Bonyolultságelmélet gyakorlat – 11
Approximálás II.

A következő problémánk, melynek eldöntési változata (annak ellenére, hogy 2SAT **P**-ben, sőt **NL**-ben van):

Recap: MAX2SAT

- **Input:** egy olyan F CNF, melyben minden klóz pontosan két literált tartalmaz, melyek változói különböznek és egy $K > 0$ egész szám
- **Egy megoldás:** egy értékadás
- **A megoldás haszna:** az általa egyszerre kielégített klózok száma

(ez egy maximalizálási probléma.)

Erre a problémára létezik egy 4/3-approximáló (determinisztikus) algoritmus, ami azért lesz 4/3-approximáló, mert a klózoknak garantáltan kielégíti 3/4-ét (és így az optimumnak is legalább a 3/4-ét persze):

Recap

- Ciklusban értékadunk előbb x_1-nek, majd x_2-nek, ..., végül x_n-nek.
- Az x_i változó értékét a következőképp határozzuk meg: kiszámolunk az $x_i = 0$ és az $x_i = 1$ (parciális) értékadás mellett is egy-egy értéket, és amelyik esetben nagyobbat kaptunk, azt választjuk (egyenlőségnél mindegy).
- Az értéket úgy kapjuk, hogy az x_i-t tartalmazó klózok mindegyikéhez rendelünk egy súlyt, és ezeket a súlyokat összegezzük. Egy klóz súlya
  - 2, ha szerepel benne az x_i változó, és az x_i-t tartalmazó literál értékét a vizsgált értékadás 1-re állítaná;
  - 1, ha szerepel benne az x_i változó, mellette egy másik, és az x_i-t tartalmazó literál értékét a vizsgált értékadás 0-ra állítaná;
  - 0 különben (azaz ha a klózban csak egyetlen literál szerepel, ennek változója x_i, és a vizsgált értékadás a literált hamisra állítaná).
- Miután beállítottuk a fenti súlyok összegét maximalizáló értékre x_i-t:
  - azokat a klózokat, melyek így igazzá válnak, töröljük a CNF-ből;
  - azokat a klózokat, melyek így biztosan hamissá válnak, szintén töröljük a CNF-ből;
  - a maradék klózokból pedig töröljük az x_i-t tartalmazó literált (ha van benne).

1. feladat.

Hajtsuk végre a 4/3-approximáló algoritmusunkat a következő 2CNF-en:

$$(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2) \wedge (x_1 \vee x_3) \wedge (x_2 \vee \neg x_3) \wedge (\neg x_2 \vee \neg x_4) \wedge (x_2 \vee \neg x_5) \wedge$$
$$(\neg x_2 \vee x_5) \wedge (x_2 \vee x_5) \wedge (\neg x_2 \vee \neg x_5) \wedge (\neg x_3 \vee x_4) \wedge (x_3 \vee \neg x_4) \wedge (\neg x_3 \vee \neg x_4)$$

Az utolsó approximációs feladattípusunk a metrikus TSP:

Recap: Metrikus TSP

- **Input:** N város és a köztük lévő távolságok $d_{i,j}$ mátrixa, ami teljesíti a következőket:
  - $d_{i,i} = 0$ minden i-re,
  - szimmetrikus: $d_{i,j} = d_{j,i}$ minden i-re és j-re;
  - háromszög-egyenlőtlenség: $d_{i,j} + d_{j,k} \leq d_{i,k}$ minden i-re, j-re és k-ra.
- **Egy megoldás:** a csúcsoknak egy tetszőleges sorrendje (melyben minden csúcs pontosan egyszer szerepel).
- **Egy megoldás költsége**: a csúcsok megadott sorrendje (mint kör) szerint megadott össz távolság, azaz ha a megoldás $i_1, \ldots, i_N$, akkor

$$d_{N,1} + \sum_{j=1}^{N} d_{i_{j-1},i_j}.$$

(Ez persze egy minimalizálási probléma.)

Recap

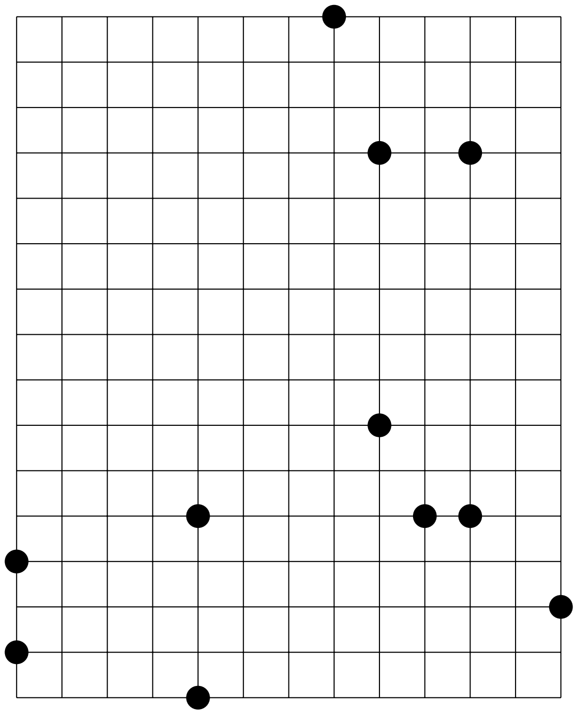
A Metrikus TSP problémára egy 2-approximáló algoritmus:

- Határozzuk meg (pl. Prim algoritmusával, ld. algoritmusok és adatszerkezetek kurzus vagy wiki) a távolságmátrix által megadott teljes gráfnak egy minimális feszítőfáját
- Ezt a fát járjuk be egy tetszőlegesen választott pontból mélységi bejárással, a csúcsok visszaadott sorrendjét az elérési idejük (amikor először láttuk őket) határozza meg.

Ezt „rajzban” könnyű úgy megadni, hogy „körberajzoljuk a fát”, ha egy 2D rácson kapjuk meg a pontokat az euklideszi távolságuk szerint, és minden csúcsot akkor veszünk fel, mikor „először látjuk”.

2. feladat.

Hajtsuk végre a tanult 2-approximáló algoritmust a következő METRIKUS TSP inputon, ahol a városok a jelölt rácspontok, távolságuk pedig az euklideszi („vonalzós”) távolság:



1. feladat megoldása.

Először x_1-nek adunk értéket, az őt tartalmazó klózok: $(x_1 \vee \neg x_2)$, $(\neg x_1 \vee x_2)$, $(x_1 \vee x_3)$.

- Ha $x_1 = 0$, akkor
  - az első klóz még nem elégül ki, de van benne másik változó, így súlya ekkor 1
  - a második klóz kielégül, súlya tehát ekkor 2
  - a harmadik klóz még nem elégül ki, de van benne másik változó, így súlya ekkor 1
  - tehát $x_1 = 0$ esetén az összsúly 4.
- Ha $x_1 = 1$, akkor
  - az első klóz kielégül, tehát súlya ekkor 2
  - a második nem, de van benne másik változó, tehát súlya ekkor 1
  - a harmadik kielégül, tehát súlya ekkor 2
  - tehát $x_1 = 1$ esetén az összsúly 5.

Mivel $5 > 4$, így az $x_1 = 1$-et választjuk és töröljük az így garantáltan igazzá / hamissá váló klózokat, a maradék klózokból pedig az x_1-et tartalmazó literálokat. Ezt kapjuk:

$$x_2 \wedge (x_2 \vee \neg x_3) \wedge (\neg x_2 \vee \neg x_4) \wedge (x_2 \vee \neg x_5) \wedge$$
$$(\neg x_2 \vee x_5) \wedge (x_2 \vee x_5) \wedge (\neg x_2 \vee \neg x_5) \wedge (\neg x_3 \vee x_4) \wedge (x_3 \vee \neg x_4) \wedge (\neg x_3 \vee \neg x_4)$$

Most x_2-nek adunk értéket, az őt tartalmazó klózok: x_2, $(x_2 \vee \neg x_3)$, $(\neg x_2 \vee \neg x_4)$, $(x_2 \vee \neg x_5)$, $(\neg x_2 \vee x_5)$, $(x_2 \vee x_5)$ és $(\neg x_2 \vee \neg x_5)$.

- Ha $x_2 = 0$, akkor közülük
  - kielégül: $(\neg x_2 \vee \neg x_4)$, $(\neg x_2 \vee x_5)$, $(\neg x_2 \vee \neg x_5)$, nekik a súlyuk 2, ez eddig összesen 6
  - nem elégül ki, de marad még benne változó: $(x_2 \vee \neg x_3)$, $(x_2 \vee \neg x_5)$, $(x_2 \vee x_5)$, nekik a súlyuk 1, ez még összesen 3
  - nem elégül ki és nem marad benne változó: x_2, neki a súlya 0
  - tehát ekkor az összsúly 9.
- Ha $x_2 = 1$, akkor:
  - kielégül: x_2, $(x_2 \vee \neg x_3)$, $(x_2 \vee \neg x_5)$, $(x_2 \vee x_5)$, nekik a súlyuk 2, ez eddig összesen 8
  - nem elégül ki, de marad még benne változó: $(\neg x_2 \vee \neg x_4)$, $(\neg x_2 \vee x_5)$, $(\neg x_2 \vee \neg x_5)$, nekik a súlyuk 1, ez összesen 3
  - nem elégül ki és nem marad benne változó: ilyen most nincs
  - tehát ekkor az összsúly 11.

Mivel $11 > 8$, így az $x_2 = 1$-et választjuk és töröljük az így garantáltan igazzá / hamissá váló klózokat, a maradék klózokból pedig az x_1-et tartalmazó literálokat.

Ezt kapjuk:

$$(\neg x_4) \wedge x_5 \wedge (\neg x_5) \wedge (\neg x_3 \vee x_4) \wedge (x_3 \vee \neg x_4) \wedge (\neg x_3 \vee \neg x_4)$$

Most x_3-nak adunk értéket, az őt tartalmazó klózok: $(\neg x_3 \vee x_4)$, $(x_3 \vee \neg x_4)$, $(\neg x_3 \vee \neg x_4)$.

- Ha $x_3 = 0$, akkor közülük
  - kielégül: a szélső kettő, összes súlyuk $2 \times 2 = 4$,
  - nem elégül ki, de marad benne változó: a középső, súlya 1,
  - nem elégül ki, nem marad benne változó: ilyen nincs,
  - tehát ekkor az összsúly 5.
- Ha $x_3 = 1$, akkor közülük
  - kielégül: a középső, súlya 2,
  - nem elégül ki, de marad benne változó: a szélső kettő, súlyuk összesen $2 \times 1 = 2$,
  - nem elégül ki, nem marad benne változó: ilyen nincs,
  - ekkor tehát az összsúly 4.

Mivel $5 > 4$, most az $x_3 = 0$-t választjuk és töröljük az így garantáltan igazzá / hamissá váló klózokat, a maradék klózokból pedig az x_1-et tartalmazó literálokat. Ezt kapjuk:

$$\neg x_4 \ \wedge \ x_5 \wedge \ \neg x_5 \ \wedge \ \neg x_4.$$

(Note: vegyük észre, hogy itt NEM hagyjuk el az ismétlődő klózokat!) Most x_4-nek adunk értéket, az őt tartalmazó klózok $\neg x_4$ és $\neg x_4$.

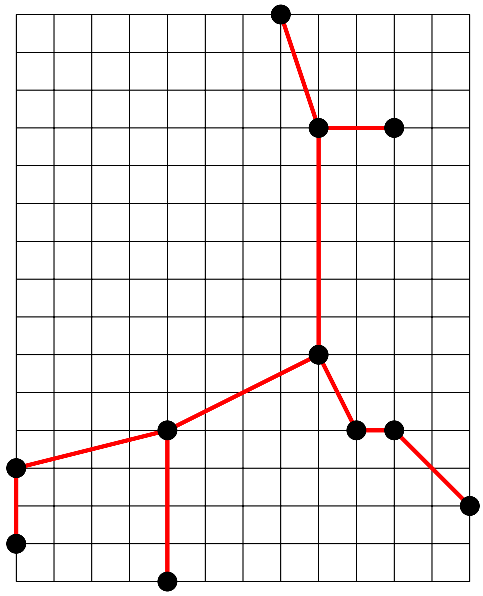
- Ha $x_4 = 0$, akkor az összsúly $2 \times 2 = 4$;
- ha $x_4 = 1$, akkor az összsúly $2 \times 0 = 0$,
- ezért az $x_4 = 0$-t választjuk.

Marad: $x_5 \wedge \neg x_5$. Ezután x_5-nek is adunk értéket, mindkét esetben 1 lesz az összsúly, mindegy, melyiket választjuk, legyen mondjuk $x_5 = 0$.

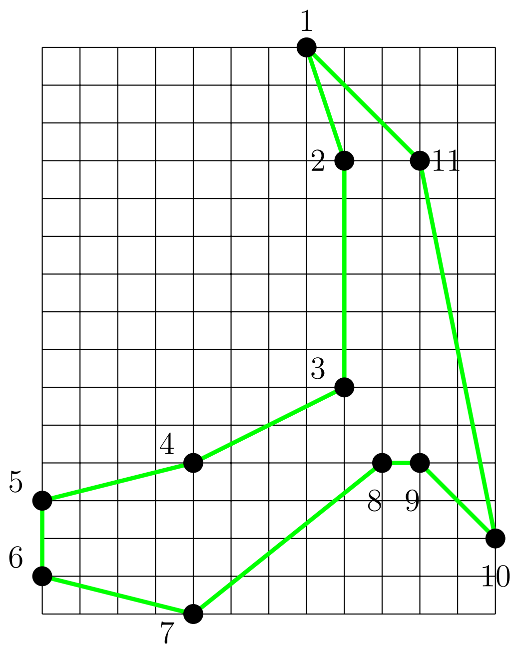
Tehát az étékadás, amit kaptunk: $x_1 = 1$, $x_2 = 1$, $x_3 = 0$, $x_4 = 0$, $x_5 = 0$, ami az eredeti formula klózaiból kielégít 11-et (nem elégíti ki az eredeti 12 formulából a $(\neg x_2 \vee x_5)$ klózt), ami vagy a maximum, vagy nem, de garantáltan a klózok 3/4-e.

2. feladat megoldása.

Ezen pl a $(7, 15)$ pontból indítva a Prim algoritmust (mely mindig a még bekötetlen csúcsok közül a legrövidebb éllel beköthetőt választja és jelöli meg bekötöttként) a következő feszítőfát (piros) kapjuk:



amiből a visszaadott körút, ha megint a $(7, 15)$-ből mint gyökérből indítjuk a bejárást (ha a „rajzos” módszerben gondolkodunk, akkor „balra indulva”):



Ez a körút tehát legfeljebb kétszer olyan hosszú, mint az optimális.