n takže algoritmu ak nájde riešenie určite nájde jednu cestu medzi prvým a posledným vrcholom. Pričom ak tých ciest je viac vyberie tú ktorá splní čo najviac bonusových dvojíc.

Čiže výsledný program by mohol vyzerať takto:

$maximize \sum_{i=1}^{k} 100c_i$

$\forall 1 \leq i \leq k, 2c_i - x_{u_i} - x_{v_i} \leq 0$
$\forall (i,j) \in E, 2e_{i,j} - x_i - x_j \leq 0$
$\forall 2 \leq j \leq n-1, U = \{i|(i,j) \in E\}, x_j - \sum_{i \in U} e_{i,j} = 0$
$V = \{i|(j,i) \in E\}, x_j - \sum_{i \in V} e_{j,i} = 0$
$x_1 = 1$
$V = \{i|(1,i) \in E\}, \sum_{i \in V} e_{1,i} = 1$
$x_n = 1$
$U = \{i|(i,n) \in E\}, \sum_{i \in V} e_{i,n} = 1$

$\forall 1 \leq i \leq k, c_i \in \{0,1\}$
$\forall 1 \leq i \leq n, x_i \in \{0,1\}$
$\forall (i,j) \in E, e_{i,j} \in \{0,1\}$

b)

Na tento modifikovaný problém použijeme dynamické programovanie. Využijeme to, že všetky hrany vedú od vrcholov s menšou hodnotou k vrcholom s väčšou. Na začiatok inicializujeme zoznam dĺžky n na nuly potom v cykle od n do 1 pre i-ty vrchol vyberieme meximálnu hodnotu z tabuľky z j-tych pozícii pre ktoré platí, že existuje hrana z i do j a ak je i-ty vrchol bonusový tak pripočítame 100. Ako speudokod použijeme niečo python-like (len pre jednoduchosť budeme indexovať od 1).

```
Bonus = [w_1..w_k]
Hodnoty = [0]*n
Hrany = [0]*n

for u in [n..1]:
        Vhodnota = max([Hodnota[v] | (u, v) ∈ E]+[0])
        Vdalsi = argmax([Hodnota[v] | (u, v) ∈ E]+[0])
        if u ∈ Bonus:
                Hodnoty[u] = 100+Vhodnota
        else:
                Hodnoty[u] = Vhodnota
        Hrany[u] Vdalsi

Cesta = [1]
while Cesta[-1] ≠ n:
        Cesta.append(Hrany[Cesta[-1]])

rerurn Hodnoty[1], Cesta
```

Z pohľadu zložitosti, tento graf potrebuje v konečnom dôsledku pozrieť každú hranu práve raz (respektíve dvakrát) keďže pre každú hranu existuje iba jeden vrchol z ktorého vychádza, takže zložitosť je O(m) kde m je počet hrán grafu (nanajvýš kompletný graf čiže najhoršie $O(n^2)$) (zložitosť môžeme riešiť cez počet hrán aj napriek tomu, že cyklus ide cez vrcholy keďže zložitosť jednej iterácie cyklu závisí od počtu hrán z daného vrcholu čiže v konečnom dôsledku je dôležité koľko je celkovo hrán). Z pohľadu priestorovej zložitosť tento algoritmus potrebuje iba pár zoznamov dĺžky n takže priestorová zložitosť je O(n).

Správnosť algoritmu ukážeme induktívne.

Bázový prípad: začínajúci vrchol je vrchol n -> trivialne ak začínajúci vrchol je n a konečný vrchol je n tak jediná cesta neobsahuje žiadnu hranu a optimálna hodnota je závislá od toho či n patrí medzi bonusové vrcholy.

Indukčný predpoklad: ak začínajúci vrchol je niektorý z k+1 až n poznáme optimálnu cestu a hodnotu

Indukčný krok: chceme optimálnu cestu a hodnotu ak začíname z vrchola k. Táto cesta bude tvorená jednou hránou z vrcholu k do vrcholu väčšieho ako k a zvyšku cesty, v prípade zvyšku cesty vieme, že chceme optimálnu podcestu, ktorú poznáme, takže optimálna cesta z k je cesta do vrcholu l a potom optimálna cesta z vrchola l do n. Optimálna hodnota je optimálna hodnota z l do n plus 100 ak k patrí medzi bonusové.

Teraz sme induktívne ukázali, že náš algoritmus nájde optimálnu pre začínajúci k-ty vrchol, čiže optimálna celá cesta je ak začíname z vrchola 1.