Laskuharjoitus 6

22.02.2001

1. Hae oheisen graafin suurin klikki Carraghan–Pardalosin algoritmilla (kirjan algoritmi 4.19 rajoittavalla funktiolla 4.15).

2. Hae oheisen graafin suurin klikki (luennolla käsitellyllä) Östergårdin algoritmilla.

3. Esitä jokin mielekäs kustannusfunktio ja naapuristo seuraaville vaikeille kombinatorisille optimointiongelmille.

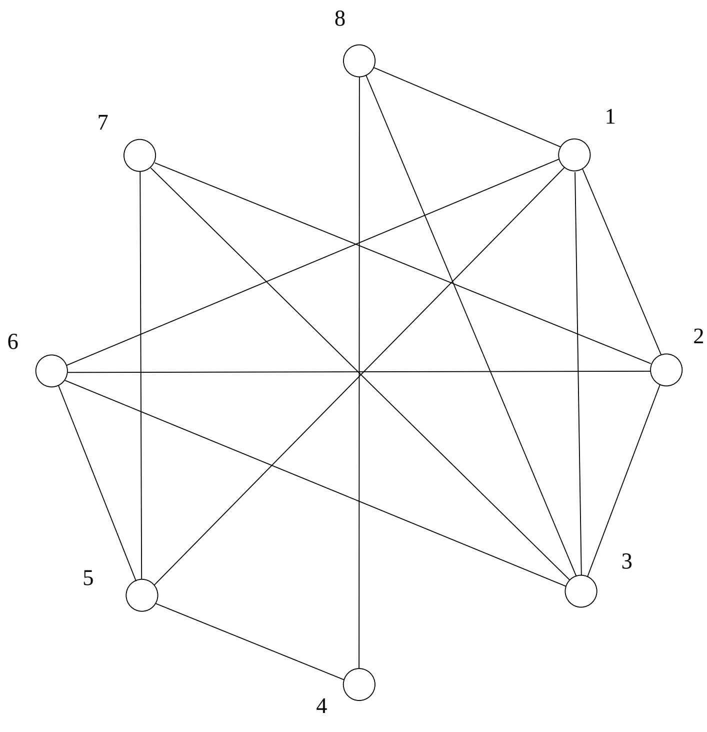
   (a) Symmetrinen kauppamatkustajan ongelma.

   (b) Graafin väritys n värillä siten, että naapurisolmuilla on eri väri.

   (c) Graafin ositus: Jaa graafin solmut, joita on parillinen määrä, kahteen yhtäsuureen osaan siten, että osien välisten kaarien määrä on mahdollisimman pieni.

   (d) Lukujen ositus: jaa joukko $A = \{a_1, a_2, \ldots, a_n\}$ kahteen osaan A_1 ja A_2 siten, että $A_1 \cup A_2 = A$, $A_1 \cap A_2 = \emptyset$ ja

$$\sum_{a \in A_1} a = \sum_{a \in A_2} a.$$



Kuva 5: Eräs graafi

Laskuharjoituksen 6 vastaukset

1. Algoritmissa käydään graafin noodit läpi depth-first-järjestyksessä. Noodit järjestetään aluksi johonkin järjestykseen. Kussakin hakupuun solmussa tarkastellaan vuorollaan kutakin graafin noodia, joka on kaikkien hakupolun varrella senhetkiseen klikkiin lisättyjen solmujen naapuri, ja jonka järjestysluku on suurempi kuin senhetkiseen klikkiin viimeksi lisätyn graafin noodin järjestysluku. Jos jossakin vaiheessa havaitaan, että parempaa klikkiä ei voida löytää, vaikka kaikki jäljelläolevat noodit voitaisiin lisätä klikkiin, karsitaan hakupuuta.

Tämänhetkinen klikki	Yhteiset naapurit	Huom!
-	1,2,3,4,5,6,7,8	
1	2,3,5,6,8	
1,2	3,6	
1,2,3	6	
1,2,3,6	-	{1,2,3,6}
1,2	6	karsitaan
1	3,5,6,8	
1,3	6,8	karsitaan
1	5,6,8	karsitaan
-	2,3,4,5,6,7,8	
2	3,6,7	karsitaan
-	3,4,5,6,7,8	
3	6,7,8	karsitaan
-	4,5,6,7,8	
4	5,8	karsitaan
-	5,6,7,8	karsitaan

Taulukko 4: Carraghan–Pardalosin algoritmin kulku esimerkissä

2. Östergårdin algoritmi toimii kuten Carraghan–Pardalosin algoritmi, mutta huipputasolla graafin noodeja tarjotaan klikin pienimmäksi käänteisessä järjestyksessä. Kun c(k) on suurimman sellaisen klikin koko, jonka järjestysnumeroltaan pienin alkio on k, on c(k) aina joko c(k+1) tai c(k+1)+1. Hakupuuta voidaan nyt karsia edellisen lisäksi silloin, kun tämänhetkisen klikin koko ja järjestysnumeroltaan pienimmän yhteisen naapurin c-arvo yhteensä eivät ylitä suurinta tähän asti löydettyä klikkiä.

3. Tässä tehtävässä ei ole parhaita mahdollisia ratkaisuja, vaan yleensä jotkin ratkaisut toimivat hyvin joissakin tapauksissa ja jotkin toiset ratkaisut joissain muissa tapauksissa.

   (a) Kustannusfunktiona luonnollisin valinta on reitin pituus. Eräs mahdollisuus naapuristoksi on ns. 2-OPT -siirrot: muodostetaan nykyistä reittiä kuvaavasta graafista uusi reitti poistamalla kaksi kaarta ja lisäämällä kaksi kaarta.

   (b) Tässä kuvataan ns. fixed-k lähestymistapa. Ratkaisu on mikä tahansa solmujen ositus k osaan (tyhjät osat sallitaan). Kaksi väritystä ovat tois-

Tämänhetkinen klikki	Yht. naapurit	Huom	
8	-	{8}	c(8)=1
7	-		c(7)=1
6	-		c(6)=1
5	6,7		
5,6	-	{5,6}	c(5)=2
4	5, 8		
4,5	-		
4	8	karsitaan	c(4)=2
3	6,7,8	karsitaan!	c(3)=2
2	3,6,7		
2,3	6,7		
2,3,6	-	{2,3,6}	c(2)=3
1	2,3,5,6,8		
1,2	3,6		
1,2,3	6		
1,2,3,6	-	{1,2,3,6}	c(1)=4

Taulukko 5: Östergårdin algoritmin kulku esimerkissä

tensa naapurit, jos toinen saadaan toisesta siirtämällä jokin solmu osasta toiseen. Kustannusfunktiona esim. niiden kaarien lukumäärä, joiden molemmat solmut ovat samassa osituksen osassa. Naapuristoa voidaan rajoittaa seuraavasti: nykyistä ratkaisua voidaan muuttaa vain niiden solmujen osalta, joista lähtee ainakin yksi väritysehtoa rikkova kaari; tällöin tosin naapuristo ei enää ole symmetrinen.

(c) Yksinkertainen lähestymistapa: ratkaisujoukkona on kaikki solmujen ositukset kahteen osaan. Siirto on kahden solmun vaihto osien välillä. Kustannusfunktiona kaarien määrä osien välillä.

Parempi suorituskyky on saatu simuloidulla jäähdytyksellä, kun on sallittu mielivaltaiset ositukset kahteen osaan V_1 ja V_2. Siirtoina on nyt yhden solmun siirto osasta toiseen, ja kustannusfunktiossa on lisäksi sakkona summatermi $\alpha(|V_1| - |V_2|)^2$, joka lisää kustannusta aina kun osat ovat erikokoiset. Optimointialgoritmi saattaa sakkotermistä huolimatta päätyä epätasapainoiseen ratkaisuun, jolloin voidaan joko kasvattaa α:n arvoa tai lisätä jokin yksinkertainen heuristiikka, joka muutamaa solmua siirtämällä pääsee tasapainotettuun ratkaisuun.

(d) Otetaan ratkaisujoukoksi kaikki ositukset kahteen osaan. Määritellään k-naapuristo sellaiseksi, jossa yhdellä siirrolla saadaan siirtää enintään k alkiota osituksesta toiseen. Kustannusfunktioksi voidaan ottaa

$$\min \left| \sum_{a \in V_1} a - \sum_{a \in V_2} a \right| .$$

Tämä ongelma on hiukan hankala sikäli, että siinä on pahimmillaan hyvin paljon lokaaleja optimeja. Simuloitu jäähdytys esitetyllä naapuristolla ei pärjää erityisesti tähän ongelmaan kehitetylle Karmarkar-Karp -heuristiikalle, mutta parempaa on kehitetty.