

Tiina Tutkija

Pro gradun kirjoittamisesta LaTeX-ohjelmalla

Tietotekniikan (tietoliikenne)
pro gradu -tutkielma
3.5.2000

Jyväskylän yliopisto
Tietotekniikan laitos

Sisältö

1	Johdanto	4
2	Symmetrisen matriisin symbolinen kolmiointi	4
2.1	Täyttyminen	4
2.2	Eliminointigraafit	5
2.3	Minimiastemenetelmä hyvän permutaation määrittämiseksi	6
2.4	Tekijägraafien käyttö symbolisessa kolmioinnissa	6
3	Langaton tiedonsiirto	7
3.1	Onko kuu juustoa?	8
3.2	Parametrisoidut splinit	8
4	Globaali optimointi	9
4.1	Geneettiset algoritmit	9
	Viitteet	11

Tiivistelmä

Tämän tekstin tarkoituksena on esitellä gradu.cls tyylitiedoston käyttöä prog gradu -tutkielman tekemisessä. Teksti sisältää sieltä täältä napattuja tekstinpätkiä, eikä siis mitään varsinaista asiaa!

Title in english: Writing MSc thesis using LaTeX.

1 Johdanto

T_EXon Donald E. Knuthin kehittämä ladontaohjelma, joka soveltuu erityisesti matemaattisen tekstin tuottamiseen. L^AT_EX-makrokirjasto sisältää monia sellaisia piirteitä, joita paljas T_EXei sisällä. Näitä ovat esimerkiksi sisällysluettelo, lukujen ja kaavojen automaattinen numerointi ja symbolinen viittaus niihin,...

gradu.cls -tyylitiedostossa asetetaan oikeankokoiset marginaalit ja määritellään ympäristöt maaritelma, lause, lemma, todistus, esimerkki ja algoritmi.

2 Symmetrisen matriisin symbolinen kolmiointi

2.1 Täyttyminen

Eräs mahdollinen valinta painofunktioksi on

$$w_i(x) = |x - x_i|^{-\mu}, \quad \mu > 0. \quad (2.1)$$

Edellinen painofunktio (2.1) ottaa huomioon kaikki interpolaatiopisteet x_i , joten työ määrä kasvaa suureksi.

Olkoon

$$\text{Nonz}(A) = \{(i, j) \mid a_{ij} \neq 0, i \neq j\}. \quad (2.2)$$

Olkoon A :lla kolmiohajotelma $A = LL^T$. Asetetaan $F = L + L^T$. Alkiot a_{ij} saadaan aina lausekkeista, missä esiintyy a_{ij} ja jotain muuta. Jos $a_{ij} \neq 0$, tulee vastaava tekijän alkio nolaksi vain sattumalta. Täten useimmiten

$$\text{Nonz}(A) \subseteq \text{Nonz}(F).$$

Määritellään

$$\text{Fill}(A) = \text{Nonz}(F) \setminus \text{Nonz}(A). \quad (2.3)$$

Tämä käsittää siis ne kolmiotekijöiden nolasta eroavat alkiot, joita vastaavissa kohdissa alkuperäisessä matriisissa A on nollat.

Jos tiedetään etukäteen mitkä L :n alkiot ovat $\neq 0$, voidaan käyttää matriisille A tiivistä listaesitystä. Tilaa varataan vain alkioille $l_{ij} : (i, j) \in \text{Nonz}(A) \cup \text{Fill}(A), i \geq j$ ja tarvittaville indekseille ja linkeille.

$$u(x) = \int_{\sqrt{2}}^{\sqrt{7}} \frac{x + \sin x}{1 + x^2 + \eta + \zeta} dx \quad (2.4)$$

Toimenpiteitä, joilla selvitetään $\text{Fill}(A)$, kaavan (2.4) syvin olemus ja muodostetaan tarvittavat tietorakenteet L :n tallettamiseksi, sanotaan matriisin *symboliseksi kolmioinniksi*.

Jotta symbolisesta kolmioinnista olisi hyötyä, tulee $\text{Fill}(A)$ pitää mahdollisimman pienenä. Osoittautuu, että muuttujien numeroinnilla voidaan siihen vaikuttaa suuresti.

2.2 Eliminointigraafit

Tarkastellaan, miten sarakemenetelmässä nollasta eroavien alkoiden määrä lisääntyy. Sarakkeesta k vähennetään sarakkeen j loppuosa kerrottuna $\overline{a_{kj}}$:lla. Sarakkeessa k on sen jälkeen nollasta eroava alkio a_{ik} , jos sellainen oli jo ennestään, tai $a_{kj} \neq 0$ ja $a_{ij} \neq 0$.

Olkoon $H^{(j)}$ symmetrinen osamatriisi, kun saraketta j käsitellään. Tarkastellaan tilannetta matriisin $H^{(j)}$ graafin avulla: Todetaan, että $H^{(j)}$:n ensimmäisessä sarakkeessa $v^{(j)}$ tietty komponentti $v_k^{(j)} \neq 0$ tarkalleen silloin, kun vastaavassa graafissa solmut k ja j ovat naapureita. Matriisissa $H^{(j+1)}$ taas solmut k ja i tulevat naapureiksi silloin, kun ne olivat naapureita jo graafissa $H^{(j)}$ tai ne molemmat olivat solmun j naapureita. Täten matriisia $H^{(j+1)}$ vastaava graafi saadaan matriisin $H^{(j)}$ graafista seuraavasti:

- poistetaan solmu j ja siitä sen naapureihin johtavat tiet
- kaikki solmun j naapurit graafissa $H^{(j)}$ yhdistetään pareittain teillä, mikäli niiden välillä ei ollut jo ennestään tietä.

Esimerkki 2.1 Tarkastellaan kuvan 2.1 matriisia $H^{(0)} := A$. Eliminoidaan muuttuja x_1 . Alkuperäistä matriisia vastaavassa graafissa G_0 solmun 1 naapurit ovat 2 ja 6. Poistetaan solmu 1 ja siitä johtavat tiet. Yhdistetään solmut 2 ja 6 tiellä, joka matriisissa vastaa sitä, että alkio a_{26} ja a_{62} tulevat nollasta eroaviksi.

Eukleideen algoritmi voidaan toteuttaa Fortran90-kielellä seuraavasti:

```
program GCD
  integer:: m, n, r
  do
    write(*,*) "Please give values for m and n"
    read(*,*) m, n
    do while ( n .ne. 0 )
      r = mod( m, n )
      m = n
      n = r
    end do
    write(*,*) "gcd = ", m
  end do
end program GCD
```

Lisää kielestä voi opiskella kirjoista [1], [2].

Sen jälkeen eliminoidaan muuttuja x_2 . Graafissa sen naapurit ovat 3,4 ja 6. Solmu 2 poistetaan ja solmut 3,4, ja 6 yhdistetään teillä jos niitä ei ennestään ole. Tässä tapauksessa ei ole, joten alakolmioon tulee kolme uutta nollasta eroavaa alkioita. Matriisit $H^{(j)}$ ja niitä vastaavat graafit on esitetty kuvassa 2.1.

: -)) - :

Kuva 2.1: Teatteri

2.3 Minimiastemenetelmä hyvän permutaation määrittämiseksi

Tavoitteena on saada sellainen muuttujen permutaatio, että lopullisessa tekijämatriisissa L olisi mahdollisimman vähän nollasta eroavia alkioita.

Olkoon G^F graafi, joka vastaa matriisin $F = L + L^T$ rakennetta. Tämä saadaan lisäämällä alkuperäiseen graafiin ne tiedot, jotka lisättiin eliminointigraafeihin eri vaiheissa. Näiden lukumäärä olisi pidettävä mahdollisimman pienenä valitsemalla solmujen järjestys sopivasti. Toistaiseksi ei tiedetä voidaanko tämä ongelma ratkaista polynomialisessa ajassa. Siksi käytetään heuristisia menetelmiä.

2.4 Tekijägraafien käyttö symbolisessa kolmioinnissa

Olkoot E^A ja E^F matriiseja A ja $F = L + L^T$ vastaavien graafien teiden joukot. Seuraava lause antaa L :n nollasta eroavien alkioiden paikat:

Lause 2.1 Suoritetaan eliminointi järjestyksessä x_1, x_2, \dots, x_n . Tie (i, j) kuuluu joukkoon E^F silloin ja vain silloin, kun se kuuluu joukkoon E^A tai on olemassa $k < \min\{i, j\}$ siten, että sekä tie (k, i) että tie (k, j) kuuluvat joukkoon E^F .

Olkoon S jokin solmujoukko suuntaamattomassa graafissa. Solmun i S -naapureiden joukko $\text{Reach}(i, S)$ määritellään seuraavasti: Solmu j on solmun i S -naapuri ($i, j \notin S$), jos j on i :n tavallinen naapurisolmu, tai jos on olemassa matriisi

$$C = \begin{bmatrix} \sqrt{2} & \frac{1}{3} & \frac{5}{6} \\ 0 & 12 & \pi \\ 45 & \frac{1}{9} & \sqrt{3} \end{bmatrix} \quad (2.5)$$

missä jokainen solmuista s_1, \dots, s_m kuuluu joukkoon S .

Lause 2.2 Suoritetaan eliminointi järjestyksessä x_1, x_2, \dots, x_n . Silloin solmun i naapurit graafissa G^F ovat $\text{Reach}(i, S_{i-1})$, missä S_{i-1} on solmujoukko $1, \dots, i-1$. Tässä Reach -operaatio suoritetaan graafissa G^A .

Olkoon P_1, P_2, \dots, P_m jokin graafin G solmujen partitiio. Muodostetaan uusi graafi, jonka solmut ovat P_1, \dots, P_m . Solmujen P_i ja P_j välillä on tie tarkalleen silloin, kun ainakin yhdestä joukon P_i solmusta johtaa tie johonkin joukon P_j solmuun alkupe-
räisessä graafissa G .

Huomautus 2.1 Koska \bar{S} koostuu aina erillisistä solmuista, voidaan joukko $\text{Reach}(i, \bar{S})$ määrätä yksinkertaisesti. Ensin kaikki ne i :n naapurit, jotka eivät kuulu joukkoon \bar{S} , otetaan mukaan. Olkoon sitten $P(i)$ niiden solmun i naapureiden joukko, jotka kuuluvat joukkoon \bar{S} . Jokaisella solmulla $p \in P(i)$ otetaan mukaan vielä ne p :n naapurit, mitkä eivät jo ennestään ole mukana. Huomattakoon, että mikään \bar{S} :n solmu ei voi olla $P(i)$:n solmujen naapurina. Täten hakua ei tarvitse laajentaa solmujen $P(i)$ ulkopuolelle.

Algoritmi 2.1 Symbolinen kolmiointi tekijägraafia käyttäen

1. Olkoon T vuorossa oleva tekijägraafi. Asetetaan aluksi $T := G^A$, missä G^A on matriisia A vastaava graafi. $\bar{S} := \emptyset$. Määrätään solmujen asteluvut.
2. Olkoon i se T :n solmu $\notin \bar{S}$, jonka tämänhetkinen asteluku on pienin. Solmun asteluku $:= |\text{Reach}(i, \bar{S})|$.
3. Muuttuja x_i on seuraava eliminointijärjestyksessä. Vastaavan matriisin L sarakkeen nolasta eroavien rivien numerot saadaan joukosta $\text{Reach}(i, \bar{S})$.
4. Solmu i liitetään joukkoon \bar{S} . Jos mikään i :n naapureista ei kuulunut joukkoon \bar{S} , tulee i :stä \bar{S} :n yksinäinen solmu. Jos i :llä on yksi tai useampia joukossa \bar{S} olevia naapureita, sulautetaan ne yhdeksi \bar{S} :n solmuksi.
5. Kohdassa 3 muodostetun joukon $\text{Reach}(i, \bar{S})$ solmujen naapurilistat muutetaan uutta tekijägraafia vastaaviksi, mikäli edellä sulautettiin solmuja yhteen. Joka tapauksessa määrätään uudelleen em. joukon solmujen asteluvut ja muutetaan astelukuihin perustuvaa solmujen ($\notin \bar{S}$) prioriteettijonoa vastaavasti. Toistetaan kohdasta 2 mikäli vielä on jäljellä solmuja, jotka eivät kuulu joukkoon \bar{S} . Muuten lopetetaan. Silloin solmujen i käyntijärjestys antaa minimias-temielessä määrätyn permutaation.

Realistisen näkymän aikaansaamiseksi käytetään perspektiiviprojektiota. Tämä voidaan tehdä seuraavasti:

```
...
glViewport( 0, 0, w, h );
glMatrixMode( GL_PROJECTION );
glLoadIdentity();
gluPerspective( 45.0, w/h, 1.0, 3*z );
...
```

Tässä $z = \max(w, h) \dots$

3 Langaton tiedonsiirto

Langaton tiedonsiirto on tullut suuren yleisön tietoisuuteen GSM-järjestelmän myötä. Kaksi pääteknologiaa ovat TDMA ja CDMA.

s	M	E
-----	-----	-----

Kuva 3.2: Liukuluvun talletusmuoto tietokoneessa.

3.1 Onko kuu juustoa?

Nollasta poikkeava luku x esitetään IEEE-aritmetiikan mukaisessa liukulukuesityksessä muodossa

$$x = (-1)^s M \cdot 2^E. \quad (3.6)$$

Kuva 3.2 havainnollistaa liukuluvun binääristä talletusmuotoa tietokoneessa.

IEEE-aritmetiikka tuntee myös aritmeettisten erikoistapausten symbolit Inf (ääretön) ja NaN (not-a-number).

Määritelmä 3.1 Jos ahne algoritmi antaa joukkosysteemissä I aina optimiratkaisun e.m. optimointiprobleemalle, niin I on matroidi.

3.2 Parametrisoidut splinit

Lemma 3.1 Jos solmulla ei ole olemassa mitään sellaista lyhintä reittiä, missä olisi enemmän kuin k tietä, määrää algoritmi lyhimmän reitin k ensimmäisen kierroksen kuluessa.

Todistus: Todetaan, että lemma on tosi kun $k = 1$. Alkusolmun naapureihin saadaan lyhimmat reitit selville jo ensimmäisellä kierroksella, mikäli niissä ei voi olla enempää kuin yksi tie. Induktio-oletus: Lemma on tosi arvoon $k = h$ saakka...

... jolloin viimeistään $h + 1$. kierroksella kiinnitetään lyhin reitti solmuun I . \square

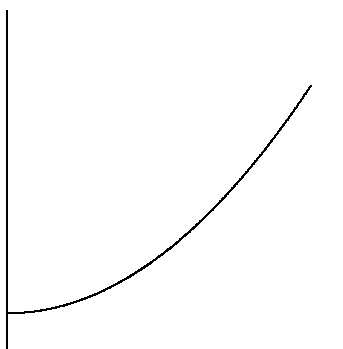
Tarkastellaan ohjelmaa, jonka skalaariversion kuluttama CPU-aika on T ja siitä osaa S kuluu skalaarikodissa ja V koodissa, joka voidaan vektoroida. Jos vektorointi nopeuttaa suoritusta n -kertaisesti, on vektoroidun ohjelman suoritukseen kuluva aika $S + V/n$, joten saavutettava nopeuden lisäys on

$$s = \frac{S + V}{S + V/n}.$$

Tämä tunnetaan *Amdahlin lakina*. Kuva 3.3 esittää suorituskvyn lisäystä vektorikoodin osuuden funktiona.

Algoritmi 3.1 (Kapsäkkiprobleeman LP-relaksaatio)

1. Asetetaan $N_0 := \emptyset$, $N_1 := \emptyset$, $N_F :=$ kaikki muuttujat, $\bar{a}_0 := a_0$.



Kuva 3.3: Suorituskyvyn lisäys vektorikoodin osuuden funktiona.

2. Valitaan jokin suhdeluku $q := c_i/a_i$. Suoritetaan joukon N_F ositus seuraaviin osajoukkoihin:

$$NS := \{i \in N_F \mid q_i > q\}$$

$$NY := \{i \in N_F \mid q_i = q\}.$$

Edelleen

$$S_1(q) := \sum \{a_i \mid i \in NS\}$$

$$S_2(q) := S_1(q) + \sum \{a_i \mid i \in NY\}$$

3. Jos $S_1(q) < \bar{a}_0 \leq S_2(q)$, lopetetaan. LP-relaksaation optimiratkaisu saadaan asettamalla....
4. Jos joukossa N_F on enemmän kuin θ muuttujaa, toistetaan askeleesta 2. Muuten ratkaistaan muuttujien N_F määräämä kapsäkkiprobleema järjestämällä nämä muuttujat suhteen c_i/a_i mukaiseen vähenevään järjestykseen...

4 Globaali optimointi

Epälineaaristen optimointitehtävien ratkaisualgoritmit löytävät lokaalin minimin, eikä globaalisuutta yleensä voida osoittaa, elleivät kohdefunktio ja käypä alue ole konvekseja. Yleensä...

4.1 Geneettiset algoritmit

Geneettiset algoritmit ovat luonnon evoluutiomekanismeja matkivia optimointimenetelmiä. GA-menetelmät eivät vaadi kohdefunktion jatkuvuutta, joten niitä voidaan käyttää perinteisillä menetelmillä vaikeasti ratkaistaviin ongelmiin.

: -)

Kuva 4.4: Hymiö on suosittu symboli sähköposteissa

Huomautus 4.1 I/O:ta voidaan tehdä myös samanaikaisesti ohjelman muun suorituksen kanssa. Myös undefine-tyypistä tiedostoa voi käsitellä asynkronisesti.

Lause 4.1 Olkoon G^A eräs graafi ja S eräs sen solmujen joukko. Muodostetaan S :n yhtenäisistä komponenteista tekijägraafi \bar{G}^A edellä selostetulla tavalla. Olkoon \bar{S} tekijägraafin solmujen joukko, joka koostuu S :n yhtenäisiä komponentteja vastaavista solmuista P_1, P_2, \dots . Oletetaan, että niiden solmujen numerointi, jotka eivät kuulu joukkoihin S tai \bar{S} on kummassakin graafissa sama. Silloin $\text{Reach}(i, S) = \text{Reach}(i, \bar{S})$, missä edellinen otetaan graafissa G^A ja jälkimmäinen tekijägraafissa \bar{G}^A .

Uusien nollasta eroavien alkioden määrä kierroksella i on korkeintaan eliminotavan solmun i naapureiden lukumäärä eli solmun i asteluku eliminointigraafissa juuri ennen kuin se poistetaan. Tästä syystä on luonnollista käyttää poistettavan solmun valinnassa ahnetta menetelmää: eliminoidaan aina se solmu, jonka asteluku on pienin. Tämä on antanut käytännössä hyviä tuloksia.

Lause 4.2 Kuvassa 4.4 on suoritettu esimerkin 2.1 matriisin eliminointi kahdella tavalla. Ylemmällä rivillä eliminointi suoritetaan ahneella menetelmällä, jolloin saadaan kaksi uutta tietä. Alemmalla rivillä suoritetaan eliminointi mahdollisimman huonolla tavalla, jolloin saadaan kuusi uutta tietä!

Viitteet

- [1] M. Counihan, "Fortran 90", Pitman, London, 1991
- [2] J. Haataja, J. Rahola & J. Ruokolainen, "Fortran 90/95" (2. uudistettu painos), CSC-Tieteellinen laskenta Oy, Espoo, 1998
- [3] J. D. Foley, "Computer graphics: principles and practice", Addison–Wesley, 1996.
- [4] M. Rabin, Probabilistic algorithms for testing primality, *J. Number Theory*, **12**, 128–138, 1980.
- [5] T. Rossi ja J. Toivanen, A Nonstandard Cyclic Reduction Method, its Variants and Stability , *SIAM J. Matrix Anal. Appl.*, **20**, 628–645, 1999.
- [6] D. Weber, An implementation of the general number field sieve to compute discrete logarithms mod p , kirjassa "Advances in Cryptology - Proc. Eurocrypt '95", Springer-Verlag, 95–105, 1995.