

# Konveksin QP-tehtävän laskenta-ajat sisäpistemenetelmillä

---

Noora Hyttinen 69673A

15.3.2012

*Työn saa tallentaa ja julkistaa Aalto-yliopiston avoimilla verkkosivuilla. Muilta osin kaikki oikeudet pidätetään.*

## Sisällysluettelo

1	Johdanto .....	2
2	Käsitteiden ja menetelmien esittely .....	2
2.1	Konvekssi QP-tehtävä .....	2
2.2	Markowitzin portfolio.....	2
2.3	Sisäpistemenetelmät .....	3
2.3.1	Estefunktiomenetelmä.....	3
2.3.2	Primaali-duaali-menetelmä.....	5
3	Työn toteutus .....	7
3.1	Estefunktiomenetelmä .....	7
3.2	Primaali-duaali-menetelmä .....	7
4	Tulokset .....	8
4.1	Estefunktiomenetelmä .....	8
4.2	Primaali-duaali-menetelmä .....	8
4.3	Menetelmien laskenta-aikojen vertailua.....	9
5	Yhteenveto .....	11
6	Lähteet .....	11
LIITTEET .....		11
	Estefunktiomenetelmä.....	11
	Primaali-duaali-menetelmä.....	12

# 1 Johdanto

Perinteisessä lineaaristen optimointitehtävien ratkaisemiseen tarkoitettussa simplex-algoritmissa tehtävää ratkaistaan kulkemalla käyvän alueen reunapisteissä kohti optimaalista ratkaisua. Suurissa tehtävissä saattaa kuitenkin olla tuhansia kulmapisteitä, jolloin tehokkaallakin tietokoneella tehtävän ratkaiseminen saattaa olla liian vaikeaa. Tämän takia epälinearisille tehtäville on jouduttu kehittämään ratkaisumenetelmiä, joiden avulla optimiin päästään suoremmin, ottamalla vähemmän askelia ja näin vältetään turhaa laskemista. Tavoitteena on saada ratkaistua päivittäin käytettäviä monimutkaisia optimointitehtäviä tarpeeksi nopeasti.

Työn tarkoituksena on tutkia kahden eri sisäpistemenetelmän, estefunktio- ja primaali-duaali-menetelmän laskenta-aikoja erään neliöllisen optimointitehtävän ratkaisussa. Ratkaistavana tehtävänä käytetään yksinkertaista konveksia Markowitzin portfoliotehtävää.

## 2 Käsitteiden ja menetelmien esittely

### 2.1 Konvekksi QP-tehtävä

Optimointitehtävät voidaan jakaa ryhmiin riippuen siitä, mihin luokkaan kohdefunktiot ja rajoitusehdot kuuluvat. Yleinen neliöllinen optimointitehtävä (QP-tehtävä) voidaan esittää muodossa

$$\min \quad \frac{1}{2}x^T Hx + Q^T x + r \quad (1)$$

$$\text{s.e.} \quad f(x) \leq h \quad (2)$$

$$Ax = b \quad (3)$$

missä kohdefunktio (1) on konvekksi ja neliöllinen, epäyhtälörajoitukset (2)  $f: \mathbf{R}^n \rightarrow \mathbf{R}^m$ .  $A \in \mathbf{R}^{p \times n}$ , jossa  $\text{rank } A = p < n$ . QP-tehtävässä rajoitusehdot ovat affiineja. Oletetaan että tehtävälle löytyy optimi pisteestä  $x^*$ .

Neliöllinen kohdefunktio (1) on konvekksi, kun  $H$  on positiivisesti semidefiniitti ja  $Q$  on reaalivektori.  $H$  on positiivisesti semidefiniitti, kun  $x^T Hx \geq 0$  kaikilla  $x$ :n nollasta poikkeavilla reaaliarvoilla. Kaikki  $H$ :n ominaisarvot ovat ei-negatiivisia.

### 2.2 Markowitzin portfolio

Yleinen käyttösovellus neliöllisen optimointialgoritmin käyttöön on erilaiset portfoliotehtävät, joissa käyttötarkoituksesta riippuen on kohdefunktiona tuoton varianssin minimointi tai tuoton maksimointi. Varianssin minimoinnissa käytetään rajoitusehtona tuoton odotusarvolle alarajaa, jota suurempi tuoton halutaan olevan. Tällainen tehtävä on yleensä muotoa:

$$\min \quad x^T H x \quad (4)$$

$$\text{s.e.} \quad -Cx \leq -D$$

$$\mathbf{1}^T x = 1$$

missä  $H$  on kovarianssimatriisi,  $C = \begin{bmatrix} p^T \\ I \end{bmatrix}$  ja  $D = \begin{bmatrix} r_{min} \\ 0 \end{bmatrix}$ ,  $p$  on tuoton odotusarvo ja  $r_{min}$  on vakio. Kovarianssimatriisit ovat määritelmän mukaisesti positiivisesti semidefiniittejä matriiseja [2], joten tehtävä on konvekksi.

Tehtävän tarkoituksena on minimoida portfolion varianssia, minkä avulla pyritään pitämään riski pienenä. Rajoitusehtona portfolion tuoton täytyy olla vähintään  $r_{min}$ . Epäyhtälörajoituksen viimeisen rivin ehto estää lyhyeksi myynnin ja yhtälörajoitus ostamisen yli omien varojen.

## 2.3 Sisäpistemenetelmät

Sisäpistemenetelmillä tarkoitetaan optimointimenetelmiä, joissa lähdetään ratkaisemaan tehtävää lähestymällä optimia käyvän alueen sisäpuolelta, toisin kuin ulkopistemenetelmissä.

Sisäpistemenetelmiä käytetään suurissa lineaarisissa optimointitehtävissä ja lisäksi epälineaarisissa tehtävissä. Sisäpistemenetelmät sopivat muun muassa konvekseille QP-, GP- ja QCQP-tehtäville[1].

### 2.3.1 Estefunktiomenetelmä

Estefunktiomenetelmässä epäyhtälörajoitukset siirretään kohdefunktioon estefunktion avulla ja ratkaistaan saatu yhtälörajoitteinen optimointitehtävä jollain optimointialgoritmillä.

Tehtävän ratkaiseminen aloitetaan jostain käyvästä sisäpisteestä. Estefunktion avulla luodaan algoritmille este, jonka ylityksestä minimoitavan kohdefunktion arvo kasvaa äärettömäksi. Näin estetään algoritmin pääseminen käyvän alueen ulkopuolelle. Kun rajoitusehdot esitetään muodossa  $f_i(x) \leq 0$ , ideaalinen estefunktio on sellainen, joka toteuttaa seuraavan porraskfunktion

$$I(f_i(x)) = \begin{cases} 0, & \text{kun } f_i(x) \leq 0 \\ \infty, & \text{kun } f_i(x) > 0 \end{cases} \quad (5)$$

Tällöin rajoitusehdot eivät vaikuta kohdefunktion suuruuteen käyvän alueen sisällä. Porraskfunktion implementointi tietokoneelle on sen verran hankalaa, että estefunktiolle käytetään approksimaatiota. Yleensä estefunktiona käytetään logaritmistä estefunktiota, joka on QP-tehtävällä muotoa

$$I = -\frac{1}{t} \log(-f(x) + h). \quad (6)$$

Estefunktion kanssa tehtäväksi tulee

$$\min \quad \frac{1}{2}x^T Hx + Q^T x + r + -\frac{1}{t} \log(-f(x) + h) \quad (7)$$

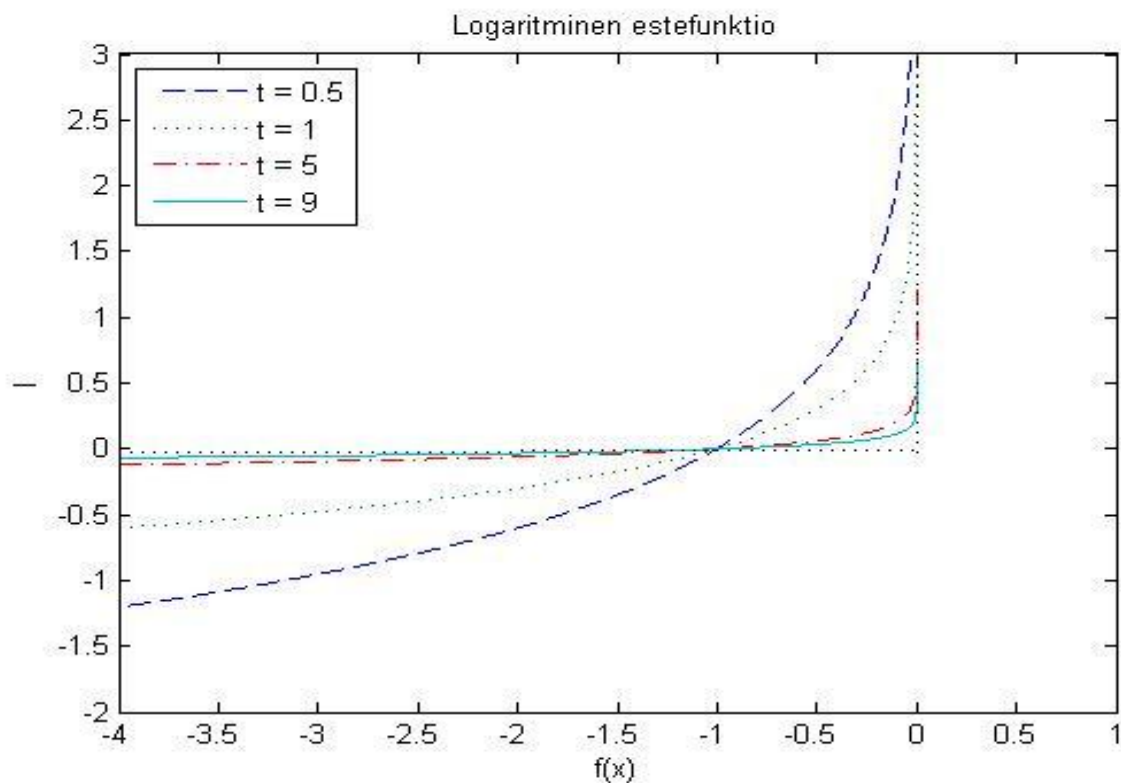
$$\text{s.e.} \quad Ax = b$$

Työssä käytettävälle optimointitehtävälle (4) saadaan estefunktiomenetelmää vastaavaksi tehtäväksi (7) yhtälörajoitettu minimointitehtävä

$$\min \quad x^T Hx - \frac{1}{t} \log(Cx - D). \quad (8)$$

$$\text{s.e.} \quad \mathbf{1}^T x = 1$$

Kun epäyhtälörajoitukset on lisätty kohdefunktioon, jäljellä on enää konvekssi optimointitehtävä, jolla on lineaariset yhtälörajoitukset. Tällainen tehtävä voidaan ratkaista helposti esimerkiksi MATLABilla sqp-algoritmillä. Sqp ratkaisee tehtävän Newtonin menetelmällä, jos tehtävässä ei ole rajoitteita tai KKT-ehtojen avulla, jos rajoitusehtoja on.



Kuva 1 Logaritmisen estefunktion vakion määrittäminen.

Yllä (Kuva 1) nähdään muutama estefunktio eri  $t$ :n arvoilla. Suuremmalla  $t$ :n arvolla saadaan tarkempi approksimaatio oikeasta kohdefunktion arvosta, mutta  $t$ :n suuri arvo myös hankaloittaa optimin laskemista, kun lähdetään pisteestä, joka on kaukana optimista. Optimaalinen tulos saadaan, kun  $t$ :tä päivitetään laskun aikana. Suoritetaan  $t$ :n päivittäminen seuraavasti:

1. Optimoidaan tehtävää pienellä  $t > 0$ , enintään 100 iteraatiokierrosta.
2. Asetetaan uudeksi aloituspisteeksi edellinen optimi.
3. Lopetusehtona  $m/t < \epsilon$ ,  $m$  on epäyhtälörajoitusten määrä.
4. Kasvatetaan  $t$ :n arvoa  $t := \mu t$ ,  $\mu > 1$

Koska käyvän alueen ulkopuolella kohdefunktion arvo on ääretön, on aina optimoinnilla saatu väliaikainen optimi myös käyvän alueen sisällä, joten sitä voidaan käyttää algoritmin alkupisteinä.

### 2.3.2 Primaali-duaali-menetelmä

Primaali-duaali-menetelmässä lähestymistapa optimointiin on hieman erilainen kuin estefunktiomenetelmässä. Primaali-duaali-menetelmässä primaali- ja duaali-asteet eivät välttämättä ole aina käypiä ratkaisuja.

Karush-Kuhn-Tucker –ehdot (KKT-ehdot) ovat epälineaarissa optimoinnissa välttämättömät optimaalisuusehdot. Oletetaan, että optimointitehtävällä (7) on lokaali minimi pisteessä  $x^*$ . Silloin ovat olemassa KKT kertoimet  $\lambda_i, i = 1, \dots, m$  ja  $\nu_j, j = 1, \dots, l$ .

QP-tehtävän Lagrangen funktio on muotoa

$$L(x, \lambda, \nu) := \frac{1}{2} x^T H x + Q^T x + r + \sum_{i=1}^m \lambda_i (f(x) - h) + \sum_{j=1}^l \nu_j (A x - b) \quad (9)$$

Näiden avulla tehtävälle saadaan KKT-ehdot:

$$\nabla L(x^*, \lambda, \nu) = 0 \quad (10)$$

$$\lambda_i (f(x^*)) = 0, \text{ kun } 1 \leq i \leq m \quad (11)$$

$$\lambda_i \geq 0, \text{ kun } 1 \leq i \leq m \quad (12)$$

$$A x^* - b = 0, \text{ kun } 1 \leq i \leq l \quad (13)$$

Tehtävän optimi saadaan ratkaisemalla yhtälöryhmä(10)-(13).

Primaali-duaali-menetelmässä tarkastellaan muunnettuja KKT ehtoja. Primaali-duaali-menetelmässä ratkaisua etsitään kulkemalla optimia kohti käyvän alueen sisältä päin. Tästä johtuen KKT-ehdot eivät toteudu sekä primaalille että duaalille samanaikaisesti. Muunnettuihin KKT-ehtoihin on lisätty rako primaali- ja duaali-alueiden väliin, mistä johtuen duaali ei aivan toteuta KKT-ehtoja optimissa. Suurella  $t$ :n arvolla duaali melkein täyttää muunnetut KKT-ehdot.

Muunnetut KKT ehdot voidaan esittää vektorina:

$$r_t(x, \lambda, v) = \begin{bmatrix} \nabla \left( \frac{1}{2} x^T H x + Q^T x + r \right) + Df(x)^T \lambda + A^T v \\ -diag(\lambda) f(x) - \left( \frac{1}{t} \right) \mathbf{I} \\ Ax - b \end{bmatrix} = 0 \quad (14)$$

missä  $t > 0$ . Tässä derivaatta  $Df(x)$  on määritelty

$$Df(x) = \begin{bmatrix} \nabla f_1(x)^T \\ \vdots \\ \nabla f_m(x)^T \end{bmatrix} \quad (15)$$

$r_t$  voidaan approksimoida  $y$ :n lähellä ensimmäisen asteen Taylorin approksimaatiolla  $r_t(y + \Delta y) \approx r_t(y) + Dr_t(y)\Delta y = 0$ , mistä saadaan ratkaistua etsintäsuunta  $\Delta y_{PD} = -Dr_t(y)^{-1}r_t(y)$ . Hakusuunta saadaan laskettua yhtälöryhmällä

$$\begin{bmatrix} \nabla^2 \left( \frac{1}{2} x^T H x + Q^T x + r \right) + \sum_{i=1}^m \lambda_i \nabla^2 (f_i(x) - h) & D(f(x) - h)^T & A^T \\ -diag(\lambda) f(x) & -diag(f(x) - h) & 0 \\ A & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \\ \Delta v \end{bmatrix} = -r_t(x, \lambda, v) \quad (16)$$

Portfoliotehdävälle viivahaun hakusuunta saadaan sijoittamalla käytetty portfoliotehtävä (7) yleisen QP-tehtävän tilalle yhtälöön (16)

$$\begin{bmatrix} H & -C^T & \mathbf{1} \\ diag(\lambda)C & diag(Cx - D) & 0 \\ \mathbf{1}^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \\ \Delta v \end{bmatrix} = - \begin{bmatrix} Hx - C\lambda + \mathbf{1}^T v \\ diag(\lambda)(Cx - D) - (1/t)\mathbf{I} \\ \mathbf{1}^T x - 1 \end{bmatrix}. \quad (17)$$

Kun hakusuunta on tiedossa, tarvitaan enää askelpituus  $s$ . Ensinnäkin määritetään suurin positiivinen askelpituus, jolla  $\lambda + s\Delta\lambda_{PD} \geq 0$ ,  $s^{max} = \min\{1, \min\{-\lambda_i/\Delta\lambda_i \mid \Delta\lambda_i < 0\}\}$ . Lähdetään pienentämään askelta ensin  $s = 0,995s^{max}$ , jonka jälkeen kerrotaan askelpituutta kertoimella  $\beta \in (0,1)$ , kunnes  $f(x + s\Delta x_{PD}) < h$  eli ollaan vielä käyvän alueen sisäpuolella askeleen ottamisen jälkeen. Kertomista jatketaan, kunnes  $\|r_t(x + s\Delta x_{PD}, \lambda + s\Delta\lambda_{PD}, v + s\Delta v_{PD})\|_2 \leq (1 - \alpha s)\|r_t(x, \lambda, v)\|_2$ , missä  $\alpha$  on yleensä valittu väliltä  $(0,01; 0,1)$ .

Kun muuttuja  $x$  toteuttaa epälineaariset rajoitusehdot,  $\lambda > 0$ ,  $\mu > 1$ ,  $\epsilon_{käypä} > 0$ ,  $\epsilon > 0$ , voidaan tehtävän optimi ratkaista primaali-duaali-menetelmän algoritmilla, jossa silmukkaa

1. Asetetaan  $t := \mu m / \hat{\eta}$ , missä  $\hat{\eta}$  on duaaliaukko.
2. Muodostetaan primaali-duaali etsintäsuunta  $\Delta y_{PD}$
3. Viivahaku ja päivitys. Määritetään askelpituus  $s > 0$  ja asetetaan uudeksi  $y := y + s\Delta y_{PD}$ .

toistetaan kunnes minimimuutoskriteeri tulee vastaan.

Etukäteen tiedetään, että primaali-duaali-menetelmä on usein tehokkaampi kuin estefunktiomenetelmä varsinkin tehtävissä, joissa vaaditaan suurta tarkkuutta. [1]

### 3 Työn toteutus

Laskenta-aikojen mittaamista varten tarvitaan ensin konvekssi neliöllinen tehtävä. Luodaan yksinkertainen portfoliotehtävä muotoa (7), missä  $H$  on  $n \times n$  diagonaalimatriisi, jonka alkiot ovat tasajakautuneita satunnaislukuja väliltä  $[0,1]$ . Matriisi  $p$  on  $n \times 1$  vektori, joka muodostuu tasajakauman satunnaisluvuista välillä  $[0,1]$ . Tehtävän kokoa voidaan muunnella ottamalla matriiseista  $H$  ja  $p$  osia, koska määritelmän mukaan positiivisesti semidefiniitin matriisin osamatriisit ovat myös positiivisesti semidefiniittejä. Tuoton minimiarvoksi valitaan  $r_{min} = p * x_0 - \varepsilon_r$ , jolloin aloituspiste on myös käypä.  $\varepsilon_r$ :lle käytettiin arvona  $10^{-2}$ .

Sekä estefunktiomenetelmä että primaali-duaali-menetelmä implementoitiin MATLABilla siten, että tehtävän ratkaisemisen lisäksi mitataan samalla laskemiseen kuluvaa aikaa. MATLAB koodit ovat liitteenä (LIITE 1).

#### 3.1 Estefunktiomenetelmä

Estefunktiomenetelmässä lähdettiin alkupisteestä  $x_i = 1/n$ , joka on käyvän alueen sisällä oleva piste.  $t$ :n arvoa päivitettiin siten, että lähdettiin arvosta  $t_0 = 10$ , jota kasvatettiin kertoimella  $\mu = 10$ . Optimointi lopetettiin, kun  $m/t < \epsilon$ , missä  $\epsilon = 10^{-6}$ .

Joka kierroksella optimia ratkaistiin MATLABin sqp-algoritmilla, jossa lopetuskriteerinä käytettiin minimiaskelpituutta. Laskenta-aikoja mitattiin kymmenellä eri minimiaskeleella välillä  $[10^{-3}, 10^{-12}]$  ja lisäksi minimiaskeleella  $10^{-20}$ . Samaa neliöllistä tehtävää ratkaistiin erisuuruksilla lopetuskriteereillä ja tehtävän ollessa kokoa  $n \times n, n = 2, \dots, 100$ . Sqp:ssä lopetuskriteerinä oli myös iteraatioiden maksimimäärä 100, mikä tuli vastaan suuremmilla tehtävillä ja pienillä minimiaskelpituuksilla.

#### 3.2 Primaali-duaali-menetelmä

Primaali-duaali-menetelmässä käytettiin myös alkupisteenä  $x_i = 1/n$  ja lopetuskriteerinä  $x$ :n minimiaskelpituutta. Minimiaskeleina käytettiin samoja arvoja kuin estefunktiomenetelmällä. Alkuarvoina duaalimuuttujille käytetään  $\lambda_i = 1$  ja  $\nu = 2$ .

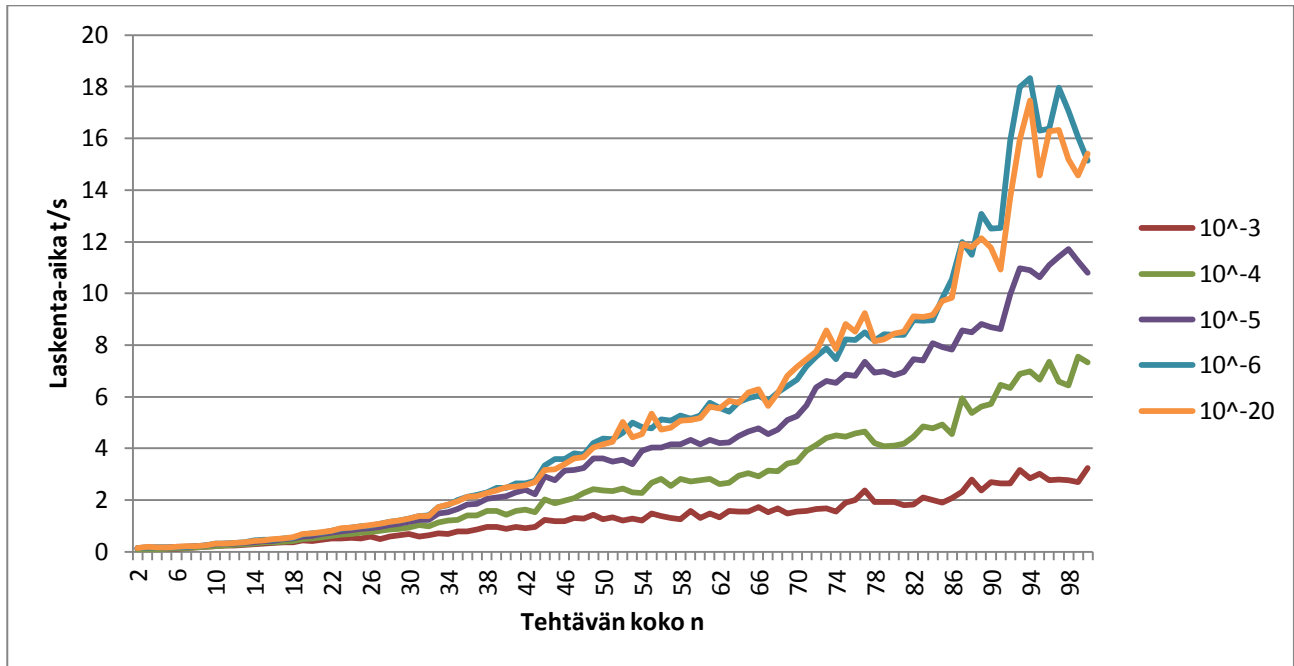
Askelpituuden määrittämiseen käytettiin parametreja  $\alpha = 0,01$  ja  $\beta = 0,7$ . Samoin kuin estefunktiomenetelmässä,  $\mu$ :n arvona käytettiin 10.



## 4 Tulokset

### 4.1 Estefunktiomenetelmä

Estefunktiomenetelmällä laskuaikoja tarkasteltiin muuttamalla yhden iteraatioaskeleen minimiaskelpituutta sqp-algoritmissa. Estefunktiomenetelmällä saatiin hyvin erisuuruisia laskenta-aikoja muuttamalla tehtävän tarkkuutta. Kun minimiaskelpituutta pienennetään tarpeeksi,  $< 10^{-6}$ , saadaan samansuuruisia laskenta-aikoja.

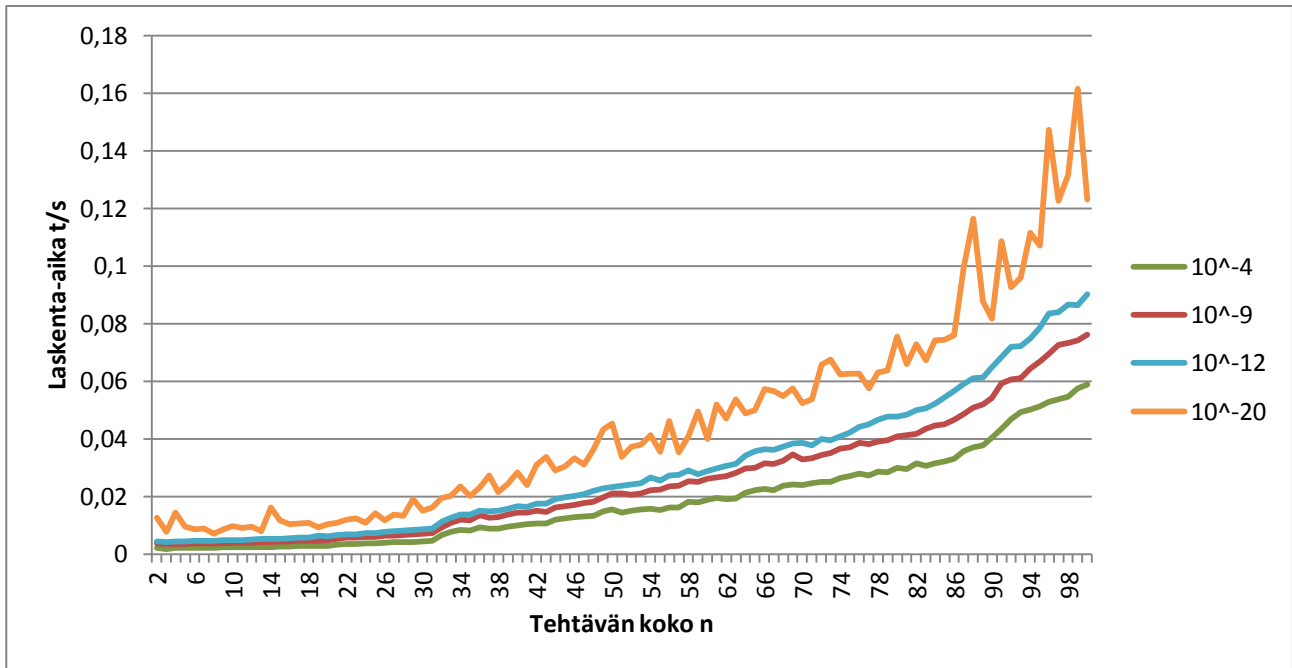


Kuva 2 Estefunktiomenetelmän laskenta-aikoja viidellä eri minimiaskelpituudella tehtäväkoon funktiona.

Kuvasta (Kuva 2) nähdään, että suuremmalla minimiaskeleella saadaan laskenta-aikoja pienennettyä selvästi. Tällöin ei tietysti päästä aivan minimiin asti, mutta halutun tarkkuuden puitteissa voidaan päättää, miten paljon aikaa tehtävän ratkaisemiseen voidaan käyttää. Pienimmillä minimiaskelpituuksilla sqp-algoritmissa päästään optimiin asti. Tästä johtuen laskenta-aikaan ei enää vaikuta, pienennetäänkö minimiaskelpituutta vai ei.

### 4.2 Primaali-duaali-menetelmä

Primaali-duaali-menetelmässä laskenta-aikoja tarkasteltiin myös muuttamalla minimiaskelpituutta.



Kuva 3 Primaali-duaali-menetelmän laskenta-aikoja neljällä minimiaskelpituudella tehtäväkoon funktiona.

Kuvassa (Kuva 3) on laskenta-aikoja neljällä eri minimiaskelpituudella. Minimiaskeleiden pienentäminen kasvattaa laskenta-aikoja tasaisesti ja ajat kasvavat koon kasvaessa polynomisesti. Työssä käytetyllä tehtävällä kaikki laskenta-ajat ovat alle sekunnin, mutta tehtävää kasvattamalla saadaan myös aikaisiksi pidempiä aikoja. Suuremmilla tehtävillä tarkkuutta ei voida pienentää, sillä minimiaskeleiden ollessa alle  $10^{-4}$ , algoritmi ei saa otettua yhtään askelta, koska minimiaskeleiden saavutetaan ennen ensimmäistä askelta.

#### 4.3 Menetelmien laskenta-aikojen vertailua

Tuloksista nähdään selvästi, että primaali-duaali-menetelmällä saadaan neliölliset tehtävät ratkaistua nopeammin kuin estefunktiomenetelmällä. Toisaalta kun lopetuskriteerijä löysennetään, päästään estefunktiomenetelmällä myös kohtuullisiin laskenta-aikoihin.

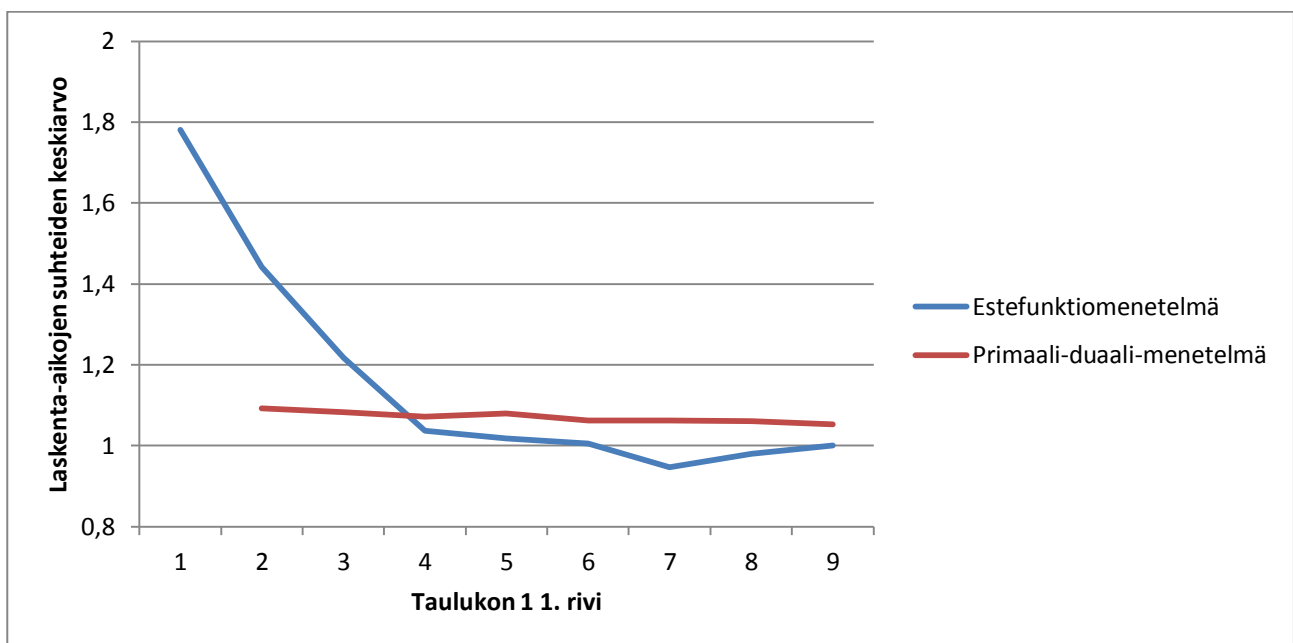
Primaali-duaali-menetelmässä tehtävää ei saada ratkaistua suurella lopetuskriteerillä, jos lähdetään aloituspisteestä, joka on melko lähellä ratkaisua. Heti ensimmäinen askel on niin pieni, että laskeminen katkeaa ja algoritmi antaa ratkaisuksi aloituspisteen. Estefunktiomenetelmässä päästään hieman suuremmallakin lopetuskriteerillä lähelle optimia. Esimerkiksi minimiaskeleella  $10^{-3}$  algoritmi ratkaisee tehtävän estefunktiomenetelmällä, primaali-duaali-menetelmällä ei.  $10^{-2}$  on minimiaskeleidenä liian suuri molemmissa menetelmissä. Toisaalta primaali-duaali-menetelmässä kohdefunktiolle saadaan viiden merkitsevän numeron tarkkuudella samat arvot lopetuskriteeristä riippumatta, kun minimiaskeleiden  $\leq 10^{-4}$ .

Laskenta-aikojen muutoksen tarkastelua varten lasketaan aikojen osamäärät, kun lopetuskriteeriä pienennetään kymmenesosaan edellisestä. Suhteiden keskiarvoista saadaan taulukon (Taulukko 1) mukaiset muutokset. Esimerkiksi primaali-duaali-menetelmässä, kun muuttujien minimimuutosta pienennetään  $10^{-4}$ :stä  $10^{-5}$ :een, kasvaa laskenta-aika keskimäärin 9,3 %.

**Taulukko 1** Laskenta-aikojen muutosten keskiarvot estefunktio- ja primaali-duaali-menetelmillä.

	$\frac{t_{10^{-4}}}{t_{10^{-3}}}$	$\frac{t_{10^{-5}}}{t_{10^{-4}}}$	$\frac{t_{10^{-6}}}{t_{10^{-5}}}$	$\frac{t_{10^{-7}}}{t_{10^{-6}}}$	$\frac{t_{10^{-8}}}{t_{10^{-7}}}$	$\frac{t_{10^{-9}}}{t_{10^{-8}}}$	$\frac{t_{10^{-10}}}{t_{10^{-9}}}$	$\frac{t_{10^{-11}}}{t_{10^{-10}}}$	$\frac{t_{10^{-12}}}{t_{10^{-11}}}$
Estefunktio- menetelmä	1,781	1,442	1,217	1,038	1,017	1,005	0,946	0,980	1,001
Primaali-duaali - menetelmä	-	1,093	1,082	1,071	1,080	1,063	1,062	1,060	1,053

Taulukon (Taulukko 1) arvot voidaan myös esittää graafisesti (Kuva 4), jolloin aikojen muutos nähdään selvemmin.



**Kuva 4** Laskenta-aikojen muutosnopeus estefunktioimenetelmällä ja primaali-duaali-menetelmällä.

Estefunktioimenetelmässä suuren minimiaskelpituuden pienentäminen kymmenesosaan johtaa siihen, että laskenta-aika kasvaa reilusti. Minimimuutoksen ollessa pieni ( $< 10^{-6}$ ), minimiaskelen pienentäminen kymmenesosaan ei enää kasvata laskenta-aikaa merkittävästi. Primaali-duaali-menetelmässä laskenta-ajat kasvavat pienempienkin minimimuutosten pienentämisestä.

Laskenta-aikoja eri menetelmillä ei voida tarkasti verrata, koska laskemisessa käytetyt koodit eivät ole aivan samanlaisia. Voidaan kuitenkin nähdä, miten estefunktioimenetelmällä laskenta-ajat ovat huomattavasti suurempia kuin primaali-duaali-menetelmällä.

## 5 Yhteenveto

Työssä käytetyllä tehtävällä estefunktiomenetelmällä saadut laskenta-ajat ovat keskimäärin yli 100-kertaisia primaali-duaali-menetelmään verrattuna.

Primaali-duaali-koodissa tehtävän matriisit vaikuttavat suuresti siihen, saadaanko tehtävää edes ratkaistua. Ennakkoon tiedettiin, että primaali-duaali-menetelmä on nopeampi tarkemmillä tehtävillä. Tämä saatiin myös todennettua, mutta toisaalta tämän työn primaali-duaali-menetelmän toteutuksessa oli jotain ongelmia, joiden takia algoritmi jumittui joskus paikoilleen. Estefunktiomenetelmässä samanlaista ongelmaa ei ollut. Esimerkiksi tehtävällä, jossa matriisi  $H$  ei ole diagonaalimatriisi, primaali-duaali-menetelmällä ei saada optimia ratkaistua suurilla tehtävillä, kun taas estefunktiomenetelmä ratkaisee tehtävän samassa ajassa kuin diagonaalimatriisitehtävän.

Eri sisäpistemenetelmät kannattaa valita sen mukaan, haluaako ratkaista monimutkaista tehtävää vai useita yksinkertaisia tehtäviä. Tulevaisuudessa sisäpistemenetelmiä olisi syytä tutkia myös muiden tehtävätyyppien ja eri lopetuskriteerien kannalta.

## 6 Lähteet

1. Boyd, Stephen & Vandenberghe, Lieven: *Convex Optimization*. Cambridge: University Press, 2004  
[http://www.stanford.edu/~boyd/cvxbook/bv\\_cvxbook.pdf](http://www.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf), 12.2.2010
2. Markowitz, Harry M. *Portfolio Selection: Efficient Diversification of Investments*, Wiley, 1959

## LIITTEET

### Estefunktiomenetelmä

```
eps = 1e-6;
mu = 10;
% Epäyhtälörajoituksia ei ole, koska ne liitetään kohdefunktioon
A2 = [];
B2 = [];
t_este = zeros(100,1);
options = optimset('Algorithm','sqp','MaxIter',100,'TolX',1e-6);

for j = 2:100
    m = 1+j;
    matriisit
    % Yhtälörajoitus
    Aeq = Aeqi(1:j)';
    Beq = Beqi;
    x0 = ones(j,1)/j; % Ensimmäinen aloituspiste
    t = 10; % Alkuarvo logaritmisestä estefunktiosta t:lle
    % Silmukka, joka ratkaisee tehtävän kasvattamalla t:tä kertoimella
    % mu ja ottamalla aloituspisteeksi edellisen t:n tuottaman ratkaisun.
    tic
    [x,fval] = fmincon(@estefunktio,x0,A2,B2,Aeq,Beq,lb,ub,[],options,t,j);
```

```

while m/t>=eps           % Lopetetaan, kun t on tarpeeksi suuri
    x0 = x;               % Uusi aloituspiste
    t = mu*t;            % t:n päivitys
    [x,fval] = fmincon(@estefunktio,x0,A2,B2,Aeq,Beq,lb,ub,[],options,t,j);
end
t_este(j) = toc;
j = j+1;
end

```

## matriisit.m

```

avektori           % 100x1 vektori, jossa tasajakautuneita satunnaislukuja
                  % välillä [0,1]
% Epäyhtälörajoitukset
Ai = [a(1:100) diag(ones(100,1))];
Bi = [(ones(j,1)/j)'*a(1:j)-1e-2;zeros(j,1)];
% Yhtälörajoitus, x:en summa on 1
Aeqi = ones(100,1);
Beqi = 1;
% x:n ala- ja yläraja. Alaraja sisältyy myös epäyhtälörajoitukseen
lb = 0;
ub = Inf;

```

## estefunktio.m

```

% Estefunktiomenetelmällä ratkaistava funktio
function f = estefunktio(x,t,j)
DH2;                % H kokonaisena (100x100)
matriisit;          % rajoitusmatriisit
H = diag(H1(1:j,1)); % H pienennettynä tehtävälle sopivaan kokoon
f = x'*H*x + sum(-(1/t)*log10(Ai(1:j,1:j+1)'*x-Bi)); % Kohdefunktio

```

## Primaali-duaali-menetelmä

```

t_prim = zeros(100,1);
for i = 2:100
    x0 = ones(i,1)/i;
    tolerance = 1e-6; %minimiaskel MUUNNETTAVA KRITTEERI
    mu = 10;
    tic
    [x,f] = pdmenetelma(x0,i,100,tolerance,mu);
    t_prim(i) = toc;
    i = i+1;
end

```

## pdmenetelma.m

```

function [x,f] = pdmenetelma (x,i,maxiter,tolerance,mu)
% Vakioiden arvoja
s_max = 0.995;
beta = 0.7;
alpha = 0.01;
m = 1+i;
nu = 2;

```

```

% Tehtävän matriisit
matriisit;
DH2;
% Epäyhtälörajoitukset
c = -Ai(1:m,1:i)*x + Bi(1:m);
lambda = ones(m,1);
H = diag(H1(1:i,1));
% Kohdefunktio
f = x'*H*x;

for iter = 1:maxiter

    dualitygap = -c'*lambda;
    t = mu*m/dualitygap;
    % rt alussa
    r1 = H*x - Ai(1:m,1:i)'*lambda + ones(i,1)*nu;
    r2 = -diag(lambda)*(-Ai(1:m,1:i)*x + Bi) - (1/t)*ones(m,1);
    r3 = ones(i,1)'*x - 1;
    rt = [r1; r2; r3];

    % D(rt)
    dr1 = [H -Ai(1:m,1:i)' ones(i,1)];
    dr2 = [-diag(lambda)*(-Ai(1:m,1:i)) -diag(-Ai(1:m,1:i)*x+Bi) zeros(m,1)];
    dr3 = [ones(1,i) zeros(1,m) 0];
    drt = [dr1; dr2; dr3];
    % Hakusuunta
    d = -inv(drt)*rt;
    dx = d(1:i);
    dlambda = d(i+1:i+m);
    dnu = d(i+m+1:end);
    % Askelpituuden määrittäminen
    s = s_max;
    is = find(dlambda < 0);
    if length(is)
        s = s_max * min(1,min(lambda(is) ./ -dlambda(is)));
    end
    % Kierrosten määrä
    while true
        % Uudet arvot muuttujille
        xnew = x + s * dx;
        lambdanew = lambda + s * dlambda;
        nunew = nu + s * dnu;
        % Minimiaskelkriteeri
        if norm(s*dx) < tolerance
            return
        end
        % Kohdefunktion arvo uudella pisteellä
        f = xnew'*H*xnew;
        % Rajoitukset uudella pisteellä
        c = -Ai(1:m,1:i)*xnew + Bi;
        % rt:n arvot uudessa pisteessä
        r1new = H*xnew - Ai(1:m,1:i)'*lambdanew + ones(i,1)*nunew;
        r2new = -diag(lambdanew)*(-Ai(1:m,1:i)*xnew + Bi) - (1/t)*ones(m,1);
        r3new = ones(i,1)'*xnew - 1;
        rtnew = [r1new; r2new; r3new];
        % Askelpituuden sopivuus
        if sum(c > 0) == 0 && norm(rtnew) <= (1-alpha*s)*norm(rt)
            x = xnew;
            lambda = lambdanew;
            nu = nunew;
            break
        end
    end
end

```

```
        end
        % Askeleen pienennys
        s = s*beta;
    end
end
end
```

#### matriisit.m

```
avektori          % 100x1 vektori, jossa tasajakautuneita satunnaislukuja
                  % välillä [0,1]
%Yhtälörajoitukset
Aeqi = ones(100,1);
Beqi = 1;
% Epäyhtälörajoitukset
Ai = [a(1:100)';diag(ones(100,1))];
Bi = [(ones(i,1)/i)'*a(1:i)-1e-2;zeros(i,1)];
```