Approksimointialgoritmit (kevät 2010)
Harjoitus 11 (26. huhtikuuta)
Ratkaisuja (Jouni Siren)

1. Laitostensijoitteluongelman approksimointialgoritmin vaiheessa 2 valitaan väliaikaisesti avattujen laitosten muodostamasta verkosta H maksimaalinen riippumaton joukko I, jonka laitokset avataan. Verkossa H on kaari laitosten i ja i' välillä, jos jokin kaupunki j on maksanut laitosten i ja i' avaamisesta eli kaaret (i, j) ja (i', j) ovat erityisiä alkuperäisessä verkossa.

   Muutetaan algoritmia niin, että laitosten i ja i' välille tulee kaari, jos jokin kaupunki j on maksanut niin paljon, että se voitaisiin yhdistää kumpaan tahansa laitoksista i ja i'. Tällöin sanotaan, että alkuperäisen verkon kaaret (i, j) ja (i', j) ovat tiukkoja. Osoitetaan, että nyt luentojen lemma 3.76 ei enää päde, vaan laitokseen i yhdistetyllä kaupungilla j voi olla $c_{ij} > 3\alpha_j^e$, missä α_j^e on yhdistämiseen käytetty osuus kaupungin j maksamasta hinnasta.

   Tarkastellaan verkkoa, jossa on laitokset i ja i' sekä kaupungit j, j' ja k. Kummankin laitoksen avauskustannus on $f_1 = f_2 = 1$, ja yhdistämiskustannuksille pätee $c_{i'j} = 1$, $c_{ij'} = c_{i'j'} = 100$ ja $c_{ik} = 100$. Muut yhdistämiskustannukset saadaan lyhimmistä poluista. Algoritmi etenee nyt seuraavasti:

t	α_j	$\alpha_{j'}$	α_k	β_i	$\beta_{i'}$	Tapahtumat
0	0	0	0	0	0	
1	1	1	1	0	0	(i', j) tiukka
2	**2**	2	2	0	**1**	(i', j) erityinen
100	**2**	**100**	100	0	**1**	$(i, j'), (i', j')$ ja (i, k) tiukkoja
101	**2**	**100**	**101**	**1**	**1**	(i, k) erityinen

   Lihavoitu arvo tarkoittaa, että kyseinen kaupunki on yhdistetty tai laitos on väliaikaisesti avattu.

   Vaiheessa 1 avattiin siis molemmat laitokset väliaikaisesti. Verkkoon H tulee niiden välille kaari, sillä kaaret (i, j') ja (i', j') ovat tiukkoja. Jos riippumattomaksi joukoksi valitaan $\{i\}$, yhdistetään kaikki kaupungit laitokseen i. Koska $c_{ij} = 201$, pätee sille $c_{ij} > 3\alpha_j^e = 6$.

   Ongelma voidaan korjata numeroimalla laitokset sen mukaan, missä järjestyksessä ne avattiin, ja valitsemalla verkon H maksimaaliseksi riippumattomaksi joukoksi niistä leksikografisesti ensimmäinen. Lisäksi jos kaupunki j joudutaan yhdistämään epäsuorasti johonkin sen yhdistystodisteen i' riippumattomassa joukossa I olevaan naapuriin verkossa H, valitaan näistä naapureista järjestyksessä ensimmäinen. Olkoon tämä laitos i. Koska H on leksikografisesti ensimmäinen maksimaalinen riippumaton joukko, seuraa tästä $i < i'$.

   On syytä huomata, että vaikka monta asiaa saattaa tapahtua hetkellä t, on näillä tapahtumilla aina jokin järjestys. Jos siis jollain kaupungilla j muuttuisi samanaikaisesti monta kaarta tiukoiksi, ei näitä muuttumisia kuitenkaan enää käsitellä sen jälkeen, kun jokin tiukaksi muuttunut kaari (i, j) yhdistää kaupungin väliaikaisesti avoimeen laitokseen i.

   Tarkastellaan kaupunkia j, joka on yhdistetty epäsuorasti laitokseen i ja jonka yhdistystodiste on laitos i'. Laitosten i ja i' välillä on kaari verkossa H, joten kaaret (i, j') ja (i', j') ovat tiukkoja jollain kaupungilla j'. Koska laitos i' on kaupungin j yhdistystodiste, pätee selvästi $\alpha_j \geq c_{i'j}$.

   Oletetaan, että hetkellä t laitos i on avautunut väliaikaisesti ja kaupunki j' on liittynyt siihen. Koska myös kaari (i', j') on tiukka, täytyy sen olla muuttunut tiukaksi ennen hetkeä t, eli $c_{i'j'} \leq t$. Jos kaari (i, j') muuttui tiukaksi hetkellä t, ei laitos i' voinut olla vielä auki. Jos taas laitos i avautui hetkellä t, ei laitos i' voinut vielä auki, koska $i < i'$. Laitos i' avautuu siis vasta hetken t jälkeen. Koska α_j kasvaa, kunnes laitos i' avautuu, täytyy olla $\alpha_j \geq t \geq c_{i'j'}$.

   Toisaalta $\alpha_{j'}$ lakkaa kasvamasta viimeistään hetkellä t, mistä seuraa $\alpha_j \geq t \geq \alpha_{j'} \geq c_{ij'}$. Näin ollen lemma 3.76 on taas voimassa.

2. Esitetään tuotantolaitosten sijoitteluongelmaa vastaava kokonaislukuohjelma. Oletetaan, että jokaisella tuotantolaitoksella $i \in F$ on kapasiteetti u_i ja että kukin laitos voidaan avata useita kertoja.

$$
\begin{array}{lll}
\text{minimoi} & \displaystyle\sum_{i\in F, j\in C} c_{ij}x_{ij} + \sum_{i\in F} f_i y_i & \\
\text{ehdoilla} & \displaystyle\sum_{i\in F} x_{ij} \geq 1 & \text{kaikilla } j \in C \\
& y_i - x_{ij} \geq 0 & \text{kaikilla } i \in F \text{ ja } j \in C \\
& u_i y_i - \displaystyle\sum_{j\in C} x_{ij} \geq 0 & \text{kaikilla } i \in F \\
& x_{ij} \in \{0,1\} & \text{kaikilla } i \in F \text{ ja } j \in C \\
& y_i \in \mathbb{N} & \text{kaikilla } i \in F
\end{array}
$$

Lineaarisessa löysennöksessä on luonnollisesti $x_{ij} \geq 0$ ja $y_i \geq 0$. Duaaliksi tulee tällöin:

$$
\begin{array}{lll}
\text{maksimoi} & \displaystyle\sum_{j\in C} \alpha_j & \\
\text{ehdoilla} & \alpha_j - \beta_{ij} - \gamma_i \leq c_{ij} & \text{kaikilla } i \in F \text{ ja } j \in C \\
& u_i\gamma_i + \displaystyle\sum_{j\in C} \beta_{ij} \leq f_i & \text{kaikilla } i \in F \\
& \alpha_j \geq 0 & \text{kaikilla } j \in C \\
& \beta_{ij} \geq 0 & \text{kaikilla } i \in F \text{ ja } j \in C \\
& \gamma_i \geq 0 & \text{kaikilla } i \in F
\end{array}
$$

Kiinnitetään duaalissa $\gamma_i = 3f_i/4u_i$ jolloin sitä vastaava primaali kuvaa tuotantolaitosten sijoitteluongelmaa ilman kapasiteetteja:

$$
\begin{array}{lll}
\text{minimoi} & \displaystyle\sum_{i\in F, j\in C} \left(c_{ij} + \frac{3f_i}{4u_i}\right) x_{ij} + \sum_{i\in F} \frac{f_i}{4} Y_i & \\
\text{ehdoilla} & \displaystyle\sum_{i\in F} x_{ij} \geq 1 & \text{kaikilla } j \in C \\
& Y_i - x_{ij} \geq 0 & \text{kaikilla } i \in F \text{ ja } j \in C \\
& x_{ij} \geq 0 & \text{kaikilla } i \in F \text{ ja } j \in C \\
& Y_i \geq 0 & \text{kaikilla } i \in F
\end{array}
$$

Kolmioyhtälö on edelleen voimassa, joten ongelma voidaan ratkaista luennoilla (sivut 330–349) esitetyllä 3-approksimointialgoritmilla. Näin saadaan kokonaislukuratkaisu (x, Y), jonka kustannukselle pätee

$$
\sum_{i\in F, j\in C} \left(c_{ij} + \frac{3f_i}{4u_i}\right) x_{ij} + 3\sum_{i\in F} \frac{f_i}{4} Y_i \leq 3\sum_{j\in C} \alpha_j. \tag{1}
$$

Tästä saadaan käypä ratkaisu alkuperäiseen ongelmaan avaamalla kukin laitos riittävä määrä kertoja:

$$
y_i = \left\lceil \frac{\sum_{j\in C} x_{ij}}{u_i} \right\rceil \quad \text{kaikilla } i \in F.
$$

Koska $Y_i = 1$, jos laitokseen F_i on kytketty kaupunkeja, pätee nyt

$$
y_i \leq Y_i + \frac{\sum_{j\in C} x_{ij}}{u_i} \quad \text{kaikilla } i \in F. \tag{2}
$$

Epäyhtälöistä 1 ja 2 seuraa nyt

$$
\sum_{i\in F, j\in C} c_{ij}x_{ij} + \frac{3}{4}\sum_{i\in F} f_i y_i \leq 3\sum_{j\in C} \alpha_j.
$$

Nyt siis (x, y) on käypä ratkaisu alkuperäiseen kokonaislukuohjelmaan, ja sillä pätee

$$
\sum_{i\in F, j\in C} c_{ij}x_{ij} + \sum_{i\in F} f_i y_i \leq 4\sum_{j\in C} \alpha_j,
$$

joten meillä on 4-approksimointialgoritmi tarkasteltavaan ongelmaan.

3. Esitetään vakioapproksimointialgoritmi ryvästysongelmaan.

   Oletetaan, että että optimaalisessa ryvästyksessä jotkin pisteet $u_1, \ldots, u_t$ on liitetty keskipisteeseen $c = (u_1 + \cdots + u_t)/t$, ja u_1 on näistä pisteistä lähimpänä keskusta c. Jos ryvästyskeskuksena käytettäisiinkin pistettä u_1, saataisiin näiden pisteiden osuudeksi ryvästyksen kustannuksesta

   $$\sum_{i=1}^{t} ||u_i - u_1||_2^2 = \sum_{i=1}^{t} ||u_i - c||_2^2 + t||u_1 - c||_2^2 \leq 2\sum_{i=1}^{t} ||u_i - c||_2^2.$$

   Tässä yhtälö seuraa siitä, että c on pisteiden u_i keskiarvo, ja epäyhtälö taas siitä, että u_1 on pisteistä lähinnä keskusta c.

   Valitsemalla pisteet $v_1, \ldots, v_n$ sekä kaupunkien että laitosten joukoksi saadaan siis k-mediaaniongelman tapaus, jonka optimiratkaisu on korkeintaan kaksi kertaa kalliimpi kuin ryvästysongelman tapauksen. Lisäksi jokainen k-mediaaniongelman ratkaisu on myös sellaisenaan ryvästysongelman ratkaisu.

   Ylimääräisenä haasteena on kuitenkin se, että pisteiden väliset etäisyydet eivät toteuta kolmioepäyhtälöä, vaan ainoastaan etäisyyksien neliöjuuret. Tämä vaikuttaa k-mediaaniongelman approksimointialgoritmissa aliohjelmana käytettävään tuotantolaitosten sijoitteluongelman approksimointialgoritmiin niin, että approksimointikerroin 3 muuttuukin kertoimeksi 9. Ero tulee lemmasta 3.76, jonka todistuksessa tarvitaan kolmioepäyhtälöä. Jos yhdistämiskustannuksen neliöjuuri enintään kolminkertaistuu epäsuorassa yhdistämisessä, kasvaa kustannus korkeintaan yhdeksänkertaiseksi.

   Kolmioepäyhtälöä tarvitaan myös pyöristämisen yhteydessä lemmassa 3.79. Luentojen sivun 369 toinen epäyhtälörivi muuttuu nyt muotoon

   $$\sqrt{c_{i_3,j}} \leq 2\sqrt{c_{i_1,j}} + \sqrt{c_{i_2,j}} \iff c_{i_3,j} \leq 4c_{i_1,j} + 4\sqrt{c_{i_1,j}c_{i_2,j}} + c_{i_2,j}$$

   ja sitä seuraava odotusarvo vastaavasti muotoon

   $$\begin{aligned}
   \mathsf{E}[c_{\phi(j),j}] &\leq bc_{i_2,j} + a^2 c_{i_1,j} + ab(4c_{i_1,j} + 4\sqrt{c_{i_1,j}c_{i_2,j}} + c_{i_2,j}) \\
   &= ac_{i_1,j}(a + 4b) + bc_{i_2,j}(1 + a) + 4ab\sqrt{c_{i_1,j}c_{i_2,j}} \\
   &\leq ac_{i_1,j}(1 + 3b) + bc_{i_2,j}(1 + a) + 4ab \cdot \frac{c_{i_1,j} + c_{i_2,j}}{2} \\
   &= ac_{i_1,j}(1 + 5b) + bc_{i_2,j}(1 + 3a) \\
   &\leq (ac_{i_1,j} + bc_{i_2,j})(1 + 5 \cdot \max\{a, b\}).
   \end{aligned}$$

   Pyöristämisvaihe kasvattaa siis kustannuksen korkeintaan kuusinkertaiseksi (k-mediaaniongelmassa kaksinkertaiseksi). Löydetty ratkaisu on siis korkeintaan $2 \cdot 9 \cdot 6 = 108$ kertaa huonompi kuin ryvästysongelman optimiratkaisu.

4. Osoitetaan, että $\mathsf{PCP}(\log n, poly(n)) \subseteq \mathsf{NP}$.

   Tarkastellaan mielivaltaista luokan NP kieltä L ja muodostetaan sille ratkaisualgoritmi A, joka simuloi tarkastajaa. Algoritmi A aloittaa arvaamalla epädeterministisesti syötettä x vastaavan todistuksen y. Tämän jälkeen se käy läpi kaikki mahdolliset tarkastajan satunnaisjonot ja selvittää jokaisella satunnaisjonolla r, hyväksyykö tarkastaja syötteen x todistuksella y ja satunnaisjonolla r.

   Koska erilaisia satunnaisjonoja on polynominen määrä, toimii algoritmi A polynomisessa ajassa. Jos syöte x kuuluu kieleen L, tarkastaja hyväksyy syötteen kaikilla satunnaisjonoilla. Muussa tapauksessa tarkastaja hylkää syötteen ainakin puolella satunnaisjonoista. Polynomisessa ajassa toimiva epädeterministinen algoritmi A tunnistaa siis kielen L, joten kieli kuuluu luokkaan NP.

   Edellä todistettiin $\mathsf{PCP}(\log n, poly(n)) \subseteq \mathsf{NP}$. PCP-teoreeman perusteella tiedetään, että $\mathsf{PCP}(\log n, 1) = \mathsf{NP}$. Koska jokainen luokan $\mathsf{PCP}(\log n, 1)$ mukainen tarkastaja on myös luokan $\mathsf{PCP}(\log n, poly(n))$ mukainen tarkastaja, pätee $\mathsf{PCP}(\log n, 1) \subseteq \mathsf{PCP}(\log n, poly(n))$. Näin ollen pätee $\mathsf{PCP}(\log n, 1) = \mathsf{PCP}(\log n, poly(n))$.