TEKNILLINEN KORKEAKOULU
Informaatio- ja luonnontieteiden tiedekunta

Arttu Klemettilä

EPÄLINEAARISTEN HINNOITTELUTEHTÄVIEN NUMEERINEN ANALYSOINTI JA RATKAISEMINEN

Kandidaatintyö

Espoo 1.9.2008

Työn ohjaaja:

DI Kimmo Berg

Tekijä: Arttu Klemettilä

Työn nimi: Epälineaaristen hinnoittelutehtävien numeerinen analysointi ja ratkaiseminen

Päivämäärä: 1.9.2008 Kieli: Suomi Sivumäärä: 4+25

Tutkinto-ohjelma: Teknillinen fysiikka ja matematiikka

Valvoja: Prof. Harri Ehtamo

Ohjaaja: DI Kimmo Berg

Työssä tutkittiin epälineaarisen hinnoittelutehtävän ominaisuuksia numeeristen menetelmien avulla. Tehtävässä monopolinen yritys yrittää löytää sopivat hinnat sekä laadut myytäville tuotepaketeille. Asiakaskunta on jaettu eri asiakasluokkiin, joista kullakin on omat tunnetut hyötyfunktionsa. Tuotteet on suunniteltava niin, että jokainen ostaa oman tuotteensa, eikä koe toisen asiakasluokan tuotetta parempana vaihtoehtona.

Tässä työssä keskityttiin tehtävän numeerisen ratkaisun ominaisuuksiin. Tehtävän numeerisen vaativuuden, eli laskenta-ajan, määräävät pääosin asiakasluokkien ja laatu-parametrien lukumäärä. Laskenta-aika on verrannollinen lukumäärien neliöihin, ja suurien tehtävien laskenta voi tehokkaallakin laitteistolla olla kohtuullisessa ajassa mahdotonta. Erityisesti asiakasluokkien määrä nostaa vaadittua laskenta-aikaa, sillä rajoitusehtoja muodostuu luokkien määrän neliön verran.

Tehtävän rajoitusehtojen vaiheittaisella lisäämisellä voidaan kuitenkin saavuttaa merkittäviä laskennallisia nopeutuksia. Tällä menetelmällä voidaan vähentää sopivantyyppisten tehtävien laskenta-aikaa suurilla tehtävillä parhaimmillaan muutamaan prosenttiin alkuperäisestä. Lisäksi mikäli tehtävän ratkaisun tyyppi osataan päätellä etukäteen joko kokonaan tai osittain, voidaan tietoa käyttää nopeuttamaan ratkaisua entisestään.

Lisäksi selvitettiin, onko algoritmin antama optimi globaali optimi. Tämän huomattiin riippuvan voimakkaasti asiakasluokkien hyötyfunktioista. Todettiin, että neliöjuurimuotoiset hyötyfunktiot suppenevat selkeästi yhteen tiettyyn globaalin optimin, kun taas esimerkiksi normaalijakauman tiheysfunktion kaltaisilla hyötyfunktioilla optimin arvon riippuvuus alkuarvauksesta oli merkittävä, eikä saatu lokaali optimi ollut enää sama kuin globaali optimi.

Avainsanat: epälineaarinen hinnoittelu, numeeriset menetelmät, optimointialgoritmit, laskenta-aika, lokaali optimointi

Sisältö

Käytetyt lyhenteet ja symbolit

Lyhenteet

IC	Kannustinyhteensopivuus (Incetive compatibility) -rajoitusehto
IR	Yksilön rationalisuus (Individual rationality) -rajoitusehto
SQP	Toistettu kvadraattinen optimointi, eräs epälineaarisen tehtävän optimointialgoritmi

Symbolit

n	Asiakasluokkien lukumäärä
d	Laatu-ulottuvuuksien lukumäärä
$I = \{1, ..., n\}$	Asiakasluokkien joukko
$D = \{1, ..., d\}$	Laatu-ulottuvuuksien joukko
t_i	Asiakasluokkien painokertoimet
π	Tuotto
π^*	Monopolin optimaalinen tuotto
Φ	Tehokkuus
c	Kustannus
q_i	Asiakasluokan i laatuvektori
p_i	Asiakasluokan i tuotteen hinta
$u_i(q)$	Asiakasluokan i hyöty laadulla q

1 Johdanto

Epälineaarisessa hinnoittelussa on kyse tuoteperheen suunnittelusta ja hinnoittelusta. Monopolinen yritys valmistaa tuotteita, joita se voi räätälöidä eri asiakasryhmien tarpeisiin. Eri asiakasryhmillä eli luokilla on erilaiset tarpeet ja vaatimukset, joita voidaan kuvata kullekin luokalle ominaisten hyötyfunktioiden avulla. Täydellinen yksilöllistäminen ja hintadiskriminointi ei kuitenkaan ole mahdollista, sillä tuotteet myydään julkisesti, jolloin mikä tahansa tuote on jokaisen asiakkaan valittavissa. Tuotteet on sen vuoksi suunniteltava niin, etteivät asiakkaat halua ostaa millekään muulle kuin omalle asiakasluokalleen suunniteltua tuotetta, mutta kuitenkin siten, että yrityksen tuotteiden myynnistä saama voitto maksimoituu. Tuotteella voi olla useita säädettäviä laatuparametrejä, joita kutsutaan laatu-ulottuvuuksiksi.

Tehtävä voidaan formuloida matemaattisesti tavalliseksi epälineaariseksi optimointitehtäväksi. Kohdefunktioksi asetetaan tuotteista saatavat myyntivoitto. Rajoitteet taas syntyvät jokaisen asiakasluokkaparin välille siten, että luokan on saatava parempi hyöty omasta tuotepaketistaan kuin muista myytävistä tuotepaketeista. Tehtävä voidaan ratkaista tavanomaisilla epälineaarisen optimoinnin menetelmillä, mutta erityisesti tehtävän koon kasvaessa perinteiset algoritmit eivät enää ratkaise tehtävää kohtuullisessa ajassa. Tämän vuoksi on kehitettävä menetelmiä, jotka ratkaisevat suurenkin tehtävän nopeammin.

Ongelman sovelluksia löytyy kaikkialta. Yksinkertaisimmillaan kyseessä voi olla kuluttajatuotteita, kuten matkapuhelimia, valmistava yritys. Samaa menetelmää voidaan kuitenkin soveltaa esimerkiksi sähkön hinnoitteluun tai tilanteisiin, joissa ei varsinaisesti myydä tai valmisteta mitään tuotetta, kuten yrityksen rekrytoinnissa tai strategian suunnittelussa. Tällöin yhtälöiden tulkinta vain hieman muuttuu, mutta matemaattisesti ongelma pysyy samana. Muista sovelluksista, kuten valtion monopolin säätelystä ja huutokaupan organisoinnista voi lukea Rochetin ja Stolen tutkimuksista [1].

Hinnoitteluongelmaa on tutkittu 1970-luvulta lähtien. Tutkimus alkoi yhden asiakasryhmän ongelmasta, mutta pian siirryttiin tutkimaan useamman asiakasluokan mallia. Usean asiakasluokan ongelmaa tutki ensimmäisenä Spence [2]. Koska mallissa tarvittavia asiakkaiden hyötyfunktioita ei käytännön sovelluksissa tunneta tarkasti, on selvitetty myös tehtävän ratkaisemista epätäydellisen informaation vallitessa. Tällöin asiakkaiden hyötyfunktiot tunnetaan vain osittain, mutta niistä voidaan siitä huolimatta päätellä tehtävän ratkaisu, tai sen arvio.

Tämä työ liittyy systeemianalyysin laboratorion Harri Ehtamon ja Kimmo Bergin tutkimukseen. He ovat tutkineet muun muassa hinnoittelutehtävän ratkaisemista epätäydellisen informaation vallitessa yksiulotteisessa tapauksessa [3, 4]. Nyt tutkimus halutaan laajentaa useampaan laatu-ulottuvuuteen, mutta sitä ennen on selvitettävä, minkälaisia ominaisuuksia moniulotteisuus tuo tehtävään mukanaan. Tässä työssä on tarkoitus perehtyä moniulotteisen tehtävän ratkaisun ominaispiirteisiin. Tämän jälkeen voidaan epätäydellisen informaation tutkimusta laajentaa moniulotteiseen tehtävään.

Työssä on tarkoitus selvittää miten hinnoittelutehtävä kannattaa ratkaista numeerisesti ja mitkä ovat ratkaisun löytämisen suurimmat ongelmat. Tavoite on selvittää, miten olemassaolevia ratkaisumenetelmiä voidaan muokata siten, että ne ratkaisevat tämäntyyppisen tehtävän tehokkaammin. Lisäksi selvitetään, ovatko numeeristen menetelmien antamat ratkaisut globaaleja vai ainoastaan lokaaleja optimeja. Menetelmissä keskitytään Matlab-ohjelmistolla suoritettaviin ajoihin, joissa tehtävää ratkotaan erilaisilla parametreillä ja algoritmeilla. Ajojen avulla pyritään selvittämään erityisesti optimin globaaliutta ja ratkaisemiseen vaadittavia laskenta-aikoja.

Työn rakenne on seuraava: Seuraavassa luvussa esitellään hinnoittelutehtävä ja sen perusominaisuudet. Tämän jälkeen luvussa 3 perehdytään tehtävän ratkaisumenetelmiin yleisellä tasolla, minkä jälkeen seuraa kaksi lukua numeerisia esimerkkejä. Ensimmäisessä näistä, luvussa 4, selvitetään miten tehtävä kannattaa ratkaista mahdollisimman tehokkaasti, ja luvussa tarkastellaan 5 tehtävän optimin globaaliutta.

2 Hinnoittelutehtävän esittely

2.1 Matemaattinen formulointi

Tässä työssä käytetty formulointi on sama kuin Bergin ja Ehtamon käyttämä malli [5], joka on moniulotteinen versio Mussan ja Rosenin mallista [6] ja johon on lisätty Spencen käyttämä asiakasluokkien diskretointimalli [2]. Mallin tarkempaan analyysiin voi perehtyä Rochetin ja Stolen julkaisussa [7].

Monopolinen yritys myy tuotetta. Asiaskaskunta on jaoteltu n erilaiseen asiakasluokkaan, joihin kuuluvilla henkilöillä on keskenään samankaltaiset mieltymykset. Merkitään asiakasluokkien joukkoa $I = \{1, 2, ..., n\}$. Asiakasryhmiä varten yritys on suunnitellut tuotevalikoimansa siten, että jokaiselle luokalle pyritään räätälöimään oma tuote. Tuotteilla on d erilaista laatuparametria, jotka yritys voi valita vapaasti. Laatu-ulottuvuuksien joukko olkoon $D = \{1, 2, ..., d\}$. Merkitään asiakasryhmälle i suunnatun tuotteen laatuvektoria q_i, jonka pituus on d. Laadut voidaan rajoittaa tai skaalata sovelluskohtaisesti, mutta tässä työssä oletetaan ainoastaan, että $q_i \geq 0 \ \ \forall i \in I$.

Kullekin asiakasryhmälle on määritelty hyötyfunktio $u_i(q)$, joka kertoo, kuinka paljon kukin asiakasryhmä i arvostaa kutakin laatua q. Eri laatujen valmistaminen aiheuttaa erilaiset kustannukset $c(q)$. Lisäksi tuotteella on hinta p_i, jonka asiakas joutuu maksamaan ostaessaan tuotteen. Kaikki asiakasryhmät eivät ole välttämättä samankokoisia tai -arvoisia, joten määritellään jokaiselle luokalle painokerroin t_i.

Asiakkaan i saama hyöty tuotteesta saadaan vähentämällä asiakkaan hyötyfunktiosta tuotteen hinta, eli $u_i(q_i) - p_i$. Jotta asiakas ostaisi tuotteen, saatavan hyödyn on oltava positiivinen. Tästä saadaan ensimmäinen rajoitusehto, jota kutsutaan yksilön rationaliteetti- eli IR- (individual rationality) rajoitusehdoksi. Toisaalta kun asiakas valitsee tuotteen, hän voisi myös valita minkä tahansa toiselle asiakasryhmälle suunnitellun tuotteen. Jotta näin ei kävisi, tuote on suunniteltava siten, että jokainen asiakas saa omasta tuotteestaan suuremman hyödyn kuin mistä tahansa toisen asiakasryhmän tuotteesta, eli $u_i(q_i) - p_i \geq u_i(q_j) - p_j, \ \forall i, j \in I, j \neq i$. Tätä kutsutaan kannustinyhteensopivuus- eli IC- (incentive combatibility) rajoitteeksi. IR-rajoitteet voidaan muuttaa yhdeksi IC-rajoitteeksi yksinkertaisesti luomalla asiakasluokka 0, jonka hyöty kaikista tuotteista on aina nolla.

Yrityksen tuotekohtainen tuotto π_i koostuu myyntihinnasta, josta vähennetään kustannukset, jolloin saadaan $\pi_i = p_i - c(q_i)$. Kun yksittäisten tuotteiden tuotto kerrotaan vastaavalla asiakasryhmän painokertoimella ja painotetut tuotot lasketaan yhteen, saadaan kokonaistuotolle kaava $\pi = \sum_{i \in I} t_i \cdot \pi_i = \sum_{i \in I} t_i \cdot (p_i - c(q_i))$. Yrityksen tavoite on maksimoida tuotteista saamansa voitto ja toisaalta saada kaikki tuotteensa myytyä. On siis valittava jokaiselle asiakasluokalle tuotepaketit eli laadut q_i ja hinnat p_i niin, että yllä mainitut rajoitteet ovat voimassa. Näin saadaan kokonaisuudessan tehtävä muotoon

$$
\begin{aligned}
&\max_{q,p} \sum_{i \in I} t_i(p_i - c(q_i)) \\
&\text{s.e.} \quad u_i(q_i) - p_i \geq 0, \quad \forall i \in I && \text{(IR)} \\
&\qquad u_i(q_i) - p_i \geq u_i(q_j) - p_j, \quad \forall i, j \in I, i \neq j. && \text{(IC)}
\end{aligned}
$$

Esimerkki käytännön tilanteesta voisi olla matkapuhelinvalmistaja. Yritys valmistaa puhelimia kolmelle eri asiakasryhmälle, ja jokaisessa puhelimessa voidaan vaikuttaa akunkestoon ja näytön kokoon. Merkitään asiakasluokkien joukkoa $I = \{1, 2, 3\}$ ja laatuparametrien joukkoa $D = \{1, 2\}$. Asiakasryhmä 1 on suurin, mutta sille riittää halpa malli, jonka ei tarvitse olla hienointa mahdollista tekniikkaa. Toinen ryhmä vaatii taas hieman enemmän akun kestoa, mutta ei niinkään välitä näytön koosta. Kolmas arvostaa akunkestoa ja suuria näyttöjä selvästi muita enemmän. Ryhmä 1 on yhtä suuri kuin kaksi muuta ryhmää yhteensä. Akunkestoa merkitään paramerillä $q_{i,1}$ ja näytön kokoa $q_{i,2}$, jossa i on asiakasluokan numero. Asiakasryhmien koot ovat $t = (2, 1, 1)$ ja hyötyfunktiot esimerkiksi $u_i = \sum_{k \in D} c_{i,k}\sqrt{q_{i,k}}$. $c_{i,k}$ on hyötyfunktiossa esiintyvä painokerroin ja $q_{i,k}$ laatua k vastaava parametri. Parametreiksi $c_{i,k}$ voidaan esimerkiksi markkinatutkimuksen avulla saada $c_1 = (2, 2.2)$, $c_2 = (2.5, 1.8)$ ja $c_3 = (3, 3.5)$. Akkujen ja näyttöjen valmistamisen kustannukset riippuvat tuotteen laadusta. Akkujen kustannukset laadulle q_i ovat $0.001q_{i,1}^2$ ja näyttöjen $3.5 \cdot 10^{-5}q_{i,2}^3$. Esimerkkitehtävän ratkaisu on esitelty seuraavassa osiossa.

2.2 Tehtävän ominaisuuksia

Mikäli tehtävällä ei olisi lainkaan IC-rajoitusehtoja, on ratkaisu yksikertaisesti kohdefunktion, eli tuoton gradientin nollakohdassa, eli kun $\nabla u_i(q_i) = \nabla c(q_i)$. Tällöin jokaisen asiakkaan tuote on riippumaton toisesta. Hinta voidaan valita niin, että se on yhtä suuri kuin asiakasryhmän hyötyfunktion arvo kyseisellä laadulla. Asiakkaan ylijäämä, eli tuotteen hinta vähennettynä asiakkaan saamasta hyödystä, on tällöin nolla, ja asiakas suostuu juuri ja juuri ostamaan tuotteen. Kyseistä laatua kutsutaan tehokkaaksi laaduksi q^*. Suurin mahdollinen voitto π^* saadaan myymällä tehokasta laatua suurimmalla mahdollisella hinnalla.

Kun IC-rajoitteet huomioidaan, tilanne muuttuu. Tuotteiden hintoja ja laatuja joudutaan muuttamaan, sillä nyt jokin asiakasluokka voi kokea saavansa paremman ylijäämän ostamalla toisen asiakasryhmän tuotteen kuin oman tuotteensa. Usein tämä johtaa siihen, että kalliimman tuotteen hintaa täytyy laskea, jotta asiakkaan ylijäämä molemmilla tuotteilla olisi sama. Tällöin sanotaan, että asiakas on indifferentti eri tuotteiden välillä.

Tuottoisuus määritellään tuotteista saatavan voiton, $\pi = \sum_{i \in I} \pi_i = \sum_{i \in I}(p_i - c(q_i))$, ja maksimaalisen voiton π^* suhteena $\frac{\pi}{\pi^*}$. Tuottoisuus on suurimmillaan silloin, kun myydään tehokasta laatua ja hintaa. IC-rajoitusten vuoksi tästä joudutaan kuitenkin tinkimään, sillä muuten asiakasluokat kokisivat toisten luokkien tuotteet itselleen edullisempana.

Lisäksi määritellään tuotteille tehokkuus Φ_i, joka saadaan asiakkaan ja yrityksen saavuttaman kokonaishyödyn ja suurimman mahdollisen tuoton suhteena $\Phi_i = \dfrac{u_i + \pi_i}{\pi_i^*}$. Tehokasta ratkaisua, eli kun $\Phi = 100\%$, kutsutaan Pareto-optimaaliseksi. Tällöin asikkaan tai yrityksen hyötyä ei voi parantaa huonontamatta toista. Vapailla markkinoilla tehokkuus ajautuu kilpailun myötä automaattisesti tehokkaseen tilaan, mutta monopolisen yrityksen tapauksessa voi käydä niin, että tehokkuus laskee. Tehokkaassa ratkaisussa tuotteen hinnan laskeminen suurimmasta mahdollisesta hinnasta laskee yrityksen voittoa, mutta samalla lisää asiakkaan saamaa hyötyä, jolloin tehokkuus pysyy samana.

Joskus voi käydä niin, että optimissa kahden tai useamman asiakasluokan tuotteet ovat samoja. Tätä kutsutaan niputtamiseksi (eng. bunching). Tällöin näillä tuotteilla on samat laadut, ja IC-ehdoista johtuen myös samat hinnat. Niputettuja luokkia voidaan usein, esimerkiksi niin kutsutun single-crossing -oletuksen voimassaollessa, käsitellä yhtenä luokkana, joiden koko on kahden luokan yhteenlaskettu koko. Tästä voi kuitenkin aiheutua ongelmia tiettyjen, erityisesti analyyttisempien ratkaisutapojen yhteydessä.

Ratkaisussa on mahdollista myös tilanne, että jokin laatuparametri on nolla. Tämän voi yleensä tulkita, että tällöin tuotetta ei myydä lainkaan. Jossain tilanteissa tämä tulkinta ei kuitenkaan ole järkevä. Esimerkiksi jos nollaparametri kuvaa matkapuhelimen akunkestoa, eli asiakas luultavasti ostaisi koko tuotetta, vaikka muilta ominaisuuksiltaan se olisikin hyvä. Tällöin parametrin nolla-arvolla voidaan mallintaa esimerkiksi jotain minimitasoa, joka vähintään myydään kaikille. Kustannus on siten lisä-hinta laadun parantamisesta. Perustuotteen hinta oletetaan jokaiselle asiakasluokalle samaksi, jolloin se ei vaikuta optimoinnin tulokseen, vaan ainoastaan muuttaa kohdefunktiota vakioarvolla.

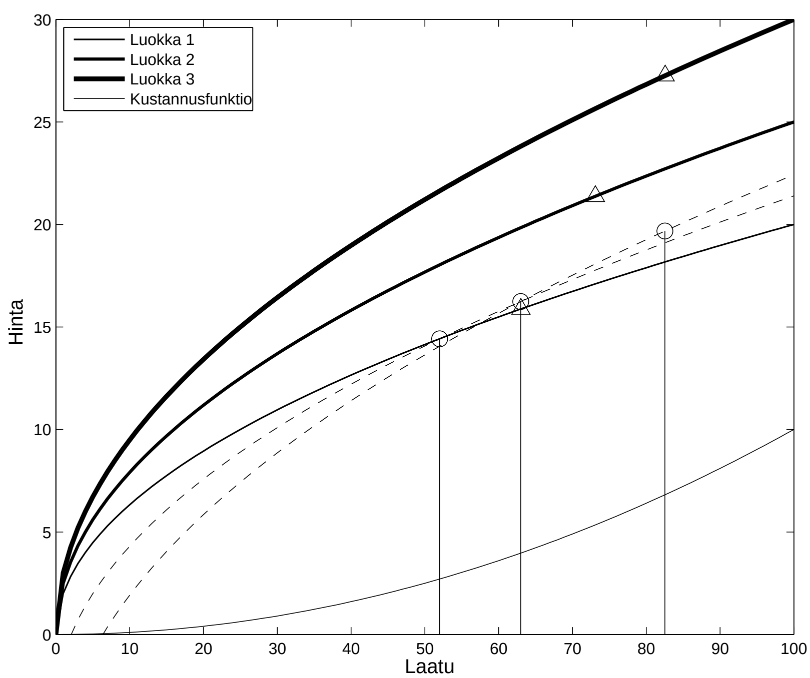
Palataan edellisen osion esimerkiin, mutta jätetään yksinkertaisuuden vuoksi toinen laatu-ulottuvuus pois. Tehtävälle saadaan ratkaisu q_1^*, q_2^*, q_3^*, ja p^*, joka on listattu taulukkoon 1 yhdessä hyötyfunktioiden parametrien c_i ja painokertoimien t_i kanssa. Ratkaisun kannalta tärkeät funktiot on kuvattu kuvassa 1. Eri paksuiset viivat kuvaavat luokkien hyötyfunktioita. Kolmioilla on merkitty tehtävän tehokas ratkaisu, ja ympyröillä optimaalinen ratkaisu. Katkoviivat kuvaavat asiakasluokkien 2 ja 3 indifferenssikäyriä, eli niitä tuotepaketteja, joilla he saavat saman ylijäämän kuin omalla paketillaan. Optimissa huomataan, että ratkaisussa kolmannelle luokalle myydään tehokasta laatua, mutta asiakasluokille 2 ja 3 myydään tehokasta laatua heikompaa laatua. Asiakasluokan 1 hintaa on jouduttu laskemaan, jotta 2 ryhmän tuote ei oli kiinnostavampi kun luokan oma tuote. Hintaa on laskettu niin paljon, että ratkaisussa 2 luokka on 3 luokan indifferenssikäyrällä. Vastaavasti käyttäytyy luokka 2, koska kokee luokan 1 tuotteen kiinnostavana. Luokka 1 ei koe muita luokkia kiinnostaviksi, jolloin se haluaa ainoastaan saada positiivisen hyödyn omasta tuotteestaan. Tästä syystä luokalle 1 myydään niin kallista tuotetta kun se suostuu ostamaan, eli tuoteen hinta on sama kuin hyötyfunktion arvo. Muut asiakasluokat taas saavat positiivista hyötyä.

Tuotteiden tehokkuus voidaan laskea ensin selvittämällä kunkin luokan tehokkaat laadut. Ratkaisemalla yhtälön $\nabla u_i(q_i) = \nabla c(q_i)$ saadaan laaduiksi $q_1 = 63.0, q_2 = 73.1$ ja $q_3 = 82.5$. Myymällä suurinta mahdollista hintaa, saadaan kokonaistuotoksi $\pi^* = 60.3$. Optimaalisen ratkaisun mukainen tuotto on $\pi = 48.6$, eli tuottoisuus on tällöin 80.6%. Asiakkasluokkien tuotteiden tehokkuudeksi saadaan $\Phi_1 = 98.4\%, \Phi_2 = 99.0\%$ ja $\Phi_3 = 100\%$. Kolmannelle luokalle myydään tehokasta laatua, kun taas muille ryhmille tehokkuudesta on jouduttu hieman tinkimään. Kaikkien asiakasluokkien yhteenlaskettu ylijäämä on $u_{tot} = 11.2$, jolloin kokonaistehokkuus on $\Phi_{tot} = \dfrac{u_{tot} + \pi}{\pi^*} = 99.1\%$. Ratkaisu on siis kokonaismarkkinoiden kannalta varsin tehokas, huolimatta siitä, että kyseessä on monopolinen yritys. Yrityksen kannalta tilanne ei kuitenkaan ole yhtä lupaava, sillä IC-rajoitteiden vuoksi sen on luovuttava noin viidesosa voitosta, jonka se saisi ilman rajoitteita.

Taulukko 1: Esimerkkitehtävä

Asiakasryhmä	t	c_i	q^*	p
1	2	2	52.0	14.4
2	1	2.5	63.0	16.2
3	1	3	82.5	19.7

Kuva 1: Esimerkkitehtävän ratkaisu



2.3 Optimaalisuusehdot

Tehtävälle voidaan kirjoittaa tavanomaiset epälineaarisen rajoitetutun tehtävän optimaalisuusehdot, [8]

$$
\begin{aligned}
t_i + \sum_{k \neq i} \lambda^{ki} &= \lambda^i + \sum_{j \neq i} \lambda^{ij}, \ \forall i && (1) \\
t_i \nabla c(q_i^*) + \sum_{k \neq i} \lambda^{ki} \nabla u_k(q_i^*) &= \nabla u_i(q_i^*) \left(\lambda^i + \sum_{j \neq i} \lambda^{ij} \right) && (2) \\
\lambda^{ij}(u_i(q_j^*) - u_i(q_i^*) + p_i^* - p_j^*) &= 0, \ \forall i \neq j && (3) \\
\lambda^i (p_i^* - u_i(q_i^*)) &= 0, \ \forall i && (4) \\
\lambda^i, \lambda^{ij} &\geq 0, \ \forall i, j, && (5)
\end{aligned}
$$

missä λ^i:t ovat IR-rajoituksia vastaavat Lagrangen kertoimet ja λ^{ij} IC-rajoituksien kertoimet. Yhtälöt 3 ja 4 aiheuttavat sen, että ainoastaan aktiivisia rajoitusehtoja vastaavat Lagrangen kertoimet voivat olla nollasta poikeavia, jolloin ne voivat vaikuttavaa yhtälöissä (1) ja (2).

Sijoittamalla yhtälö (1) yhtälöön (2) ja järjestelemällä termit uudestaan, saadaan yhtälö

$$
t_i(\nabla u_i(q_i^*) - \nabla c(q_i^*)) = \sum_{k \neq i} \lambda^{ki} (\nabla u_k(q_i^*) - \nabla u_i(q_i^*)). \tag{6}
$$

Yhtälön vasen puoli on sama, kuin tehtävän ratkaisu ilman IC-rajoitteita. Yhtälöstä nähdään, että rajoitusehtojen ollessa voimassa ($\lambda^{ij} > 0$) syntyy tehottomuutta yhtälön oikean puolen verran. Tehottomuuden suuruus riippuu asiakasryhmän ja tehottomuutta aiheuttavan ryhmän hyötyfunktioiden gradienttien erotuksesta, sekä vastaavan Lagrangen kertoimen suuruudesta [5].

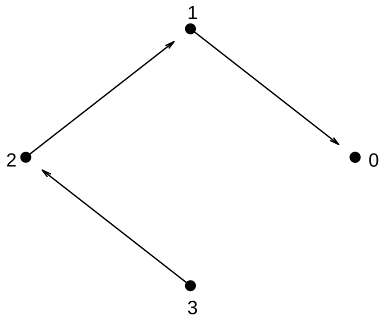
3 Tehtävän ratkaiseminen

3.1 Digraafit

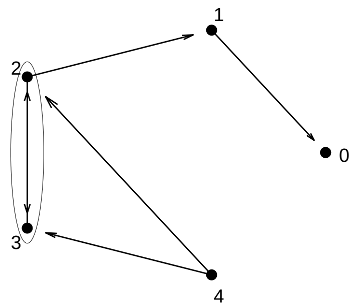
Digraafien avulla voidaan tehtävän optimin ominaisuuksia kuvata yksinkertaisella suunnatulla verkolla. Digraafissa asiakasluokat on järjestetty aktiivisten rajoitusehtojen mukaan. Digraafista nähdään suoraan joitain optimin ominaisuuksia. Ensimmäisenä tämän lähestymistavan esitteli Nahata et al. [9, 10].

IC- ja IR-rajoitusehdot voitiin yhdistää pelkiksi IC-ehdoiksi luomalla ylimääräinen nolla-asiakasluokka, jonka hyöty on aina 0. Tällöin kaikki rajoitusehdot ovat muotoa “Asiakasluokka i ei saa saada parempaa ylijäämää kuin mitä se saisi luokan j tuotteesta”, missä $i, j \in I \cup \{0\}$. Valitaan näistä aktiiviset rajoitteet, eli ne jotka optimissa estävät paremman ratkaisun löytämisen, ja joita vastaava Lagrangen kerroin on positiivinen. Kuvataan asiakasryhmiä pisteillä ja aktiivisia rajotusehtoja nuolilla näiden pisteiden välillä. Esimerkiksi jos asiakasluokka 1 rajoittaa luokan 2 tuotetta, piirretään nuoli luokasta 2 luokkaan 1. Mikäli tehtävän ratkaisussa esiintyy niputtamista, merkitään sitä ympyröimällä niputetut luokat. Näin saadaan aikaiseksi suunnattu verkko, digraafi. Esimerkkejä digraafeista löytyy kuvaajista 3a - 3d.

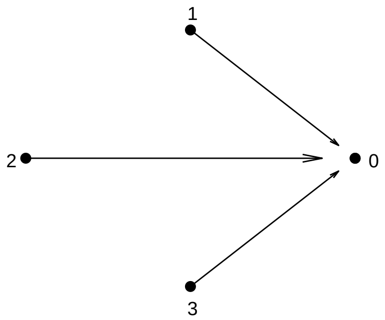
Kuva 2: Esimerkkejä digraafeista



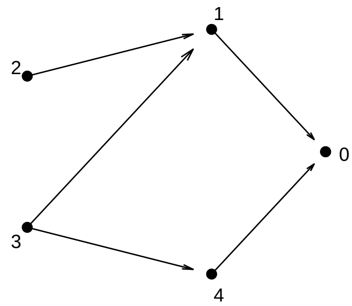
(a) Suora digraafi



(b) Niputtaminen



(c) Kaikki luokat riippumattomia toisistaan



(d) Haarautunut digraafi

Digraafin avulla saadaan optimaalisuusehtojen yhtälölle (1) tulkinta. Jokainen luokka voidaan ajatella lähteenä, jonka suuruus on luokan koko t_i. Lisäksi luokkien välille syntyy virtoja, joiden suuruus on vastaava Lagrangen kerroin λ^{ki}. Ainoa nielu graafissa on nolla-asiakasluokka. Yhtälö (1) sanoo, että luokkaan tulevien virtojen suuruus on sama kuin sieltä poistuvien virtojen suuruus.

Digraafin on siis oltava sellainen, että jokaisesta luokasta on päästävä nuolia seuraten nolla-luokkaan. Ylimmäksi luokaksi kutsutaan sellaista, johon ei tule yhtään virtaa, eli se ei rajoita muita luokkia. Luokka on toisen yläpuolella, jos siitä pääsee virran mukana toiseen luokkaan, ja vastaavasti virtaa pitkin päästään aliluokkaan. Koska nolla-luokka oli ainoa mahdollinen nielu, graafin alin luokka on aina nolla-luokka.

Mikäli digraafin rakenne on tarpeeksi yksinkertainen, voidaan Lagrangen kertoimet laskea suoraan. Esimerkiksi kuvan 3a kaltaisessa tilanteessa, jossa jokaisessa luokalla on vain yksi virta sisään ja yksi ulos, saadaan ylintä luokkaa vastaava kerroin suoraan ylimmän luokan koosta, eli painokertoimesta $\lambda^{32} = t_i$. Toiseksi ylimmän virran vastaava kerroin saadaan laskemalla edellinen kerroin ja luokan koko yhteen, eli $\lambda^{21} = \lambda^{32} + t_2$. Näin jatketaan kunnes ollaan nollaluokassa. Viimeistä kerrointa vastaa siten kaikkien luokkien yhteenlaskettu koko, $\sum_{i \in I} t_i$.

Edellisen kaltainen suora digraafi saadaan esimerkiksi olettamalla niin kutsuttu single-crossing -ominaisuus hyötyfunktioille. Tämä tarkoittaa että hyötyfunktiot toteuttavat ehdon [9]

$$u_i(q_1) - u_i(q_2) > u_{i-1}(q) - u_{i-1}(q_2) \tag{7}$$

kaikilla q_1, q_2, s.e. $q_1^k < q_2^k \quad \forall k \in D$ kun $i = 2, ..., n$. McAfee ja MacMillan formuloivat tämän ehdot moniulotteisena [11], mutta useammassa ulottuvuudessa oletus ei ole kovin mielekäs, sillä on hyvin tavallista, että hyötyfunktiot eivät toteuta tätä ehtoa [5]. Asiakasluokat voidaan nimittäin järjestää jossain ulottuvuudessa kuten yksiulotteisessa tapauksessa, mutta järjestys voi olla eri jossain toisessa ulottuvuudessa.

Jossain tapauksissa, vaikka digraafi ei olisi täysin suora, voidaan ainakin osa Lagrangen kertoimista laskea. Esimerkiksi, mikäli digraafi muodostuu useasta osa-graafista siten, että toisesta osiosta pääsee alempaan osioon voidaan ylempi osio käsitellä omana kokonaisuutenaan, jonka yhteinen valuma aliosioon on luokkien painokertoimien summa. Näin käy esimerkiksi kuvan 3b tilanteessa, kun kaikki virrat kulkevat luokan 1 kautta. Tämänkaltaisten jaotteluiden avulla tehtävä voidaan jakaa useaan osa-tehtävään, joiden erillinen laskeminen voi olla nopeampaa kuin kaikkien laskeminen kerralla.

Yhtälön (6) perusteella voidaan sanoa, että ylimmälle luokalle myydään aina tehokasta laatua [10]. Tuotteen hinta ei kuitenkaan välttämättä ole tehokas, vaan määräytyy aliluokkien perusteella. Yliluokan on oltava vähintään indifferentti aliluokkiensa kanssa. Usein se on kuitenkin indifferentti ainoastaan muutaman luokan kanssa, sillä alimpien luokkien tuotepaketit ovat selkeästi huonompia. Luokista heti nolla-luokan yläpuolella taas voimme sanoa, että niiden ylijäämä on nolla, sillä ne ovat indifferenttejä nollahyödyn kanssa. Kuvan 3c tilanteessa kaikki luokat ovat indifferenttejä nolla-luokan kanssa, ja monopoli saa suurimman mahdollisen tuoton, eli tehokkuus ja tuottoisuus ovat 100%.

Yhtälöstä (6) voidaan myös päätellä, että ylemmän luokan tuotto $\pi_i = p_i - c(q_i)$ on suurempi kuin sitä alempien luokkien tuotto [10]. Tästä johtuen, jos yläluokan koko (painokerroin) on huomattavasti suurempi kuin aliluokan, saattaa käydä niin, että aliluokalle ei kannata myydä lainkaan, sillä isolle yliluokalle voidaan nyt myydä hyvällä tuotolla paljon sen omaa tuotetta. Alemman tuotteen laatu on tehtävä niin heikoksi, että yläluokka ei missään nimessä halua ostaa halvempaa tuotetta.

3.2 Ratkaisumenetelmiä

Tehtävän ratkaisemiseen on kaksi perustekniikkaa: tavallisten epälineaaristen optimointitehtävien ratkaisualgoritmit sekä yhtälönratkaisualgoritmien solveltaminen suoraan optimaalisuusehtoihin. Epälineaaristen optimointitehtävien ratkaisualgoritmejä on kehitetty useita, mutta tässä työssä on käyttetty SQP-menetelmää. Kaikki ajot on suoritettu Matlab -laskentaohjelmiston avulla, ja optimintiin on käytetty Optimization ToolBoxin fmincon- ja fsolve-funktioita.

SQP ratkoo optimaalisuusehtoja tekemällä niistä toisen asteen approksimaation ja tämän jälkeen laskemalla Newton-iteraatiolla approksimaatiolle ratkaisun. Tämän jälkeen approksimaatio uusitaan, ja iteraatiota toistetaan kunnes ollaan tarpeeksi lähellä oletettua optimipistettä. Toisen asteen approksimaatiota varten tarvitaan Hessen matriisi, joka SQP:ssä lasketaan funktion gradienttien avulla [12] Tarkemmin SQP:n teoriasta voi lukea kirjasta [12] ja toteutuksesta The MathWorksin Internet-ohjeesta [13]. Koska menetelmä ratkoo optimaalisuusehtoja, se ratkoo jokaisen tehtävän optimointavan parametrin lisäksi yhden Lagrangen kertoimen jokaista rajoitusehtoa kohden. Tällöin yhteensä ratkottavia muuttujia on $N = n(d + 1) + n^2$, jolloin muun muassa Hessen matriisin koko on $N \times N$, eli siinä on $O(n^4)$ alkiota. Mikäli N on iso, laskut voivat olla todella hitaita.

Kuten suurin osa muistakin epälineaarisista optimointimenetelmistä, SQP on paikallinen hakumenetelmä. Tämä tarkoittaa sitä, että se löytää ainoastaan paikallisia optimipistetä, eikä aina välttämättä globaalia optimia. On siis selvitettävä, onko löydetty optimi ainoastaan paikallinen vai globaali optimi. Tämän selvittämiseksi on olemassa muutamia menetelmiä, mutta ne eivät anna täysin varmoja tuloksia.

Tässä työssä globaalia optimaalisuutta tutkittaan ratkaisupintojen avulla: Ensiksi lasketaan oletettu optimi, jonka jälkeen ratkaisuvektorin alkioita varioidaan satunnaisesti tai tietyn hilan mukaan. Tämän jälkeen lasketaan kohdefunktion arvo uudessa pisteessä, ja piirretään kuvaajaan tätä pistettä vastaava piste. Saadaan pinta, josta voidaan helposti nähdä, onko saatu optimi alueen korkein kohta.

Ongelmia kuitenkin ilmenee, kun optimoitavia parametrejä on enemmän kuin kaksi. Tällöin kuvan piirtäminen hankaloituu huomattavasti, ja on siirryttävä kuvaamaan esimerkiksi uuden pisteen etäisyyttä oletetusta optimista. Samoin kun optimoitavien muuttujien määrä kasvaa, tarvittava määrä uusia arvottavia pisteitä kasvaa. Lisäksi rajoitusehtojen määrän kasvaessa käypien pisteiden määrä tippuu huomattavasti. Ratkaisupinnan piirtäminen liian pienellä aineistolla ei taas anna luotetavaa kuvaa optimin laadusta.

Optimin laatua voidaan tutkia myös aloittamalla iteraatio useasta eri alkupisteestä. Optimointialgoritmit vaativat aina aloituspisteen, josta ne aloittavat iteroinnin. Lokaalit algoritmit kuten SQP hakeutuu aloituspisteestä johonkin lokaaliin optimiin. Kokeilemalla eri alkuarvauksia, saattaa iteraatio päätyä eri optimiin. Alkuarvauksella saattaa olla myös vaikutus tarvittavien iteraatioiden määrään. Hyvällä alkuarvauksella optimi löytyy nopeammin.

Tehtävän voi ratkaista myös löytämällä optimaalisuusehdoille ratkaisun. Suurin osa numeerisista algoritmeista toimii siten, että ne haluavat kohdefunktion joka on muotoa $f(x) = 0$. Tämän jälkeen käytetään jotain epälineaarista optimointimenetelmää löytämään minimi f:n itseisarvolle. Sinällään menetelmä ei ole kovin tehokas tämän tehtävän ratkaisemiseksi, mutta mikäli ratkaisusta tiedetään jotain parametreja jo valmiiksi, voidaan ehdot ratkaista tehokkaammin kuin suoralla epälineaariselle optimointimenetelmällä. Hinnoittelutehtävän tapauksessa voidaan esimerkiksi digraafista päätellä osa Lagrangen kertoimista. Niitä ei tarvitse siten enää laskea uudestaan.

Myös suoria ratkaisumenetelmiä voidaan muokata niin, että ne hyödyntävät tunnettua tietoa. Tämän tehtävän tapauksessa, jos digraafi tunnetaan, voidaan osa rajoitusehdoista kytkeä pois päältä. Kytkemällä turhat rajoitteet pois, optimi ei muutu, mutta laskennallisesti tehtävä helpottuu sillä matriisien koot pienenevät ja laskutoimistusten määrä laskee merkittävästi. Rajoitusehtoja voidaan kytkeä päälle myös vaiheittain ja käyttää uutta edellisen vaiheen optimia seuraavan alkuarvauksena. Tällöin optimoititehtävien määrä kasvaa, mutta ne voivat olla nopeampia kun yksi vaikeampi tehtävä.

Ongelmaksi muodostuu nyt kuitenkin se, että digraafia ei tunneta, ennen kuin optimi on jo selvitetty. Digraafin määräävät hyöty- ja kustannusfunktiot, sekä luokkien koot, mutta graafin tarkkaa muotoa ei osata vielä ennustaa. Jonkinlaisia ennusteita digraafin tai sen osien rakenteesta voidaan kuitenkin tehdä. Hyötyfunktioilta voidaan esimerkiksi vaatia niin kutsuttu single-crossing -oletus, joka takaa sen, että luokkien digraafi on suora ja Lagrangen kertoimet tunnetaan.

Mikäli käytetään muotoa $u_i = \sum_{k \in D} c_{i,k} \sqrt{q_k}$ olevia hyötyfunktioita, voidaan parametrien $c_{i,k}$ avulla päätellä mitkä rajoitusehdot ovat aktiivisia. Niiden asiakasluokkien, joita vastaavat parametrit ovat jossain dimensiossa lähellä toisiaan, väliset rajoitusehdot ovat todennäköisemmin aktiiviisia kuin niiden jotka eivät ole.

Mikäli tehtävä on likimain sama kuin jokin aikasemmin ratkaistu tehtävä, on todennäköistä, että uuden tehtävän digraafi on hyvin samankaltainen kuin edellisen. Tällöin vanhaa ratkaisua kannattaa käyttää alkuarvona uudessa optimointitehtävässä, sillä kuten myöhemmin huomataan, alkuarvauksen vaikutus tehtävän ratkaisuun on huomattava. Tämän kaltainen tilanne syntyy helposti esimerkiksi sähkön hinnoittelussa, jossa päivittäinen sähkön kysyntä ja tarjonta pysyy likimain vakiona, mutta pieniä eroja kuitenkin esiintyy.

Toisena ongelmana tehtävän ratkaisussa tulee hyvin nopeasti vastaan myös sen koko. Optimoitavien muuttujien määrä on suoraan verrannollinen asiakasluokkien määrään n ja laatu-ulottuvuuksien määrään d. Jokaisella asiakasluokalla on d muuttujaa. Lisäksi jokaiselle luokalle on omat hintansa, jolloin muuttujia on kokonaisuudessaan $n(d+1)$. Tämän lisäksi rajoitusehtojen lukumäärä kasvaa asiakasluokkien määrän mukaan voimakkaasti. Jokaista asiakasluokkaparia kohden tarvitaan kaksi rajoitusehtoa. Kun tähän vielä lisätään IR-rajoitteet, saadaan yhteensä n^2 rajoitusehtoa. Näin ollen tehtävän koko kasvaa hyvin voimakkaasti. Koska rajoitusehtojen määrä riippuu vain asiakasluokkien määrästä, ja vieläpä hyvin voimakkaasti, on asiakasluokkien määrä kriittisempi tarkasteltaessa esimerkiksi laskentaan tarvittavaa aikaa tai muistinkulutusta. Juuri tehtävän koosta johtuen on tärkeää voida kytkeä tarpeettomat rajoitusehdot pois päältä, jolloin muun muassa laskennassa tarvittavien matriisien koot pienevät.

4 Laskenta-aika

4.1 Asiakasryhmien ja laatu-ulottuvuuksien lukumäärän vaikutus

Asiakasluokkien ja laatu-ulottuvuuksien lukumäärät (n ja d) määräävät tehtävän koon. Rajoitusehtoja on n^2 ja muuttjia $n(d+1)$. Useissa sovelluksissa voidaan olla tilanteessa, jossa vähintään toinen kokoparametreista on useita kymmeniä tai jopa satoja. Vaadittavan laskenta-ajan riippuvuutta tehtävän koosta voidaan selvittää yksinkertaisesti ratkaisemalla tehtävää eri parametrien määrällä. Valitaan hyötyfunktioiksi $u_i = \sum_{k \in D} c_{i,k} \sqrt{q_k}$, ja kustannusfunktioksi $c = 0.001q^2$. Painokertoimet ovat $t = (n, ..., 2, 1)$. Parametrit $c_{i,k}$ arvotaan täysin satunnaisesti väliltä $[3, 6]$. Valitaan laskentahila siten, että asiakasluokkien lukumäärä n käy läpi arvot 1-13 kolmen välein, sekä laatu-ulottuvuudet 5-100 viiden yksikön välein. Jokaisessa hilapisteessä arvotaan uudet parametrit, ja toistetaan mittaus 5 kertaa satunnaisten virheiden vähentämiseksi. Mikäli jokin iteraatio kestää yli 25 minuuttia, iteraatiota ei suoriteta enää pidemmälle, vaan siirrytään seuraavaan.

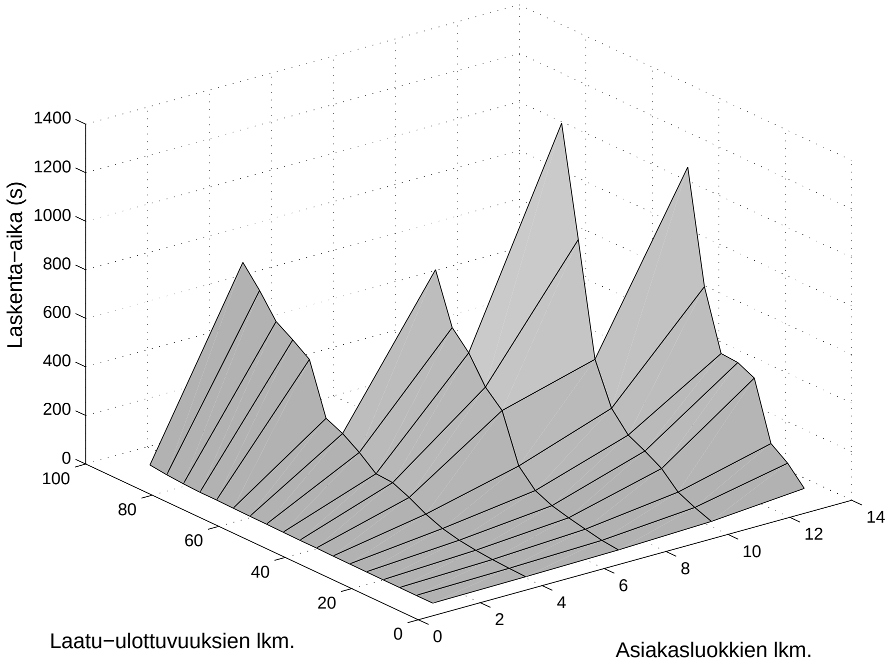
Ajosta saatiin seuraavanlainen, kuvan 3 mukainen tulos. Huomataan, että asiakasluokkien määrän kasvaessa, tehtävän laskenta-aika kasvaa huimasti. Sovittamalla polynomipinnan PNS-menetelmällä kuvaajalle, saadaan molempien parametrien suhteen melko tarkasti toisen asteen riippuvuus. Vaikka riippuvuus on samaa muotoa kummankin parametrin suhteen, on huomattava, että riippuvuus laatu-ulottuvuuksien määrästä on huomattavasti heikompi. Sama laskenta-ajan kasvu, joka aiheutuu asiakasluokkien määrän kaksinkertaistamisesta, saavutetaan viisinkertaistamalla laatu-ulottuvuuksien määrä. Esimerkiksi tilanteessa $n = 10, d = 5$ ja $n = 5, d = 25$, laskenta-ajat ovat likimain samat, vaikka toisessa on yli kaksinkertainen määrä muuttujia. Ero selittyy sillä, että ensimmäisessä tilanteessa on 100 rajoitusehtoa, ja toisessa vain 25, ja SQP:n on lisättävä tehtävään jokaista rajoitusehtoa kohden yksi muuttuja. Tällöin kokonaismuuttujamäärät ovat likimain samat. Kokonaismuuttujamäärä ei kuitenkaan riitä selittämään laskenta-aikaa, mutta sen avulla voidaan tehdä suuntaa antavia arvioita.

4.2 Tunnetut Lagrangen kertoimet tai digraafi

Jos tehtävän digraafi ja vastaavat Lagrangen kertoimet tunnetaan, ei niitä tarvitse enää laskea uudestaan, vaan optimointia voidaan helpottaa. Optimaalisuusehtojen Lagrangen kertoimet voidaan korvata vakioilla, jolloin muuttujien määrä tippuu $n(d+1)$:een. Optimaalisuusehdot voidaan tämän jälkeen ratkaista yhtälönratkaisualgoritmeilla. Tässä työssä yhtälön ratkaisemiseen on käytetty Matlab-ohjelmiston fsolve -funktiota, jonka tarkemmasta toiminnasta voi lukea The MathWorksin Intenet-ohjeesta [13].

Tätä voidaan kokeilla siten, että ensin ratkaistaan tehtävä jollain toisella menetelmällä, minkä jälkeen ratkaistaan laadut uudestaan optimaalisuusehdoista. Huomat-

Kuva 3: Asiakasluokkien ja laatu-ulottuvuuksien määrän vaikutus laskenta-aikaan.



tavaa on, että mikäli Lagrangen kertoimet tunnetaan, kaksi ensimmäistä optimaalisuusehtoa, (1) ja (2), määräävät täysin tuotteiden laatuparametrit, kun taas tämän jälkeen hinnat ovat laskettavissa ehdoista (4) ja (3).

Käyttämällä jälleen hyötyfunktioina satunnaisesti väliltä $[3, 6]$ generoituja neliöjuurimuotoisia hyötyfunktioita, kustannusfunktiona $c = 0.001q^2$ ja painokertoimina tasajakoa alkaen n:stä yhden välein 1 asti, saadaan SQP:n ja yhtälönratkaisijan kolmeenkymmenteen iteraatioon keskimäärin käyttämänkäyttämien aikojen suhteeksi 10-40%, riippuen tehtävän koosta. Suurilla tehtävillä funktioratkaisija on suhteessa nopeampi tapa kuin suora tehtävän ratkaiseminen. Yhtälöratkaisija ei kuitenkaan pystynyt ratkaisemaan kaikkia tehtäviä lainkaan, vaan jäi joka viidennellä hyötyfunktioparvella hyvin kauas optimista. Nämä jätettiin pois laskenta-aikojen keskiarvosta, mutta selvästi huomattiin, että suoran yhtälöratkaisimen käyttö vaatii myös hieman lisävirittelyjä, mikäli sen toimintavarmuutta halutaan kasvattaa.

Toinen ongelma yhtälönratkaisimen käytössä on se, että Lagrangen kertoimet tunnetaan ainoastaan silloin, kun digraafi on suora ketju. Esimerkiksi edellisen ajon kaltainen tilanne ei siis ole millään muotoa mielekäs. Tätä menetelmää voidaan kuitenkin soveltaa osana muita menetelmiä, mikäli digraafi tunnetaan, tai sen rakenne on sellainen, että ainakin osa Lagrangen kertoimista pystytään laskemaan. Mitä enemmän Lagrangen kertoimia tunnetaan, sitä vähemmän ratkottavia parametrejä jää.

Matlab-ohjelmiston profile viewer -työkalun avulla voidaan selvittää mikä osa ohjelmaa vaatii eniten prosessointiaikaa. Tarkastelu osoittaa, että juuri SQP:n suorittamat Hessen matriisin laskutoimitukset, kuten QR-hajotelman tekeminen, vaativat ylivoimaisesti suurimman osan ohjelman suoritusajasta. Suurin osa matriisilaskutoimituksia ovat kompleksisuudeltaan $O(N^2)$ tai $O(N^3)$, jolloin on kannattavampaa esimerkiksi jakaa ongelma kahteen osaan, ja ratkoa molemmat erikseen. Tämän vuoksi olisi pyrittävä pienentämään Hessen matriisin kokoa, vaikka siitä seuraisikin useampi erillinen laskentavaihe.

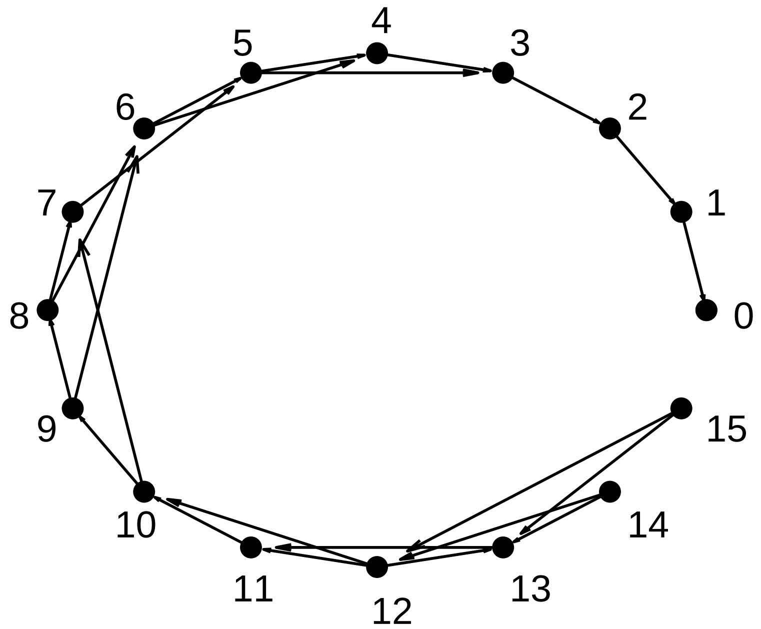
Hessen matriisin koon määräävät tehtävän rajoitusehdot ja muuttujat. Mikäli tiedämme valmiiksi digraafin rakenteen, voidaan osa rajoitteista kytkeä pois päältä. Tällöin myös algoritmin käyttämien muuttujien määrä laskee ja Hessen matriisi pienenee. Optimaalinen tilanne luultavasti olisi, että etukäteen tunnettaisiin koko digraafi. Tilanne on kuitenkin vain harvoissa tapauksissa näin selkeä, sillä usein digraafi tunnetaan vain likimain. Tätä tietoa voidaan hyödyntää niin, että kytketään ne rajoitteet päälle, jotka ovat todennäköisesti aktiivisia. Ratkaistaan tehtävä, minkä jälkeen lisätään joukko uusia rajoitteita, mutta käytetään edellisen vaiheen ratkaisua uutena alkuarvauksena.

Valitaan asiakasluokkien lukumääräksi 15 ja laatu-ulottuvuuksien määräksi 3, sekä jälleen neliöjuurimuotoiset hyötyfunktiot $u_i = \sum_{k=1}^{d} c_{i,k}\sqrt{q_k}$ ja neliöllinen kustannusfunktio $0.001q^2$. Painokertoimet edelliseen tapaan $t = (n, ..., 2, 1)$. Valitaan hyötyfunktioiden parametrit nyt niin, että ensimmäinen ulottuvuus on tasajaolla välillä $[3, 6]$. Muut ulottuvuudet määritellään samoin, mutta lisätään jokaiseen parametriin satunnainen komponentti $\frac{5}{n}r$, missä r on normaalijakaantunut satunnaismuuttuja odotusarvolla 0 ja keskihajonnalla 1. Tällöin digraafista tulee suora ketju, mutta riippuen hieman arvotuista parametreistä muitakin rajoitusehtoja voi olla aktiivisena. Luokalle on tyypillisesti aktiivisena rajoite seuraavaan, mutta myös kahdesta neljään seuraavaan luokkaan. Seuraavassa ajossa arvottujen parametrien mukaisen tehtävän digraafi on esitetty kuvassa 4.

Ratkaistaan tehtävä siten, että aloitetaan täysin ilman IC-rajoitteita. Tämän jälkeen otetaan rajoitteita käyttöön siten, että jokaiselle luokalle lisätään s kappaletta sen naapuriluokkia koskevia rajoitteita, eli esim. kun $s = 2$ luokalle 4 lisätään rajoitteet luokkiin 2, 3, 5 ja 6. Seuraavalla iteraatiolla lisätään jälleen s rajoitetta kummallekin puolelle lisää ja näin jatketaan kunnes kaikki rajoitteet ovat käytössä. Mikäli lisäyksen jälkeen edellisestä iteraatiosta saatu piste on edelleen käypä, voidaan sen kierroksen yli hypätä, sillä optimi ei parane lisäämällä rajoitteita.

Edellisen kaltainen ajo tuotti kuvan 5 kaltaiset tulokset. X-akselilla on käytetty askelpituus s ja y-akselilla käytetty aika. Pylväiden eri kerrokset kertovat kunkin iteraation keston. Käyttämällä askelpituutta $s = 1$ aikaa kuluu suhteellisen paljon, mutta askelpituudella 3 voidaan säästää huomattavat määrät aikaa laskennassa. Digraafista nähdään, että yhdelle asiakasluokalle on aktiivisena rajoitusehto korkeintaan 3 luokan päähän. Tämä tarkoittaa sitä, että askelpituudella $s = 3$ saadaan käytännössä koko tehtävä ratkaistua jo ensimmäisellä iteraatiolla, eikä viimeisillä iteraatioilla ei tarvitse enää tehdä mitään. Myös askelpituuksilla, jotka ovat suurempia kuin 3, tarvitaan vain yksi iteraatio, mutta tällöin aktiivisia rajoitteita on

Kuva 4: Tehtävän digraafi.

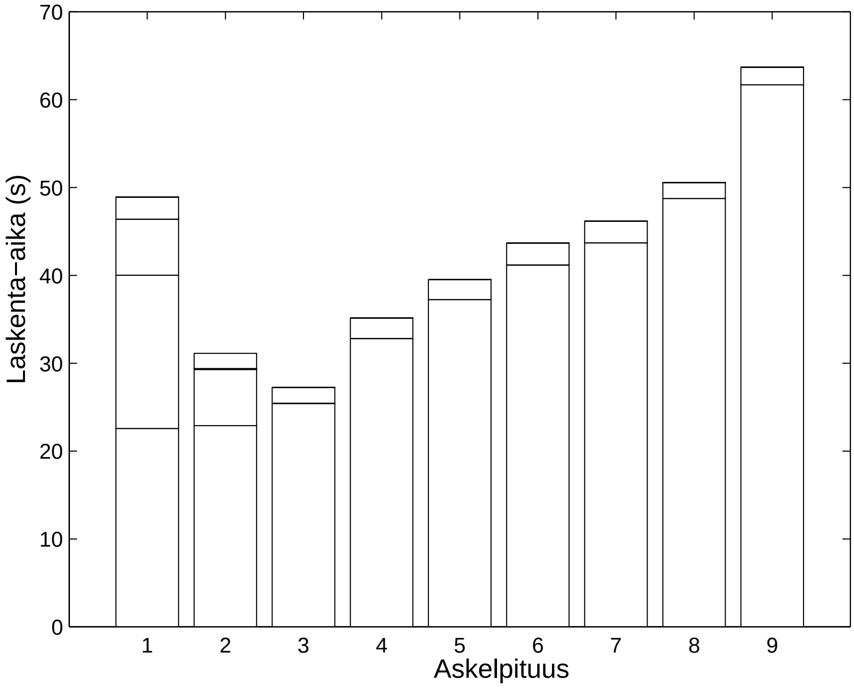


ensimmäisellä askeleella enemmän, ja laskenta on hitaampaa. Pienemmillä askelpituuksilla taas tarvitaan muutamia varsinaisia iteraatioita, vaikkakin ne ovat lyhyempiä kuin askelpituuden 3 ainoa iteraatio. Kokonaisaika jää tässä tapauksessa hieman korkeammaksi. Joka pylvään päällä oleva ylin kerros johtuu siitä, että riippumatta siitä onko viimeisen iteraation aloituspiste käypä, tehtävä ratkaistiin lopuksi kaikilla rajoitteilla, jotta varmistuttiin siitä, että aikasemmin löydetty optimi todella oli optimi.

Parhaiten tämän menetelmän hyödyn huomaa vertaamalla sitä kuvan 3 laskenta-aikoja tämän menetelmän laskenta-aikoihin. Kuvan mukaan 13 asiakasluokan ja viiden laadun tehtävän laskeminen kestää yli 2 minuuttia. Vaiheittaisella ratkaisemisella saman tehtävän ratkaisee 20-30 sekunttiin. Erittäin suuret tehtävät, kuten 40 asiakas-luokan ja 40 laatu-ulottuvuuden tehtävä kestää tällä menetelmällä noin kymmenen minuuttia, kun taas suoralla ratkaisemisella 16 tunnin odottelun jälkeen iteraatio ei ollut edes päässyt ensimmäistä askelta pidemmälle.

Parhaisiin tuloksiin päästäisiin, mikäli edelliset kaksi menetelmää yhdistettäisiin. Tämä kuitenkin vaatii sen, että digraafin rakennetta osattaisiin ennustaa tarkemmin. Jonkinlaista suuntaa mahdollisista aktiivisista rajoitteista saa esimerkiksi neliöjuurifunktioiden tapauksessa parametrien c_i^k vertailulla. Mikäli hyötyfunktiot muistuttavat toisiaan jossain dimensiossa, ne ovat todennäköisemmin yhdistettynä digraafissa. Pelkästään valitsemalla joka ulottuvuudesta luokat, joiden parametrit ovat lähinpänä toisiaan välille rajoitusehdon, päästään usein 50-70% oikeellisuuteen digraafien

Kuva 5: Vaiheittaisen ratkaisun vaatima laskenta-aika eri askelpituuksilla.



suhteen. Tämä ei kuitenkaan vielä riitä, vaan rajoitusehtoja tulee valituksi liian paljon ja lisäksi muutama aktiivinen ehto jää valinnan ulkopuolelle. Tästä syystä menetelmä häviää esimerkiksi edelläkuvatulle vaiheittainratkaisemiselle selvästi.

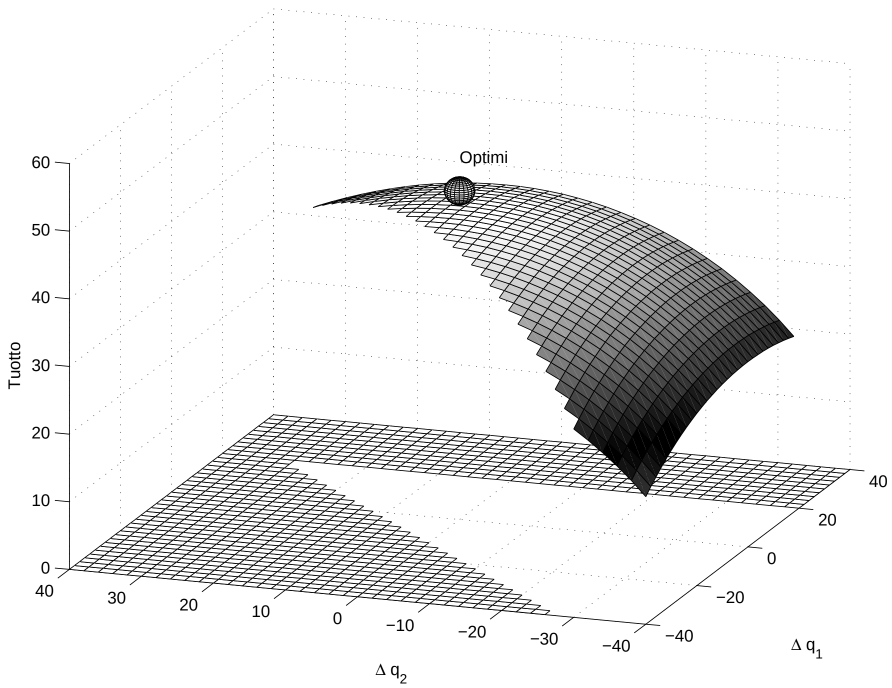
5 Ratkaisun yksikäsitteisyys

5.1 Ratkaisupintojen visualisointi

Tehtävän ratkaisupinnan visualisointia kokeiltiin muutamilla erilaisilla hyötyfunktiotyypeillä. Ensimmäisessä ajossa käytettiin kolmea asiakasluokkaa, joille määriteltiin yksiulotteiset hyötyfunktiot $u_1(q) = 2\sqrt{q}$, $u_2(q) = 2.5\sqrt{q}$ ja $u_3(q) = 3\sqrt{q}$. Luokkien koot olivat $t = (2, 1, 1)$ ja kustannusfunktio $c = 0.001q^2$. Tehtävän ratkaisu on esitetty taulukossa 1. Koska hyötyfunktiot toteuttavat single-crossing ehdon, kolmas asiakasryhmän laatu-parametri on optimissa aina sama.

Tarkastellaan tehtävän ratkaisupintaa. Lasketaan tuotto hilassa, jossa laadut q_1 ja q_2 poikkeavat optimista 40 yksikön päähän molempiin suuntiin. Jokaisessa pisteessä lasketaan tuotteille uudet optimaaliset hinnat, joilla asiakkaat vielä ostavat tuotteet. Näin varmistutaan, että piste on käypä. Kokonaistuotto on kuvattu kuvassa 6.

Kuva 6: Ratkaisupinta yksinkertaiselle tehtävälle



Huomataan, että ratkaisupinta muodostaa kolmion muotoisen kielekkeen, jota rajoittavat suorat $q_2 = q_3$ ja $q_1 = q_2$. Kielekkeen ulkopuolella tuotto tippuu roimasti, sillä tällöin asiakasryhmien laadut ja hyötyfunktiot eivät ole oikeassa järjestyksessä. Hyötyfunktiot toteuttavat single-crossing ehdon, jolloin ratkaisun digraafi on suora. Koska tällöin yliluokka sai aina parempaa tuottoa kuin aliluokka, on yhtälön (6) perusteella hinnat pakko tiputtaa nollaan, jotta asiakkaat ostaisivat tuotteensa. Sanotaan, että laadut eivät ole implementoitavissa. Kieleke taas on osa kupua, jonka huipulta löytyy optimipiste. Kupu on konkaavi ja optimointialgoritmin kannalta hyvin käyttäytyvä. Voidaan siis päätellä, että ratkaisu on globaali optimi.

Näin pienellä tehtävällä optimin laadun tutkiminen on suhteellisen helppoa, kun mahdollisia pisteitä ei ole kovin montaa. Ongelmia tuottaa kuitenkin tilanne, jossa dimensioita on useampia. Seuraavassa esimerkissä on laskettu neljän asiakasluokan tehtävä, jossa jokaisella luokalla on 3 laatu-ulottuvuutta. Hyötyfunktiona on käytetty $u_i = \sum_{k \in D} c_{i,k}\sqrt{q_k}$, jonka kertoimet $c_{i,k}$ on lueteltu taulukossa 2. Kustannusfunktio on $c = 0.001q^2$ ja luokkien painokertoimet $t = (4, 3, 2, 1)$. Tällä kertaa poikkeamat on arvottu siten, että jokainen parametri poikkeaa optimista satunnaisen luvun $\Delta q_{i,j} \in [-30, 30]\ \forall j \in D$ verran.

Kuvaajassa 7 on esitetty tuotto etäisyyden funktiona optimista. Huomataan, että vaikka löydetty optimi todella on tehtävän ratkaisu ja myös kuvaaja näyttää siltä, tämä menetelmä antaa vaan suuntaa siitä, miltä pinta todellisuudessa näyttä. Tilanteesta voisi piirtää useita kuvan 6 kaltaisia kuvaajia, mutta tällöin tämäkin tehtävä vaatisi vähintään 6 kuvaajaa, eikä niistä näkyisi kuin murto-osa kaikista variaatiokombinaatioista. Mikäli tehtävän koko kasvaa entisestään, kuvaajiakin tulee lisää.

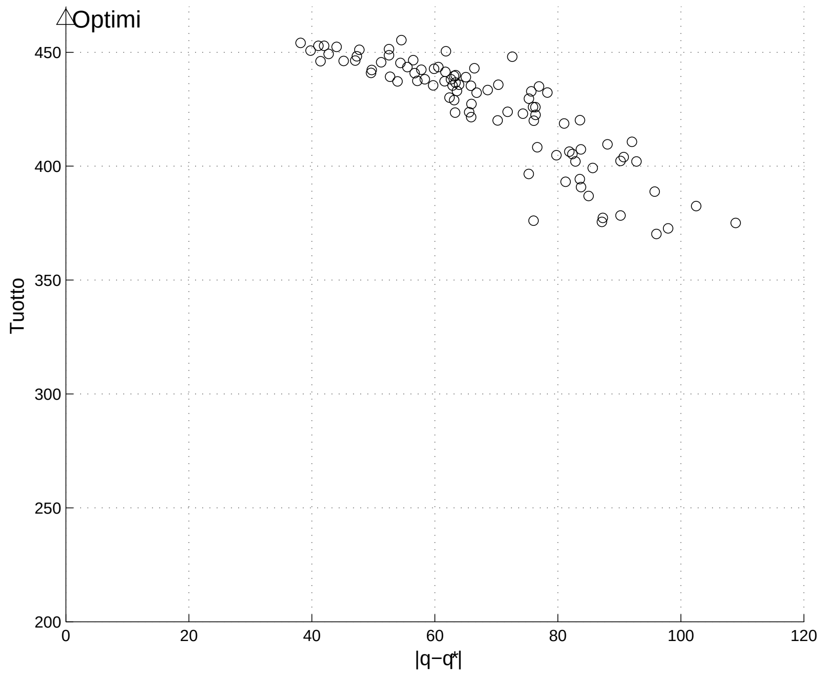
Kasvattamalla asiakasluokkien ja laatu-ulottuvuuksien määrää tilanne pysyy samankaltaisilla hyötyfunktioilla hyvin samanlaisena. Huomattavin ongelma on kuitenkin hylättyjen pisteiden määrässä. On hyvin tyypillistä, että vaikka jokaiselle pisteelle lasketaan uudet optimaaliset hinnat, arvotulle laatu-parametriyhdistelmälle ei löydy sellasia hintoja, joilla laadut voitaisiin implementoida. Erityisesti tehtävän kasvaessa, tarvittavien pisteiden määrä kasvaa, sillä arpomalla ei saada järkeviä pisteitä tarpeeksi paljon. Laatu-parametrien määrä on esimerkiksi kymmennellä asiakasluokalla ja laatu-ulottuvuudella 100. Käypien pisteiden löytäminen 100-ulotteisesta avaruudesta on hyvin vaikeaa. Jos sitä käydään läpi hilassa, jossa on ainoastaan kaksi pistettä, kaikkien pisteiden läpikäyminen vaatisi 2^{110} laskua. Tähän kuluisi aikaa vuosia, eikä kahden pisteen perusteella voida vielä sanoa mitään pinnanmuodoista.

Taulukko 2: Tehtävän parametrit

Asiakasryhmä	c_i^1	c_i^2	c_i^3
1	2.5	2.4	2.4
2	2.3	2.5	2.5
3	2.8	2.8	2.6
4	2.4	2.8	2.5

Tämän lähestymistavan pohjalta on siis vaikea sanoa mitään varmaa useampiulotteisten tehtävien ratkaisupinnoista, mutta tehtävät, joissa on vähemmän ulottuvuuksia, näyttävät muodostavan tietyillä hyöty- ja kustannusfunktiotyypeillä hyvinkäyttäytyviä pintoja. Tästä ei voi tietenkään olla täysin varma, mutta esimerkiksi neliöjuuri-, normaalijakauma-, ja normaalijakauman kertymäfunktion muotoisilla

Kuva 7: Ratkaisupinta useampiulotteiselle tehtävälle



hyötyfunktioilla tuotto - etäisyys -kuvaaja muistuttaa hyvin paljon kuvan 7 tapausta. Tällä menetelmällä ei tosin voi varmaksi osoittaa kuin että jollain hyötyfunktiotyypillä näin ei käy. Saadut tulokset kuitenkin viittaavat siihen, että mainituilla hyötyfunktiotyypeillä ratkaisu olisi yksikäsitteinen.

5.2 Alkuarvon vaikutus optimiin

Alkuarvon vaikutusta algoritmin antamaan optimiin voidaan tutkia aloittamalla iteraatio satunnaisista pisteistä ja vertailemalla eri alkuarvojen tuottamia lopputuloksia. Tätä kokeiltiin aluksi neliöjuurimuotoisilla funktioilla $u_i = \sum_{k \in D} c_{i,k}\sqrt{q_k}$, ja kustannusfunktiolla $c = 0.001q^2$. Asiakasluokkia oli 4 ja laatu-ulottuvuuksia 5. Parametrit on valittu satunnaisesti ja lueteltu taulukossa 4a ja luokkien painokertoimet olivat $t = (5, 4, 3, 2, 1)$. Kuvaajaan 8 on piirretty tuoton poikkeama optimituotosta alkuarvauksen etäisyyden funktiona oikeasta optimista. Ajossa käytettiin 400 iteraatiota, jolloin järkevien aloituspisteiden joukko tuli katettua suhteellisen hyvin. Havaitaan, että tuotto pysyy vakiona riippumatta alkuarvauksesta. Erot ovat suuruusluokkaa 10^{-3}, mikä johtuu lähinnä siitä, että algoritmi etsii ratkaisun vain tiettyyn tarkkuuteen asti. Käytännössä voidaan siis sanoa, että neliöjuurimuotoisilla hyötyfunktioilla optimi näyttää olevan riippumaton alkuarvauksesta. Tulostaa vahvistavat myös vastaavanlaiset ajot muunkokoisilla tehtävillä ja eritavalla arvotuilla parametreillä.

Taulukko 3: Tehtävän parametrit alkuarvokokeessa.

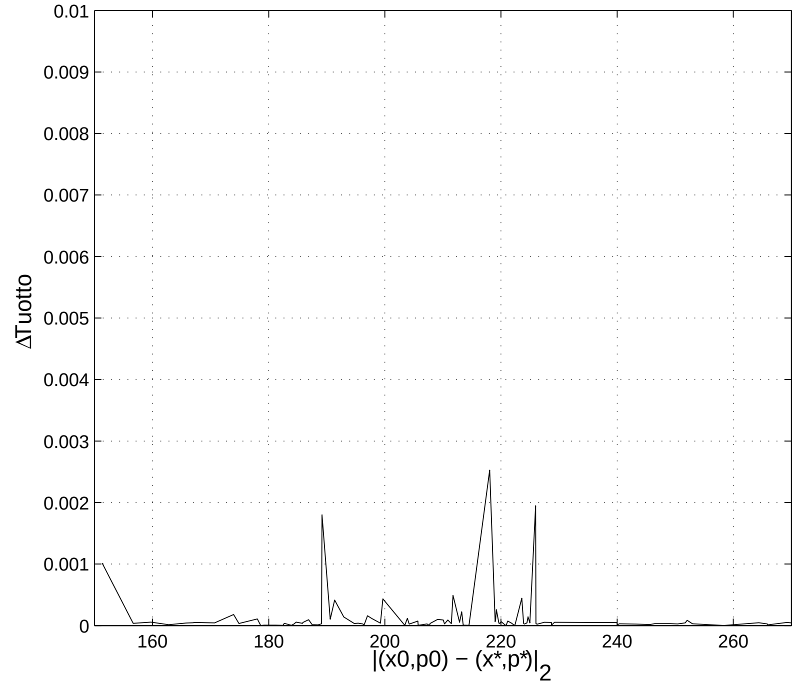
Asiakasryhmä	c_i^1	c_i^2	c_i^3	c_i^4
1	2.1	2.3	3.0	3.7
2	3.8	2.3	2.9	3.3
3	3.1	2.7	2.3	4.0
4	2.7	2.7	3.8	4.0

(a) Neliöjuurimuotoiset hyötyfunktiot

Asiakasryhmä	μ_i^1	μ_i^2	σ_i^1	σ_i^2
1	30	30	25	25
2	50	50	25	25
3	70	70	25	25

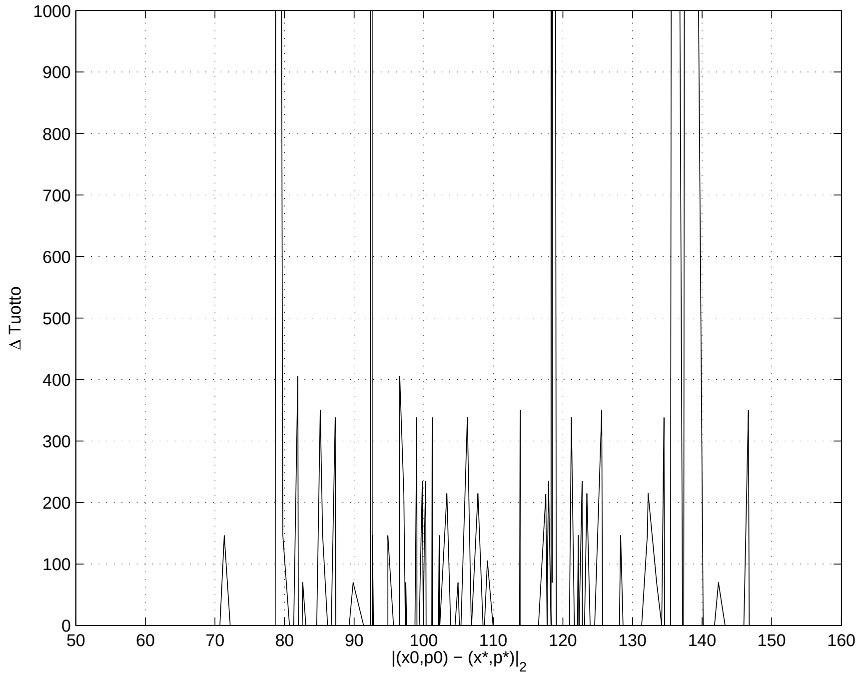
(b) Normaalijautuneet hyötyfunktiot

Kuva 8: Alkuarvon vaikutus optimin arvoon neliöjuurimuotoisilla funktioilla.



Vaihtamalla hyötyfunktiot normaalijakaantuneiksi huomataan, että tilanne ei ole aina näin yksinkertainen. Uudet hyötyfunktiot ovat $u_i = N_d(\mu_i, \sigma_i, q_i)$, missä $N_d(\mu, \sigma, q)$ on d-ulotteisen normaalijakauman tiheysfunktio. Yksinkertaisuuden vuoksi laatuulottuvuuksien määrä on tiputettu kahteen ja asiakasluokkien määrä kolmeen. Käytetyt parametrit on lueteltu taulukkoon 4b. Muut tehtävän funktiot ja parametrit on pidetty samoina. Nyt vastaavasta tuotto - etäisyyskuvaajasta 9 huomataan, että optimi ei ole enää sama kaikilla alkuarvauksilla, vaan vaihtelee voimakkaasti. Vaihteluväli on luokkaa $100 - 4000$, eikä 10^{-3} kuten edellä. Optimiratkaisu, joka on tämän tehtävän tapauksessa myös tehokas ratkaisu, ja sen läheltä löytyvät pisteet näyttävät edelleen suppenevan globalliin optimiin, mutta kauempaa lähtiessä optimin arvo on täysin satunnaista. Optimi siis löytyy, mikäli ollaan valmiiksi normaalijakauman huipun lähistöllä. Sen sijaan kaukaa hyötyfunktio näkyy algoritmille käytännössä pelkkänä nollana, eikä se löydä suuntaa, josta lähteä etsimään parempaa pistettä.

Kuva 9: Alkuarvon vaikutus optimin arvoon normaalijautuneilla hyötyfunktioilla.



Vastaavanlaisia tuloksia saadaan myös muunlaisilla hyötyfunktiotyypeillä. Normaalijakauman kertymäfunktio käyttäytyy hyvin pitkälle samoin kun neliöjuurifunktiotkin, mutta porrasfunktiot taas samoin kun normaalijakauman tiheysfunktio. Tämä on sinänsä luontevaa, sillä samalla tavalla käyttäytävät funktiot ovat silmämääräisesti lähellä toisiaan. Ominaisuudet jotka vaikuttavat siihen, miten herkkä algoritmi on alkuarvaukselle, ovat sileys ja tietynlainen kupumaisuus. Funktiot joissa on derivaatan epäjatkuvuuskohtia saattavat aiheuttaa ongelmia, sillä derivaatta muuttuu äkillisesti jossain pisteessä, eikä algoritmi osaa aavistaa sitä etukäteen. Kupumaisilla, eli konkaaveilla funktioilla taas ei ole tasaisia kohtia, jolloin samankaltaista ilmiötä kuin normaalijakauman yhteydessä ilmeni ei havaita lainkaan. Porrasfunktion arvo taas on laatu-asteikon alkupäässä ja loppupäässä vakio, jolloin suunnan löytäminen on lähes mahdotonta.

6 Yhteenveto

Epälineaarinen hinnoitteluongelman sovelluksia löytyy teollisuudesta ja useilta eri aloita. Usein sovellusten asettamat tehtävät ovat kuitenkin hankalahkoja, sillä ne vaativat useita laatu-parametrejä ja asiakasluokkien lukumäärä voi olla suuri. Tähän kun vielä liitetään se, että tämän tehtävän numeerisen ratkaisun vaikeus riippuu erittäin voimakkaasti tehtävän koosta, joudutaan kehittämään menetelmiä ratkaisun löytämiseksi tehokkaammin.

Työn alussa käsiteltiin ensiksi optimin yksikäsitteisyyttä. Huomattiin, että globaalin optimin löytyminen riippuu erityisesti käytettävistä hyötyfunktioista. Neliöjuurimuotoiset hyötyfunktioit käyttäytyivät hyvin sekä alkuarvo- että ratkaisupintatesteissä. Tästä voidaan päätellä, että tämäntyyppisten hyötyfunktioiden ratkaisut ovat erittäin todennäköisesti aina yksikäsitteisiä. Samaa ei voida kuitenkaan sanoa yleisesti hyötyfunktioista. Toinen perustyyppi hyötyfunktiosta, normaalijakautunut hyötyfunktio, ei aina tuota globaalia optimia. Ratkaisupinnan huippu ei ole selkeä ja eri alkuarvot saattavat tuottaa täysin eri optimeja.

Laskentaan vaadittavat ajat riippuvat asiakasluokkien ja laatu-ulottuvuuksien määrästä likimain suoraan verrannollisesti niiden neliöön. Asiakasluokkien vaikutus on kuitenkin moninkertainen verrattuna laatu-ulottuvuusien määrään. Tämä johtuu siitä, että asiakasluokkien lisääminen lisää tehtävän rajoitusehtojen määrää, jolloin algoritmin tarvitsemat matriisit kasvavat, ja laskenta hidastuu. Jokaista rajoitusehtoa ja muuttujaa kohden algoritmi tarvitsee yhden muuttujan, joka sen on ratkaistava.

Mikäli tehtävän digraafista osataan sanoa jotain etukäteen, voidaan tiedon avulla kertoa algoritmille osa tarvittavista Lagrangen kertoimista. Tällöin ratkaistavien muuttujien määrä laskee, ja tehtävä ratkeaa nopeammin. Lagrangen kertoimet tunnetaan kuitenkin käytännössä vain kahdessa tapauksessa: silloin kun jonkin asiakasluokan yläluokkien kaikki valuma tulee yhteen solmuun, josta lähtee ainoastaan yksi aktiivinen rajoitus seuraavaan solmuun, tai jokin rajoitus on varmasti inaktiivinen, jolloin vastaava Lagrangen kerroin on nolla.

Koska algoritmissa käsitellään matriiseja, joiden koko on muuttujien lukumäärän luokkaa ja suurin osa matriisien laskuoperaatioista vaatii $O(N^3)$ laskutoimitusta, kannattaa tehtävä pilkkoa useampaan pieneen tehtävään, mikäli vain mahdollista. Tämän vuoksi kannattaa osa rajoitusehdoista kytkeä pois päältä. Parhaisiin tuloksiin päästiin valitsemalla kullekin asiakasluokalle aktiiviset rajoitteet niin moneen naapuriluokkaan, että kaikki optimipisteessä aktiiviset rajoitteet on kytketty päälle. Tällöin monet aikaisemmin järkevässä ajassa ratkeamattomatkin tehtävät ratkeavat muutamissa minuuteissa.

Edellä kuvattu menetelmä valitsee kuitenkin ylimääräisiä rajoitusehtoja päälle. Laskenta-aikaa voitaisiin lyhentää valitsemalla aluksi juuri oikeat rajoitusehdot. Näiden ehtojen valitseminen on kuitenkin erittäin hankalaa, sillä digraafin rakennetta ei tunneta etukäteen.

Tästä voidaan jatkaa selvittämällä, minkälaisia digraafeja minkäkin tyypin hyötyfunktiot synnyttävät. Asiaa on jo tutkittu laajalti, mutta erityisesti useampiulotteiselle tehtävälle digraafityyppien selvittäminen ei ole vielä tuottanut juurikaan tulosta. Useamman asiakasluokan tapauksessa erilaisia digraafityyppejä on niin monta, että niiden läpikäyminen järjestyksessä ei ole järkevä vaihtoehto. Tyypillisten digraafien tyypit vaihtelevat sovellukohteittain.

Epälineaaristen hinnoittelutehtävien tutkimuksen seuraavia suuria haasteita ovat moniulotteisen tehtävän ratkaiseminen epätäydellisen informaation avulla. Tässä työssä on käsitelty tilannetta, jossa hyötyfunktiot tunnetaan täysin. Täydellisten hyötyfunktioiden selvittäminen asiakaskunnasta ei kuitenkaan todellisuudessa ole mahdollista, vaan tehtävä olisi pystyttävä ratkaisemaan myös pienemmillä tiedoilla asiakasluokkien mieltymyksistä.

Viitteet

[1] Rochet, J.-C. ja Stole, L. A. *The economics of multidimensional screening* Advances in Economics and Econometrica, vol. 1, 150-197. Cambridge University press, 2003

[2] Spence, M. *Multi-product quantity-dependant prices and profitability contraints* The Review of Economic Studies, 47:821-841, 1980

[3] Berg, K. ja Ehtamo, H. *Solving optimal screening problem with unknown utility functions* Käsikirjoitus, Saatavissa: http://www.sal.hut.fi/Publications/pdf-files/mber07.pdf

[4] Ehtamo, H., Berg, K. ja Kitti, M. *An adjustment scheme for nonlinear pricing problem with two buyers* European Journal of Operations Research, hyväksytty julkaistavaksi 2008

[5] Berg, K. ja Ehtamo, H. *Multidimensional Screening: Online Computation and Limited Information* Konferenssijulkaisu: 10th International Conference on Electronic Commerce (ICEC) 2008, Innsbruck

[6] Mussa, M. ja Rosen, S. *Monopoly and product quality* Journal of Economic Theory, 18:301-317, 1978

[7] Rochet, J.-C. ja Chone P. *Ironing, sweeping and multidimensional screening* Econometrica, 66:783-826, 1998

[8] Bertsekas, D. P. *Nonlinear Programming* Athena Scientific, Belmont, MA, second edition 2004.

[9] Nahata, B., Kokovin, S. ja Zhelobodko E. *Self-selection under non-ordered valuations: type-slpitting, envy-cycles, rationing and efficiency* Käsikirjoitus, 2001

[10] Nahata, B., Kokovin, S. ja Zhelobodko E. *Efficiency, over and under provision in package pricing: How to diagnose?* Käsikirjoitus, Maaliskuu 2006

[11] McAfee, R. P. ja MacMillan, J. *Multidimensional incentive compatibility and mechanism design* Journal of Economic Theory, 46:335-354, 1988

[12] Bazaraa, M. S., Sherali, H.D ja Shetty, C. M. *Nonlinear programming, Theory and algorithms* John Wiley and Sons, 3rd ed. 2006

[13] The MathWorks Inc., *MATLAB Online documentation R2008a* Saatavissa: http://www.mathworks.com/access/helpdesk/help/helpdesk.html, 6-8.2008

[14] Wilson, R. B. *Nonlinear Pricing* Oxford University Press, 1993