Approksimointialgoritmit (kevät 2010)
Harjoitus 3 (8. helmikuuta)
Ratkaisuja (Jouni Siren)

1. Muodostetaan approksimointisuhteen säilyttävä palautus joukkopeiteongelmasta suunnattuun Steiner-puuongelmaan.

   Olkoon joukkopeiteongelman perusjoukko U, osajoukkojen kokoelma $\mathcal{S}$ ja painofunktio cost. Muodostetaan suunnattu verkko $G = (V, E)$, jonka solmujoukko koostuu pakollisesta juuresta r, Steiner-solmuista v_S kaikilla $S \in \mathcal{S}$ ja pakollisista solmuista v_x kaikilla $x \in U$. Kaikilla $S \in \mathcal{S}$ verkossa on suunnattu kaari (r, v_S) painoltaan $\mathsf{cost}(r, v_S) = \mathsf{cost}(S)$. Lisäksi kaikilla $S \in \mathcal{S}$ ja kaikilla $x \in S$ verkossa on suunnattu kaari (v_S, v_x) painoltaan $\mathsf{cost}(v_S, v_x) = 0$.

   Palautus voidaan selvästikin tehdä polynomisessa ajassa. Jokaista verkon G Steiner-puuta T vastaa joukon U joukkopeite C, joka koostuu niistä osajoukoista $S \in \mathcal{S}$, joilla solmu v_S kuuluu puuhun T, ja päinvastoin. Lisäksi $\mathsf{cost}(C) = \mathsf{cost}(T)$. Muodostettu palautus säilyttää siis approksimointisuhteen.

2. Tarkastellaan metrisen kauppamatkustajan ongelman muunnelmaa, jossa kauppamatkustajan reitti on Hamiltonin kehän sijasta Hamiltonin polku.

   (a) Christofideen 3/2-approksimointialgoritmi yleistyy tarkasteltavaan muunnelmaan, jos polun päätepisteistä on kiinnitetty korkeintaan toinen. Seuraavaksi käsitellään tapaus, jossa polun tulee alkaa solmusta A. Jos alkupistettä ei ole kiinnitetty, suoritetaan algoritmi kaikilla alkupisteillä $v \in V$ ja valitaan ratkaisuksi paras löytyneistä ratkaisuista.

   Etsitään verkon G pienin virittävä puu T. Olkoon V' verkon niiden solmujen joukko, joiden asteluku puussa T on väärä: parillinen solmulla A ja pariton muilla solmuilla. Koska paritonasteisten solmujen määrä on parillinen, on joukon V' koko aina pariton. Olkoon M joukon V' kokonaispainoltaan pienin maksimipariutus verkossa G. Tällainen pariutus löytyy polynomisessa ajassa. Koska joukossa V' on pariton määrä solmuja, jää yksi solmuista pariutuksen ulkopuolelle.

   Tarkastellaan moniverkkoa $T \cup M$. Jos solmun A asteluku puussa T on pariton, pätee $A \notin V'$, ja siten myös jonkin toisen solmun asteluvuksi moniverkossa $T \cup M$ tulee pariton. Jos taas solmun A asteluku on parillinen, pätee $A \in V'$, ja pariutus jättää jollekin solmulle $v \in V'$ väärän asteluvun moniverkossa $T \cup M$. Jos $v = A$, kaikkien solmujen asteluvut ovat parillisia. Jos taas $v \neq A$, solmujen v ja A asteluvut ovat parittomia ja muiden parillisia.

   Tarkastellaan ongelman optimaalista ratkaisua P_{OPT} ja muodostetaan siitä polku P ohittamalla solmut, jotka eivät kuulu joukkoon V'. Kolmioepäyhtälön nojalla $\mathsf{cost}(P) \leq \mathsf{cost}(P_{\mathsf{OPT}}) = \mathsf{OPT}$. Polusta P voidaan muodostaa kaksi toisistaan erillistä joukon V' maksimipariutusta M' ja M'' valitsemalla mukaan joka toinen kaari. Näin ollen pätee

   $$\mathsf{cost}(M) \leq \min(\mathsf{cost}(M'), \mathsf{cost}(M'')) \leq \frac{\mathsf{cost}(P)}{2} \leq \frac{\mathsf{OPT}}{2}.$$

   Koska moniverkossa $T \cup M$ on aina 0 tai 2 paritonasteista solmua, ja A on toinen mahdollisista paritonasteisista solmuista, voidaan aina löytää solmusta A alkava Eulerin polku P_E. Koska T on verkon G kokonaispainoltaan pienin yhtenäinen osaverkko, pätee sillä $\mathsf{cost}(T) \leq \mathsf{OPT}$. Näin ollen $\mathsf{cost}(P_E) = \mathsf{cost}(T) + \mathsf{cost}(M) \leq 3/2 \cdot \mathsf{OPT}$. Tämä polku voidaan muuntaa Hamiltonin poluksi P_H verkossa G ohittamalla ne solmut, joissa on käyty aikaisemmin polulla. Kolmioepäyhtälön nojalla pätee $\mathsf{cost}(P_H) \leq \mathsf{cost}(P_E) \leq 3/2 \cdot \mathsf{OPT}$, joten P_H on etsitty Hamiltonin polku.

   (b) Nyt verkossa G on kiinnitetty Hamiltonin polun molemmat päätepisteet A ja B. Muodostetaan ensin edellisen kohdan algoritmilla solmusta A alkava Hamiltonin polku, jonka kustannus on korkeintaan $3/2 \cdot \mathsf{OPT}$. Jos polun päätepiste on B, voidaan lopettaa.

   Muussa tapauksessa olkoon polun päätepiste C. Muokataan polkua niin, että ohitetaan solmu B suoraa kaarta pitkin, jolloin polun kustannus pysyy samana tai pienenee kolmioepäyhtälön takia. Lisätään polun loppuun kaari (C, B), jolloin siitä tulee taas käypä ratkaisu. Koska myös optimiratkaisussa on polku $C \rightsquigarrow B$, täytyy kolmioepäyhtälön takia olla $\mathsf{cost}(C, B) \leq \mathsf{OPT}$. Näin ollen muodostetun Hamiltonin polun $A \rightsquigarrow B$ kokonaiskustannus on korkeintaan $5/2 \cdot \mathsf{OPT}$.

3. Olkoon A jokin $\alpha(n)$-approksimointialgoritmi yleiseen k-keskusongelmaan, missä $\alpha \geq 1$ on polynomisessa ajassa laskettava funktio. Johdetaan siitä polynomisessa ajassa toimiva algoritmi dominoivalle joukolle.

   Olkoon $(G = (V, E), k)$ dominoivan joukon tapaus. Muodostetaan täydellinen painotettu verkko $G' = (V, E')$ siten, että $\mathsf{cost}(u, v) = 1$, jos $(u, v) \in E$, ja $\mathsf{cost}(u, v) = \alpha(n) + 1$, jos $(u, v) \notin E$. Tässä verkossa jokaisen k-keskusongelman käyvän ratkaisun kustannus on joko 1 tai $\alpha(n) + 1$ riippuen siitä, ovatko kaikki solmut (muut kuin keskukset) jonkin ratkaisuun kuuluvan keskuksen naapureita verkossa G. Toisin sanoen käyvän ratkaisun kustannus on 1 jos ja vain jos valitut keskukset muodostavat dominoivan joukon verkossa G. Optimiratkaisun kustannus on siis 1 jos ja vain jos verkossa G on korkeintaan k solmun dominoiva joukko.

   Jos optimiratkaisun kustannus on 1, algoritmi A löytää ratkaisun, jonka kustannus on korkeintaan $\alpha(n) \cdot 1$. Koska jokaisen käyvän ratkaisun kustannus on joko 1 tai $\alpha(n) + 1$, täytyy algoritmin löytää tässä tapauksessa optimiratkaisu. Vastaavasti jos optimiratkaisun kustannus on $\alpha(n)+1$, on mikä tahansa käypä ratkaisu optimaalinen, ja algoritmi A löytää sellaisen. Soveltamalla algoritmia A verkkoon G' voidaan siis ratkaista dominoiva joukko -ongelma verkossa G polynomisessa ajassa. Koska dominoiva joukko on NP-täydellinen, seuraa algoritmin A olemassaolosta $\mathsf{P} = \mathsf{NP}$.

   Tehtävän ratkaisemiseksi riittää siis osoittaa, että $\mathsf{P} \neq \mathsf{NP}$.

4. Tarkastellaan metristä k-ryväsongelmaa, jossa ryvästyksen kustannus on suurin etäisyys kahden samaan joukkoon sijoitetun solmun välillä. Oletetaan tarkasteltavan verkon olevan täydellinen, vaikka sitä ei tehtävänannossa mainitakaan.

   (a) Algoritmi valitsee ensin k keskussolmua $v_1, \ldots, v_k$, ja sijoittaa solmun v_a joukkoon V_i. Ensimmäiseksi keskussolmuksi v_1 voidaan valita mielivaltainen solmu. Kun keskussolmut $v_1, \ldots, v_i$ on valittu, valitaan solmuksi v_{i+1} se solmuista, jonka etäisyys lähimpään jo valittuun keskussolmuun on kaikkein suurin. Kun keskussolmut on valittu, sijoitetaan kukin myöhempi solmu v siihen joukkoon V_i, jonka keskussolmua v on lähimpänä.

   Olkoon v_{k+1} se solmu, joka olisi valittu seuraavaksi keskussolmuksi, jos sellainen olisi tarvittu, ja v_a sitä lähin keskussolmu. Solmuja $v_1, \ldots, v_{k+1}$ on siis yhteensä $k+1$ kappaletta, joten ainakin kahden niistä täytyy kuulua samaan joukkoon optimiratkaisussa. Koska $\mathsf{cost}(v_i, v_j) \geq \mathsf{cost}(v_{k+1}, v_a)$ kaikilla $1 \leq i, j \leq k + 1$, täytyy olla $\mathsf{cost}(v_{k+1}, v_a) \leq \mathsf{OPT}$. Toisaalta koska v_{k+1} on se solmu, jonka etäisyys lähimpään keskussolmuun on suurin, täytyy olla $\mathsf{cost}(u, v_i) \leq \mathsf{cost}(v_{k+1}, v_a)$ kaikilla keskussolmuilla v_i ja solmuilla $u \in V_i$. Kolmioepäyhtälön nojalla $2 \cdot \mathsf{cost}(v_{k+1}, v_a) \leq 2 \cdot \mathsf{OPT}$ on siis yläraja kahden samaan joukkoon sijoitetun solmun väliselle etäisyydelle, joten yllä esitetty algoritmi on 2-approksimointialgoritmi.

   Tiukka esimerkki approksimointikertoimesta 2 on solmujen $u_0, \ldots, u_3$ muodostama verkko, jossa $\mathsf{cost}(u_i, u_{i+1}) = 1$ ja $\mathsf{cost}(u_i, u_{i+2}) = 2$ kaikilla i (yhteenlaskut modulo 4). Tapauksessa $k = 2$ algoritmi voi valita keskussolmuiksi solmut u_0 ja u_2 ja tuottaa ryvästyksen $\{\{u_0, u_1, u_3\}, \{u_2\}\}$. Kustannukseksi tulee $\mathsf{cost}(u_1, u_3) = 2$, vaikka ryvästyksen $\{\{u_0, u_1\}, \{u_2, u_3\}\}$ kustannus olisi vain 1.

   (b) Oletetaan, että algoritmi A on $2 - \varepsilon$ -approksimointialgoritmi metriseen k-ryväsongelmaan jollain $\varepsilon > 0$. Olkoon $(G = (V, E), k)$ jokin klikkeihinositusongelman tapaus, ja ratkaistaan se algoritmia A käyttäen.

   Muunnetaan verkko G täydelliseksi painotetuksi verkoksi G'. Asetetaan ensin jokaisen kaaren $e \in E$ painoksi 1. Lisätään tämän jälkeen verkkoon kaikki siitä puuttuvat kaaret ja asetetaan niistä kunkin painoksi 2. Kaarten painot toteuttavat nyt kolmioepäyhtälön. Verkon G' jokaisen k-ryvästyksen kustannus on joko 1 tai 2 riippuen siitä, ovatko kaikki lisätyt kaaret ryvästyksen osajoukkojen välisiä eli onko jokainen osajoukko klikki verkossa G.

   Jos verkossa G' on k-ryvästys, jonka kustannus on 1, on sama ositus myös verkon G ositus k klikkiin. Jos taas tällaista ryvästystä ei ole, ei verkkoa G voi myöskään osittaa k klikkiin. Koska algoritmi A on $2 - \varepsilon$ -approksimointialgoritmi, ei se voi tuottaa kustannukseltaan 2 olevaa ryvästystä, jos $\mathsf{OPT} = 1$, joten sen täytyy tuottaa aina optimaalinen ryvästys. Koska klikkeihinositusongelma on NP-täydellinen, ei algoritmia A ole olemassa, ellei $\mathsf{P} = \mathsf{NP}$.