

Tatu Anttila

D-OPTIMAALISIIN KOEASETELMIIN LIITTYVISTÄ ALGORITMEISTA

Informaatioteknologian ja viestinnän tiedekunta
Kandidaattitutkielma
Huhtikuu 2023

TIIVISTELMÄ

Tatu Anttila: D-optimaalisiin koeasetelmiin liittyvistä algoritmeista
Kandidaattitutkielma
Tampereen yliopisto
Tietojenkäsittelytieteiden tutkinto-ohjelma
Huhtikuu 2023

Tämän tutkielman aihepiiri liittyy koesuunnitteluun. Koesuunnittelussa sovelletaan tilastollisia menetelmiä kokeiden suunnitteluun ja koetulosten analysointiin. Erityisesti regressioanalyysi on keskeinen tilastollinen menetelmä koesuunnittelun kannalta. Regressioanalyysissä muodostetaan niin kutsuttu mallimatriisi, joka puolestaan muodostetaan koejärjestelyä kuvaavan koeasetelman ja valitun regressiomallin perusteella. Muodostetun mallimatriisin perusteella sovitetaan joukko sovituskertoimia kokeiden tuloksiin. Sovituskertoimien estimaattien arvojen perusteella voidaan arvioida eri koeparametrien vaikutus havaittuihin koetuloksiin. Lisäksi sovituskertoimien estimaatteihin liittyvää epävarmuutta kuvaa niin kutsuttu yhteisluottamusväli. Yhteisluottamusvälin voidaan osoittaa riippuvan niin kutsutun hajontamatriisin determinantista. Edelleen, hajontamatriisi voidaan puolestaan muodostaa mallimatriisin perusteella.

Työssä esitellään koesuunnitteluun liittyviä keskeisiä käsitteitä ja tilastollisia menetelmiä. Näiden perusteella esitetään D-optimaalisuuden määritelmä: D-optimaalisen koeasetelman liittyvän hajontamatriisin determinantti on pienin mahdollinen. Tutkielmassa havainnollistetaan sitä, että D-optimaalisen koeasetelman etsiminen saattaa osoittautua laskennallisesti haastavaksi niissä tapauksissa, joissa mallimatriisin koko on riittävän suuri. Johtopäätöksenä on, että laskennan nopeuttamiseksi tarvitaan sopivia algoritmeja.

Tutkielmassa esitellään joukko D-optimaalisiin koeasetelmiin liittyviä algoritmeja ja vertaillaan niiden suorituskykyä saatavilla olevan kirjallisuuden perusteella. Näissä algoritmeissa optimoitavaa mallimatriisia päivitetään rivi kerrallaan hyödyntäen niin kutsuttuja rivinvaihtotekniikoita. Kirjallisuudessa esiintyvien tarkastelujen perusteella todetaan, että tarkastelluista algoritmeista k - ja kl -vaihtoalgoritmit tarjoavat parhaan kompromissin tarkkuuden ja laskennallisen tehokkuuden välillä. Tämä perustuu osittain siihen, että ne ovat yleistyksiä eräästä toisesta tutkielmassa esitellystä algoritmista. Lisäksi mainitut algoritmit sisältävät ns. vapaita parametreja (k ja l), joiden arvoja voidaan säätää tapauskohtaisesti parhaan tuloksen saamiseksi. Kattavaa vertailua eri rivinvaihtoalgoritmien suhteellisesta suorituskyvystä ei kuitenkaan löytynyt kirjallisuudesta. Siten tarvitaan lisätutkimusta tarkasteltujen algoritmien keskinäisen paremmuuden todentamiseksi.

Avainsanat: koesuunnittelu, optimiasetelma, D-optimaalisuus, laskennalliset algoritmit, regressioanalyysi

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

SISÄLLYSLUETTELO

1	Johdanto	1
2	Tutkimusmenetelmä.....	2
3	Koesuunnittelun peruspiirteistä.....	2
3.1	Koeasetelma, tekijät ja vasteet	2
3.2	Tekijöihin liittyvät vaikutukset	5
4	Koesuunnittelun menetelmistä	7
4.1	Lineaariset regressiomallit	7
4.1.1	Esimerkki: koeasetelmaa vastaavan mallimatriisin muodostaminen	8
4.1.2	Esimerkki: tekijän päävaikutuksen arviointi regressioanalyysin perusteella	9
4.2	Luottamusvälien arviointi	10
4.3	Yhteisluottamusväli	10
5	Optimiasetelmista	11
5.1	Täysitekijäkokeen optimaalisuus	11
5.2	Mallimatriisien muodostaminen ehdokasmatriisin perusteella	11
5.3	D-optimaalisuus	12
6	Rivinvaihtotekniikat.....	13
7	D-optimaalisiin koeasetelmiin liittyviä algoritmeja	14
7.1	DETMAX-algoritmi	14
7.2	Fedorovin algoritmi	16
7.3	Muunnettu Fedorovin algoritmi	17
7.4	k-vaihtoalgoritmi	18
7.5	kl-vaihtoalgoritmi	18
8	Keskustelu	19
9	Yhteenveto.....	19
	Lähdeluettelo.....	20

1 Johdanto

Koesuunnittelussa hyödynnetään tilastollisia menetelmiä koejärjestelyjen suunnittelussa ja koetulosten analysoinnissa. Koesuunnittelun menetelmiä sovelletaan monissa tieteenaloissa, kuten lääketieteessä, biokemiassa ja teknillisissä tieteissä (Ilzarbe et al., 2008; Durakovic, 2017). Koesuunnittelun avulla pyritään saamaan tarkempia ja luotettavampia tuloksia vähemmällä työllä ja kustannuksella. Erityisesti tuotekehityksen yhteydessä koesuunnittelua hyödyntämällä voidaan parantaa tuotteiden laatua, tehostaa tuotantoprosessia ja vähentää kustannuksia.

Koesuunnittelun lähtökohtana on suorittaa sarja kokeita, joissa valittujen koeparametrien arvoja vaihdellaan järjestelmällisellä tavalla yksittäisten kokeiden välillä. Tällainen koejärjestely muodostetaan niin kutsutun *koasetelman* perusteella. Koasetelma sisältää koeparametrien saamat arvot eri kokeissa, ja se voidaan esittää myös matriisimuodossa. Muodostetun koasetelman perusteella voidaan edelleen muodostaa koetuloksia vastaava tilastollinen malli. Tämä mahdollistaa koetulosten tilastollisen analyysin sekä koeparametreihin liittyvien epävarmuuksien ja koeparametrien tilastollisen merkitsevyyden arvioinnin.

Niin kutsuttu *optimiasetelma* on sellainen koasetelma, joka on tietyn kvantitatiivisen kriteerin perusteella paras mahdollinen (Atkinson & Donev, 1992). Tässä työssä tarkastellaan *D-optimaalisia* koasetelmia. Muitakin optimaalisuuden kriteerejä on esitetty kirjallisuudessa (Atkinson & Donev, 1992). Todetaan kuitenkin, että *D-optimaalisuuden* kriteeri on yksi suosituimmista optimaalisuuden kriteereistä koesuunnittelun käytännön sovelluksissa (de Aguiar et al., 1995; Montgomery, 2013, luku 9.6.1).

D-optimaalinen koasetelma voidaan määrittää koesuunnittelun yhteydessä ennen kuin yhtään koetta on suoritettu. *D-optimaalisen* koasetelman etsiminen käymällä läpi kaikki mahdolliset eri koasetelmia vastaavat matriisit voi kuitenkin osoittautua laskennallisesti haastavaksi. Siten *D-optimaalisen* koasetelman muodostamiseksi hyödynnetäänkin sopivia algoritmeja.

Tässä työssä luodaan katsaus joukkoon kirjallisuudessa esiintyviin ns. rivinvaihtotekniikoita hyödyntäviin algoritmeihin, joita käytetään *D-optimaalisen* koasetelmien löytämiseksi. Lisäksi esiteltyjen algoritmien suhteellista suorituskkyä vertaillaan kirjallisuuden perusteella. Työn tutkimuskysymys voidaan siten esittää seuraavasti: ”mikä tarkastelluista, *D-optimaalista* koasetelmaa etsivistä rivinvaihtoalgoritmeista on suorituskkyvyltään paras?”.

Tutkielma on luonteeltaan kirjallisuuskatsaus. Luvussa 2 kuvaillaan työn tutkimusmenetelmää, eli käytetyt tiedonhaun menetelmät. Luvuissa 3 ja 4 esitellään koesuunnittelun keskeisiä piirteitä (luku 3) ja menetelmiä (luku 4). Luvussa 5 käsitellään

optimiasetelmia, erityisesti D-optimaalisia koeasetelmia. Luvussa 6 esitellään puolestaan rivinvaihtotekniikat, joihin luvussa 7 esitellyt algoritmit perustuvat. Lisäksi luku 8 sisältää tutkielman tuloksiin ja sen aihepiiriin liittyvän keskustelun. Lopuksi luvussa 9 esitetään työn yhteenveto.

2 Tutkimusmenetelmä

Tämä tutkielma on luonteeltaan kirjallisuuskatsaus. Sen aihepiiriin liittyviä lähteitä etsittiin Google Scholar-palvelun ja Tampereen yliopiston Andor-kirjastopalvelun tietokannoista syöttämällä hakukenttään muun muassa seuraavia merkkijonoja:

- "Design of experiments"
- "Optimal designs"
- "Optimal design of experiments"
- "D-optimal designs"

Lisäksi ylläolevat haut toistettiin lisäämällä hakukenttään: AND algorithms, ja AND review. Hakutulokset rajattiin kansainvälisissä julkaisusarjoissa julkaistuihin artikkeleihin sekä toisaalta oppikirjoihin/monografioihin. Sopivat lähteet noudettiin Andor-palvelusta tai Google Scholar-hakupalvelun kautta lähempää tarkastelua varten. Edelleen, noudettujen julkaisujen ja kirjojen sisältämät viitteet käytiin kursorisesti läpi. Siten löydettiin uusia, mahdollisesti kiinnostavia lähteitä jotka noudettiin (mikäli ne olivat saatavilla) lähempää tarkastelua varten. Tätä hakuprosessia toistettiin useampaan kertaan.

3 Koesuunnittelun peruspiirteistä

Luvuissa 3.1 ja 3.2 esitellään keskeisiä koesuunnitteluun liittyviä käsitteitä ja piirteitä sekä havainnollistetaan niitä esimerkin avulla.

3.1 Koeasetelma, tekijät ja vasteet

Koesuunnittelulla pyritään muodostamaan joukko kokeita niin, että mahdollisimman pienellä määrällä kokeita saadaan mahdollisimman suuri määrä tietoa tutkitusta koeprosessista (Montgomery, 2013). Tämä tapahtuu laatimalla koeasetelma, joka koostuu joukosta suoritettavia kokeita. Näissä kokeissa koeparametrien eli *tekijöiden* arvoja vaihdellaan systemaattisella tavalla. Yhdistämällä laadittu koesuunnitelma sen pohjalta laadittujen kokeiden mittaustuloksiin eli *vasteisiin* voidaan arvioida esimerkiksi eri tekijöiden vaikutusta mitattuun suureeseen.

Tässä luvussa havainnollistetaan koesuunnittelun keskeisiä piirteitä esimerkin avulla. Kuva 1 (ks. alla) havainnollistaa sitä, että yksittäiseen kokeeseen liittyy vain yksittäinen vaste, vaikka koeprosessiin liittyvä koejärjestely voi tuottaa havaintoja useammasta suureesta.



Kuva 1. Yksinkertaistettu kuvaus koesuunnitteluun perustuvasta koejärjestelystä.

Koesuunnitelmassa määritellään kokeisiin liittyvät tekijät, jotka ovat tässä käsitellyn esimerkin tapauksessa seuraavat: lämpötila (tekijä *A*), sekoitusnopeus (tekijä *B*) ja aika (tekijä *C*). Kaikki muut vasteeseen vaikuttavien tekijöiden arvot oletetaan pysyvän vakioina. Lisäksi seuraavassa esimerkissä ja yleisemmin tässä työssä tarkastellaan vain sellaisia koeasetelmia, joissa kukin tekijä voi saada korkeintaan kaksi eri arvoa. Tekijöiden tasoja nimitetään vastaavasti matalaksi ja korkeaksi tasoksi.

Esimerkin mukaisen koeasetelman sisältämät tekijät ja niiden tasojen suuruudet on esitelty taulukossa 1.

Taulukko 1. Koeasetelmaan liittyvät tekijät ja niiden tasojen arvot.

Parametri	Matala	Korkea
Lämpötila	100 C	200 C
Sekoitusnopeus	20 rpm	30 rpm
Aika	10 min	15 min

Taulukon 1 pohjalta voidaan laatia niin kutsuttu *täysitekijäkoe*. Kahden tason täysitekijäkokeen mukainen asetelma sisältää kaikki mahdolliset tekijöiden tasojen yhdistelmät. Voidaan helposti päätellä, että kahden tason täysitekijäkoe sisältää 2^p koetta missä p on tekijöiden lukumäärä. Esimerkin mukaisessa tapauksessa tämä asetelma sisältää siis kahdeksan koetta ($2^3 = 8$). Esimerkin mukainen täysitekijäkoe on esitetty taulukossa 2.

Taulukko 2. Esimerkin mukaisen koeasetelmaan liittyvät tekijät ja niiden arvot.

Koe	Lämpötila	Sekoitusnopeus	Aika
1	100 C	20 rpm	10 min
2	200 C	20 rpm	10 min
3	100 C	30 rpm	10 min
4	200 C	30 rpm	10 min
5	100 C	20 rpm	15 min
6	200 C	20 rpm	15 min
7	100 C	30 rpm	15 min
8	200 C	30 rpm	15 min

Taulukon 2 mukainen koesuunnitelma voidaan esittää matriisimuodossa merkitsemällä tekijän matalaa tasoa luvulla -1 ja korkeaa tasoa luvulla 1. Tällaista matriisia kutsutaan *asetelmamatriisiksi* A , ja se on tässä käsitellyn esimerkin tapauksessa yhtälön 1 mukainen.

$$A = \begin{bmatrix} -1 & -1 & -1 \\ 1 & -1 & -1 \\ -1 & 1 & -1 \\ 1 & 1 & -1 \\ -1 & -1 & 1 \\ 1 & -1 & 1 \\ -1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}. \quad (1)$$

Kuten huomataan, taulukon 2 rivit vastaavat asetelmamatriisin A rivejä siten, että esimerkiksi taulukon 1 ensimmäisellä rivillä määritelty koe vastaa matriisin A ensimmäistä riviä.

Käsitellyssä esimerkissä jokaiseen koeasetelman mukaiseen kokeeseen liittyy yksittäinen havainto eli vaste, joka on tässä tapauksessa liuoksen tiheys. Esimerkin mukaiset vasteet on esitetty taulukossa 3.

Taulukko 3. Taulukon 2 mukaisen koeasetelman tuottamat vasteet.

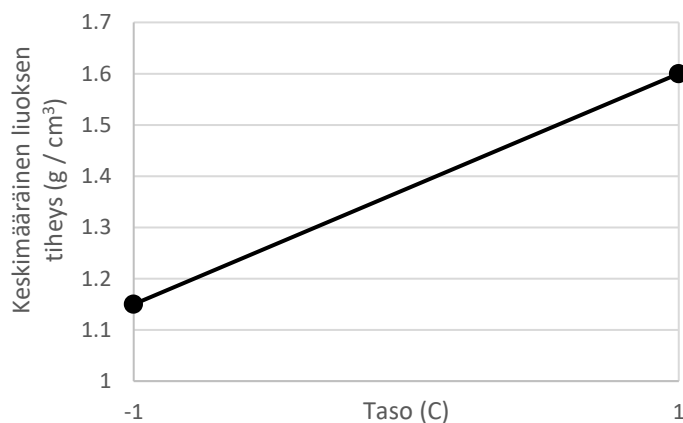
Koe	Liuoksen tiheys
1	1 g/cm ³
2	1.1 g/cm ³
3	1.2 g/cm ³
4	1.3 g/cm ³
5	1.7 g/cm ³
6	1.8 g/cm ³
7	1.6 g/cm ³
8	1.3 g/cm ³

Esimerkiksi taulukon 3 ensimmäisellä rivillä ilmoitettu vaste vastaa taulukon 2 ensimmäisen rivin koetulosta. Taulukon 3 mukaiset vasteet Y voidaan myös esittää vektorimuodossa seuraavasti:

$$Y = [1, 1.1, 1.2, 1.3, 1.7, 1.8, 1.6, 1.3]^T. \quad (2)$$

3.2 Tekijöihin liittyvät vaikutukset

Koeasetelman ja siihen liittyvän koeprosessin tuottamien vasteiden perusteella voidaan arvioida sitä, mikä on yksittäisten tekijän (tai niiden yhdistelmän) vaikutus vasteen havaittuun vaihteluun. Yksittäiseen tekijään liittyvää vaikutusta kutsutaan *päävaikutukseksi*. Kuvassa 2 on havainnollistettu tekijän C (lämpötila) päävaikutusta. Kuvan 2 perusteella voidaan havaita, että vaste (liuoksen tiheys) on keskimäärin suurempi niissä kokeissa, joissa tekijän C taso on korkea verrattuna niihin kokeisiin, joissa tekijän C taso on matala.

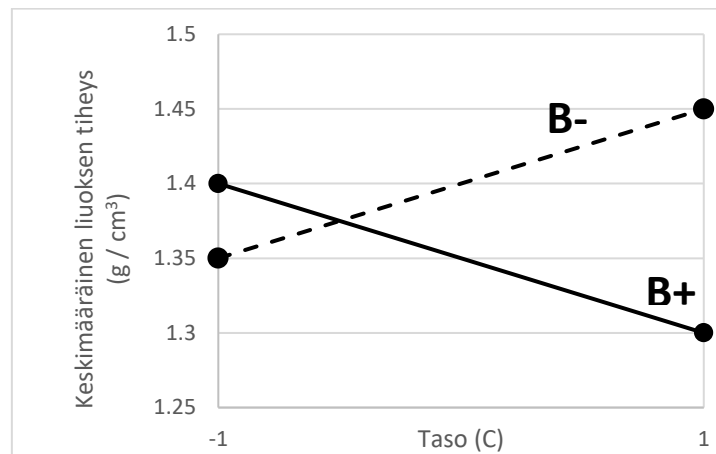


Kuva 2. Keskimääräinen liuoksen tiheys ajan tason funktiona.

Tekijään C liittyvä päävaikutus eff_C voidaan laskea niin kutsuttujen *kontrastien* avulla (ks. Montgomery, 2013, luku 6.2) seuraavasti:

$$eff_C = \frac{\Sigma(\text{vasteet joissa } C: n \text{ taso on korkea})}{\text{vastaavien kokeiden lukumäärä}} - \frac{\Sigma(\text{vasteetjoissa } C: n \text{ taso on matala})}{\text{vastaavien kokeiden lukumäärä}} \quad (3)$$
$$= (1.7 + 1.8 + 1.6 + 1.3)/4 - (1.0 + 1.1 + 1.2 + 1.3)/4 = 0.1125.$$

Edellä olevassa laskelmassa esiintyvät luvut on saatu taulukosta 3.



Kuva 3. Keskimääräinen liuoksen tiheys (vaste) tekijän ajan tason funktiona kahdessa eri tapauksessa: niissä kokeissa, joissa sekoittumisnopeuden taso oli korkea (B+) ja niissä kokeissa, joissa sekoittumisnopeuden taso oli matala (B-).

Päävaikutuksien lisäksi voidaan arvioida myös niin kutsuttuja *yhdysvaikutuksia*. Kahden eri tekijän (tässä tapauksessa aika, C , ja sekoittumisnopeus, B) yhdysvaikutusta on havainnollistettu kuvassa 3. Voidaan havaita, että mitattu vaste riippuu tekijän B tasosta niin, että kokeissa joissa tekijä B taso oli korkea, vasteen suuruus laskee tekijän C tason mukana. Toisaalta kokeissa joissa tekijän B taso oli matala, tulos on kuvan 2 mukainen: liuoksen tiheys kasvoi tekijän C tason mukana.

Tekijöiden B ja C vuorovaikutukseen liittyvän yhdysvaikutuksen suuruus eff_{BC} voidaan laskea seuraavasti:

$$eff_{BC} = eff_C(B+) - eff_C(B-)$$
$$= [(1.3 + 1.3)/2 - (1.2 + 1.6)/2] - [(1.1 + 1.8)/2 - (1.0 + 1.7)/2] = -0.1.$$

Edellä olevassa yhtälössä $eff_C(B+)$ ja $eff_C(B-)$ ovat tekijän C päävaikutukset niissä kokeissa, joissa tekijän B taso on korkea ja matala, vastaavasti.

4 Koesuunnittelun menetelmistä

Seuraavassa käydään läpi koesuunnittelun kannalta keskeisiä tilastollisia menetelmiä. Luvussa 4.1 käsitellään regressioanalyysiä. Alaluvuissa 4.1.1 ja 4.1.2 havainnollistetaan regressioanalyysin soveltamista edellisen luvun esimerkkiin. Luvuissa 4.2 ja 4.3 käsitellään vaikutuksiin liittyvien estimaattien luottamusvälejä (luku 4.2) sekä estimaattien yhteisluottamusväliä (luku 4.3).

4.1 Lineaariset regressiomallit

Tässä luvussa esitellään regressioanalyysin pääpiirteet erityisesti koesuunnittelun kannalta. Seuraavassa tarkastellaan siten kaksitasoista koesuunnitelmaa, jossa on p tekijää. Olkoon k regressiomallin sisältämien sovituskertoimien (ks. alla) lukumäärä. Olkoon lisäksi N vasteiden y_i lukumäärä, missä $N > k$.

Tarkastellaan ensin mallia, joka huomioi vain koesuunnitelman mukaiset päävaikutukset. Vasteita y_i voidaan tällöin mallintaa seuraavan yhtälön perusteella (Montgomery, 2013, luku 10.2):

$$y_i = \beta_0 + \sum_{j=1}^{j=p-1} \beta_j x_{i,j} + \varepsilon_i, i = 1 \dots N. \quad (4)$$

Tässä muuttujat $\beta_0, \dots, \beta_{p-1}$ ovat niin kutsuttuja *sovituskertoimia* ja muuttujan $x_{i,j}$ arvo on tekijän j taso kokeessa i (1 tai -1). Lisäksi termi ε_i on niin kutsuttu *jäännöstermi*, joka kuvaa regressiomallin ennusteen ja vasteen eroa. Regressioanalyysissä oletetaan, että jäännöstermit ovat normaalijakaumaa noudattavia satunnaismuuttujia, joiden keskiarvo on nolla ja varianssi σ^2 .

Vastaavasti voidaan muodostaa malleja, jotka huomioivat lisäksi muita vaikutuksia kuin tekijöiden päävaikutukset. Yhtälön 4 mukaista mallia voidaan esimerkiksi laajentaa huomioimaan kahden tekijän väliset yhdysvaikutukset:

$$y_i = \beta_0 + \sum_{j=1}^{j=p-1} \beta_j x_j + \sum_{j=1}^{j=p-1} \sum_{l=j+1}^{l=p-1} \beta_{jl} x_{i,j} x_{i,l} + \varepsilon_i, i = 1 \dots N. \quad (5)$$

Voidaan myös muodostaa malleja, jotka sisältävät kolmen tai useamman tekijän väliset yhdysvaikutukset tai niin kutsuttuja *kvadraattisia malleja*, joissa on mukana termejä, jotka vastaavat eri tekijöiden neliöitä (Montgomery, 2013, luku 10.2).

Lineaarinen regressiomalli voidaan esittää yleisesti matriisimuodossa (Kutner et al., 2005):

$$Y = X\beta + \varepsilon, \text{ missä} \quad (6)$$

$$\mathbf{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix}, \mathbf{X} = \begin{pmatrix} 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,k} \\ 1 & x_{2,1} & x_{2,2} & \cdots & x_{2,k} \\ 1 & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N,1} & x_{N,2} & \cdots & x_{N,k} \end{pmatrix}, \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_k \end{pmatrix}, \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_N \end{pmatrix}.$$

Vektori \mathbf{Y} koostuu vasteista (pituus on N). Lisäksi matriisiin \mathbf{X} koko on $N \times (k+1)$, ja $\boldsymbol{\beta}$ on sovituskertoimista koostuva vektori, jonka pituus on $k+1$. Lopuksi, vektori $\boldsymbol{\varepsilon}$ sisältää jäännöstermit (pituus on N).

Tässä työssä regressiomalliin liittyvää matriisia \mathbf{X} kutsutaan *mallimatriisiksi*, vaikka tilastotieteen sanaston (Alho et al., 2021) perusteella tämä ei ole vakiintunut käänнос englanninkielisestä termistä *model matrix*. Valittu käänнос perustuu siihen, että näin voidaan viitata toisaalta asetelmamatriisiin \mathbf{A} (ks. yhtälö 1) ja toisaalta matriisiin \mathbf{X} käyttäen niistä eri termejä.

Niin kutsutun *suurimman uskottavuuden menetelmän* perusteella saadaan sovituskertoimien $\boldsymbol{\beta}$ estimaatit $\hat{\boldsymbol{\beta}}$ laskettua seuraavasti (Kutner et al., 2005):

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}. \quad (7)$$

Regressioanalyysiin teorian mukaisesti sovituskertoimien estimaatit $\hat{\boldsymbol{\beta}}$ ovat satunnaismuuttujia. Teorian mukaisten oletusten perusteella voidaan johtaa seuraava tulos koskien estimaattien $\hat{\boldsymbol{\beta}}$ kovarianssimatriisia, $Cov(\hat{\boldsymbol{\beta}})$ (Kutner et al., 2005, luku 6):

$$Cov(\hat{\boldsymbol{\beta}}) = \sigma^2 \mathbf{H}, \text{ missä} \quad (8a)$$

$$\mathbf{H} = (\mathbf{X}^T \mathbf{X})^{-1}. \quad (8b)$$

Yhtälöistä 8a ja 8b havaitaan, että sovituskertoimien kovarianssimatriisi riippuu sekä jäännöstermien ε_i varianssista, σ^2 , että niin kutsutusta *hajontamatriisista* \mathbf{H} (yhtälö 8b). Todetaan, että estimaatin $\hat{\beta}_i$ varianssi, $Var(\hat{\beta}_i)$, on verrannollinen hajontamatriisin \mathbf{H} i :nteen diagonaalialkioon. Lisäksi voidaan todeta, että estimaattien $\hat{\beta}_i$ ja $\hat{\beta}_j$ kovarianssi $Cov(\hat{\beta}_i, \hat{\beta}_j)$ on verrannollinen hajontamatriisin \mathbf{H} alkioon (i, j) .

4.1.1 Esimerkki: koeasetelmaa vastaavan mallimatriisin muodostaminen

Havainnollistetaan edellä esitettyä teoriaa soveltamalla sitä edellisessä luvussa käsiteltyyn esimerkkiin. Muodostetaan ensin regressiomalli, joka huomioi vain tekijöiden A , B ja C päävaikutukset. Vastaava mallimatriisi \mathbf{X} on siten yhtälöiden 1, 4 ja 6 perusteella seuraava:

$$\mathbf{X} = \begin{bmatrix} 1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}. \quad (9)$$

Huomataan, että esimerkin mukaisen asetelmamatriisin A (ks. yhtälö 1) ja yhtälön 9 kuvaaman mallimatriisin X erona on se, että yhtälöön 10 on lisätty ensimmäiseksi sarakkeeksi luvuista yksi koostuva sarake.

Yhtälön 9 kuvaama regressiomalli huomioi vain tekijöiden A , B ja C päävaikutukset. Mikäli halutaan arvioida kahden eri tekijöiden väliset yhdysvaikutukset (tässä tapauksessa siis tekijöiden A ja B , tekijöiden A ja C , sekä tekijöiden B ja C yhdysvaikutukset), saadaan mallimatriisiksi (vrt. yhtälö 5):

$$\mathbf{X} = \begin{bmatrix} 1 & -1 & -1 & -1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 \\ 1 & 1 & 1 & -1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \quad (10)$$

Vertailemalla yhtälöitä 9 ja 10 huomataan, että yhtälö 10 sisältää kolme lisättyä saraketta siten, että yhtälön 10 viides sarake sisältää tekijöiden A ja B tasojen tulot kussakin kokeessa. Vastaavasti yhtälön 10 kuudes sarake sisältää tekijöiden A ja C tasojen tulot, ja yhtälön 10 seitsemäs sarake sisältää tekijöiden B ja C tasojen tulot.

4.1.2 Esimerkki: tekijän päävaikutuksen arviointi regressioanalyysin perusteella

Havainnollistetaan päävaikutusten laskemista edellä käsitellyn esimerkin tapauksessa. Sijoittamalla vasteet Y (yhtälö 2) ja yhtälön 9 mukainen mallimatriisi X yhtälöön 7, saadaan laskettua sovituskertoimien β_i estimaatit $\hat{\beta}_i$. Tässä tapauksessa kerroin $\hat{\beta}_3$ vastaa tekijän C päävaikutuksen suuruutta, ja sille saadaan arvoksi 0.225. Kyseistä lukua voidaan verrata luvussa 3 arvioituun tekijään C vaikutuksen suuruuteen, eff_C , joka on 0.1125 (ks. yhtälö 3). Havaitaan, että $\hat{\beta}_3 = 2 \times 0.1125 = 2 \times eff_C$.

Tämä esimerkki havainnollistaa yleisempää tulosta, jonka mukaan kaksitasoisen täysitekiäkokeen tapauksessa regressioanalyysi ja luvussa 3 havainnollistettu kontrasteihin perustuva laskennallinen menetelmä antavat keskenään yhtenevät tulokset (Montgomery, 2013, luvut 6.7 ja 10). Tarkemmin sanoen, näiden kahden menetelmän tuottamat arviot vaikutukselle V poikkeavat toisistaan tekijällä kaksi, ts. $\hat{\beta}_V = 2 \times eff_V$.

4.2 Luottamusvälien arviointi

Luvussa 4.1 todettiin, että sovituskerroimien estimaatit $\hat{\beta}$ ovat satunnaismuuttujia. Täten estimaatille $\hat{\beta}_i$ voidaan arvioida niin kutsuttu *luottamusväli* regressioanalyysin teorian perusteella (Montgomery, 2013, luku 10.5.1). Luottamusväli määrittää rajat, joiden sisällä sovituskerroimen β_i arvo on todennäköisyydellä $(1-\alpha)$, missä α on niin kutsuttu *merkitsevyystaso*. Luottamusvälin alarajalle, $CI_-(\hat{\beta}_i)$, ja ylärajalle, $CI_+(\hat{\beta}_i)$, voidaan johtaa seuraavat lausekkeet (Montgomery, 2013, luku 10.5.1):

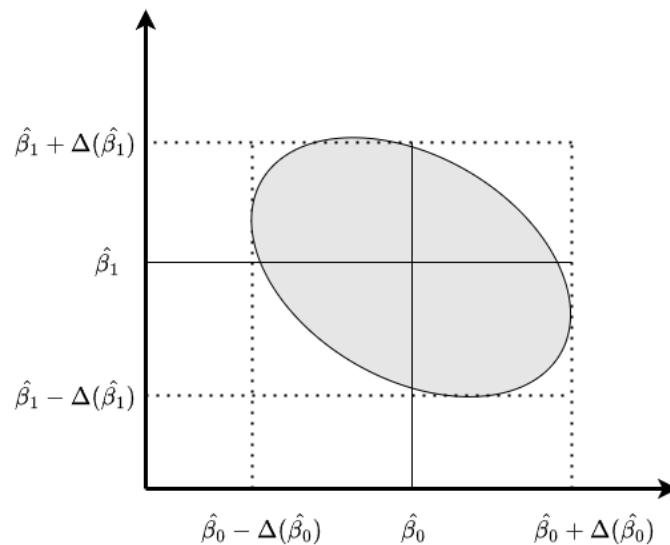
$$CI_-(\hat{\beta}_i) = \hat{\beta}_i - t_{\alpha/2, N-p} \times se(\hat{\beta}_i) = \hat{\beta}_i - \Delta(\hat{\beta}_i), \quad (11a)$$

$$CI_+(\hat{\beta}_i) = \hat{\beta}_i + t_{\alpha/2, N-p} \times se(\hat{\beta}_i) = \hat{\beta}_i + \Delta(\hat{\beta}_i). \quad (11b)$$

Yhtälöiden 11a ja 11b oikealla puolella esiintyvä termi $se(\hat{\beta}_i)$ on sovituskerroimen i estimaatin $\hat{\beta}_i$ keskihajonta. Lisäksi termi $t_{\alpha/2, N-p}$ on vapausastetta $N-k-1$ noudattavan Studentin t-jakauman todennäköisyyttä $\alpha/2$ vastaava kvantiili, jonka arvo saadaan laskettua t-jakauman kertymäfunktion käänteisfunktion perusteella.

4.3 Yhteisluottamusväli

Sovituskerroimien β_i yhteisluottamusvälin voidaan tulkita kuvaavan kertoimiin liittyvää kokonaisvirhettä, joka johtuu valitusta koeasetelmasta ja sen pohjalta suoritettujen kokeiden tuloksiin liittyvistä virheistä (de Aguiar et al., 1995, luku 3.1). Sovituskerroimien yhteisluottamusväliä voidaan havainnollistaa geometrisesti tapauksessa, jossa regressiomalli sisältää kaksi sovituskerrointa β_0 ja β_1 (kuva 4). Kuvan 4 X ja Y-akseleilla on esitetty sovituskerroimien estimaatit ja niiden luottamusvälit. Kuvassa 4 oleva ellipsoidi kuvaa puolestaan sovituskerroimien yhteisluottamusväliä.



Kuva 4. Havainnollistava kuva kahden sovituskerroimen yhteisluottamusvälistä (harmaalla värillä oleva ellipsoidi). Pisteviivat kuvaavat sovituskerroimien

luottamusvälejä. Lisäksi yhtenäisten viivojen risteyskohta vastaa sovituskerroimien estimaattia $\hat{\beta} = [\hat{\beta}_0, \hat{\beta}_1]^T$.

Kuvan 4 perusteella voidaan heuristisesti päätellä, että mitä pienempi ellipsoidin pinta-ala on, sitä tarkempia estimaatit $\hat{\beta}$ ovat. Tämä päätelmä pätee yleisemmin useamman sovituskerroimen tapauksessa, jossa kertoimien yhteisluottamusväli muodostaa hyperellipsoidin. Tällöin hyperellipsoidin tilavuus antaa mitan estimaattien $\hat{\beta}$ tarkkuudelle (Atkinson & Donev, 1992, luku 6; de Aguiar et al., 1995, luku 3). Voidaan edelleen osoittaa, että hyperellipsoidin tilavuus on verrannollinen hajontamatriisin H determinantin neliöjuureen (de Aguiar et al., 1995, luku 3).

5 Optimiasetelmista

Tässä luvussa tarkastellaan kaksitasoista koeasetelmaa, joka sisältää p tekijää. Lähtökohtana on, että koeasetelmaan liittyviä vasteita mallinnetaan valitulla regressiomallilla. Luvussa 5.1 käsitellään täysitekijäkokeen mukaisen koeasetelman optimaalisuutta. Luku 5.2 taustoittaa D-optimaalisuuden määrittelyä (luku 5.3).

5.1 Täysitekijäkokeen optimaalisuus

Tarkastellaan aluksi täysitekijäkoeetta, joka sisältää kaikki tekijöiden tasojen permutaatiot (ks. luku 3.1). Täten vastaava koesuunnitelma tarjoaa suurimman mahdollisimman määrän tietoa tutkitusta prosessista. Voidaan lisäksi todeta, että täysitekijäkokeen tapauksessa sovituskerroimet ovat toisistaan riippumattomia, ja että sovituskerroimien yhdysvaikutuksia kuvaavan ellipsoidin tilavuus on minimaalinen (Montgomery, 2013, luku 6.7). Tässä mielessä täysitekijäkoe on optimiasetelma.

Kuten luvussa 3.1 todettiin, täysitekijäkokeeseen sisältyvien kokeiden lukumäärä on 2^p . Suoritettavien kokeiden lukumäärä kasvaa siis eksponentiaalisesti tekijöiden lukumäärän mukana. Johtuen esimerkiksi ajallisista tai materiaalisista rajoitteista, ei kaikkia täysikoeasetelman mukaisia kokeita voida tai haluta suorittaa, vaan pelkästään osa niistä. Tämä johtaa optimointiongelmaan, jossa pyritään muodostamaan suppeammasta koejoukosta koostuva optimiasetelma.

5.2 Mallimatriisien muodostaminen ehdokasmatriisin perusteella

Edellisessä luvussa 5.1 esitettyjen pohdintojen nojalla tarkastellaan tilannetta, jossa suoritettavien kokeiden lukumääräksi on valittu n , missä n on pienempi kuin kaikkien mahdollisten koesuunnitelman mukaisten kokeiden lukumäärä, M , eli $n < M$. Edellyttäen, että kokeiden muodostamiselle ei ole rajoituksia (jotka voivat johtua esimerkiksi mittausteknisistä tai fysikaalisista syistä), pätee $M = 2^p$.

Voidaan siten muodostaa ehdokasmatriisi ζ_M koesuunnitelman perusteella. Tarkemmin sanoen, matriisin ζ_M rivien lukumäärä on M , ja sen yksittäinen rivi koostuu tekijöiden tasoista kyseisessä kokeessa. Olkoon lisäksi X_M mallimatriisi, joka on

muodostettu valitun regressiomallin ja matriisin ζ_M perusteella. Täten voidaan muodostaa n riviä sisältäviä mallimatriiseja X_n siten, että ne koostuvat matriisin X_M eri riveistä.

Edellisissä luvuissa käsitellyn esimerkin tapauksessa ehdokasmatriisi on yhtälön 1 mukainen. Oletetaan, että valittu regressiomalli on yhtälön 4 mukainen ja että $n=4$. Tällöin matriisi X_M on yhtälön 9 mukainen. Matriisin X_M perusteella voidaan edelleen muodostaa esimerkiksi seuraavat neljä riviä sisältävät mallimatriisit:

$$X_{1234} = \begin{bmatrix} 1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & -1 \end{bmatrix}, X_{2468} = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}. \quad (12)$$

Yhtälössä 12 esiintyvä mallimatriisi X_{1234} on muodostettu poimimalla matriisin X_M neljä ensimmäistä riviä ja mallimatriisi X_{2468} on muodostettu poimimalla matriisin X_M parilliset rivit.

5.3 D-optimaalisuus

Merkitään symbolilla $\Omega(X_M)$ kaikkien niiden mallimatriisien X_n joukkoa, jotka ovat muodostettu poimimalla n eri riviä matriisista X_M . D-optimaalinen mallimatriisi X^* on sellainen matriisi, jolle pätee (de Aguilar et al., 1995, luku 3.1):

$$|(X^*)^T X^*|^{-1} = \min_{X \in \Omega(X_M)} (|X^T X|^{-1}). \quad (13)$$

Vertaamalla yhtälöä 13 yhtälöön 8b havaitaan, että D-optimaalinen mallimatriisi minimoi hajontamatriisin H determinantin. Täten D-optimaalisuuden kriteeri voidaan tulkita niin, että D-optimaalista mallimatriisia vastaavien sovituskerroimien (ko)varianssien arvot ovat mahdollisimman pieniä. Siten myös tekijöiden vaikutusten arvioihin liittyvät epävarmuudet ovat mahdollisimman pieniä (ks. luku 4.1.2). Lisäksi luvun 4.3 perusteella voidaan todeta, että D-optimaalinen mallimatriisi minimoi sovituskerroimien yhteisluottamusväliä kuvaavan hyperellipsoidin tilavuuden.

Luvussa 5.2 käsitellyn esimerkin tapauksessa ($M=8$ ja $n=4$) voidaan kombinatoriikkaan perustavalla laskelmalla todeta, että joukon $\Omega(X_M)$ alkioden lukumäärä on 70. Laskemalla näitä mallimatriiseja vastaavien hajontamatriisien determinantit, löydetään yhtälön 13 perusteella D-optimaalinen mallimatriisi X^* . Edelleen, matriisin X^* ja valitun regressiomallin perusteella voidaan muodostaa vastaava D-optimaalinen asetelmamatriisi A^* .

Edellä kuvatun esimerkin tapauksessa D-optimaalisen asetelmamatriisin löytäminen ei ole laskennallisesti haastava tehtävä. Toisaalta voidaan tarkastella esimerkiksi koesuunnitelmaa, jossa $M=256$ ja $n=4$. Tällöin joukon $\Omega(X_M)$ alkioden lukumäärä on noin 17.5×10^6 . Toisin sanoen, D-optimaalisen asetelmamatriisin etsiminen siten, että

käydään läpi kaikki koesuunnitelman mukaiset mallimatriisit ja lasketaan näitä vastaavien hajontamatriisien determinantit, voi muodostua haasteeksi laskennallisessa mielessä.

6 Rivinvaihtotekniikat

Tässä luvussa esitellään luvussa 7 käsiteltyjen algoritmien kannalta keskeisiä matriisilaskentaan liittyviä tekniikoita. Näiden tekniikoiden taustalla on havainto siitä, että mallimatriisia vastaavan hajontamatriisin determinantin laskeminen (jonka minimintiin D-optimointi perustuu; ks. yhtälö 13) on laskennallisesti työlästä iteratiivisen algoritmin yhteydessä johtuen determinantin laskemiseen liittyvistä matriisioperaatioista.

Olkkoon \mathbf{X}_n joukkoon $\Omega(\mathbf{X}_M)$ kuuluva mallimatriisi. Seuraavassa tarkastellaan sitä, kuinka yksittäisen rivin lisääminen matriisiin \mathbf{X}_n tai siihen kuuluvan rivin poistaminen muuttaa tuloksena saadun matriisin hajontamatriisin determinanttia. Tähän liittyen, olkkoon χ_x matriisin \mathbf{X}_M sisältyvä rivi. Nyt funktio $d(\chi_x)$ määritellään seuraavasti (de Aguiar et al., 1995, luku 5):

$$d(\chi_x) = \chi_x^T |\mathbf{X}_n^T \mathbf{X}_n|^{-1} \chi_x. \quad (14)$$

Olkkoon χ_j matriisin \mathbf{X}_M kuuluva rivi, joka ei sisälly matriisiin \mathbf{X}_n . Lisäämällä rivi χ_j matriisiin \mathbf{X}_n saadaan matriisi \mathbf{X}_{n+1} . Nyt voidaan johtaa seuraava tulos (de Aguiar et al., 1995, luku 5):

$$|\mathbf{X}_{n+1}^T \mathbf{X}_{n+1}| = |\mathbf{X}_n^T \mathbf{X}_n| (1 + d(\chi_j)). \quad (15)$$

Vastaavalla tavalla voidaan tarkastella rivin χ_j poistamista mallimatriisista \mathbf{X}_n , jolloin saadaan matriisi \mathbf{X}_{n-1} . Tässä tapauksessa pätee (de Aguiar et al., 1995, luku 5):

$$|\mathbf{X}_{n-1}^T \mathbf{X}_{n-1}| = |\mathbf{X}_n^T \mathbf{X}_n| (1 - d(\chi_j)). \quad (16)$$

Voidaan lisäksi tarkastella operaatiota, jossa matriisiin \mathbf{X}_n kuuluva rivi χ_i korvataan sellaisella rivillä χ_j , joka ei sisälly matriisiin \mathbf{X}_n . Operaation tuloksena saadaan matriisi \mathbf{X}_u , jolle pätee (de Aguiar et al., 1995, luku 6):

$$|\mathbf{X}_u^T \mathbf{X}_u| = |\mathbf{X}_n^T \mathbf{X}_n| (1 + \Delta(\chi_i, \chi_j)), \text{ missä} \quad (17)$$

$$\Delta(\chi_i, \chi_j) = d(\chi_j) - \left[d(\chi_i) d(\chi_j) - d(\chi_i, \chi_j)^2 \right] - d(\chi_i), \text{ ja} \quad (18)$$

$$d(\chi_i, \chi_j) = \chi_i^T |\mathbf{X}_n^T \mathbf{X}_n|^{-1} \chi_j. \quad (19)$$

Yhtälön 18 vasemmalla puolella esiintyvä funktio $\Delta(\chi_i, \chi_j)$ on niin kutsuttu *delta-funktio*.

Rivinvaihtotekniikat mahdollistavat sen, että algoritmin suorittaman mallimatriisin päivityksen tuloksena saadun hajontamatriisin determinanttia ei tarvitse laskea eksplisiittisesti. Riittää siis, että hajontamatriisin determinantti lasketaan eksplisiittisesti ainoastaan kerran optimoitavan mallimatriisin alustuksen yhteydessä.

7 D-optimaalisiin koeasetelmiin liittyviä algoritmeja

Tässä luvussa esitellään joukko algoritmeja, jotka tuottavat arvion D-optimaaliselle asetelmamatriisille ja siten D-optimaaliselle koeasetelmalle. Kaikille esitellyille algoritmeille on yhteistä, se että ne hyödyntävät luvussa 6 esiteltyjä rivinvaihtotekniikoita. Korostetaan, että esiteltyt algoritmit eivät välttämättä tuota yhtälön 13 toteuttavaa D-optimaalista mallimatriisia, vaan algoritmit saattavat tuottaa vain likimääräisen arvion.

Otetaan käyttöön seuraavat merkinnät. Olkoon D optimoitava mallimatriisi, joka on alustettu poimimalla n riviä matriisista X_M . Lisäksi matriisi S koostuu niistä matriisin X_M riveistä, jotka eivät kuulu matriisiin D .

7.1 DETMAX-algoritmi

DETMAX-algoritmi on esitelty julkaisussa Mitchell (2000), ja siihen liittyvä pseudokoodi on esitetty alla (algoritmi 1). Lähtökohtana on optimoitava mallimatriisi D , jonka rivejä poistetaan tai siihen lisätään rivejä yksi kerrallaan.

```
Alusta matriisi  $D$  poimimalla satunnainen mallimatriisi.
Alusta joukko  $F$  tyhjäksi joukoksi.
while matriisin  $D$  rivien lukumäärä on välillä  $[n-k, n+k]$  do
  if matriisin  $D$  rivien lukumäärä on  $n$  then
    | Valitse satunnaisesti rivin poistamisen ja lisäämisen välillä
  end
  else if matriisin  $D$  rivien lukumäärä on suurempi kuin  $n$  then
    | if matriisi  $D$  ei kuulu joukkoon  $F$  then
    |   | Poista sellainen rivi  $\chi_j$ , jolle  $d(\chi_j)$  on mahdollisimman pieni
    |   end
    | else
    |   | Lisää sellainen rivi  $\chi_i$ , jolle  $d(\chi_i)$  on mahdollisimman suuri
    |   end
    end
  else if matriisin  $D$  rivien lukumäärä on pienempi kuin  $n$  then
    | if matriisi  $D$  ei kuulu joukkoon  $F$  then
    |   | Lisää sellainen rivi  $\chi_i$ , jolle  $d(\chi_i)$  on mahdollisimman suuri
    |   end
    | else
    |   | Poista sellainen rivi  $\chi_j$ , jolle  $d(\chi_j)$  on mahdollisimman pieni
    |   end
    end
  if matriisia  $D$  vastaavan hajontamatriisin  $H$  determinantti ei supistunut then
    | Lisää matriisi  $D$  joukkoon  $F$ 
  end
  else
    | Tyhjennä joukko  $F$ 
  end
end
end
```

Algoritmi 1. DETMAX-algoritmi.

DETMAX-algoritmi sallii niin kutsutut *ekskursiot*, jonka nojalla matriisiin D rivien lukumäärää voidaan kasvattaa tai vähentää siten, että matriisin D rivien lukumäärä voi vaihdella välillä $n-k$ ja $n+k$, missä $k=6$ (Mitchell, 2000, luku 2.2). Matriisin D rivien lukumäärästä riippuen (ks. algoritmi 1), matriisiin D voidaan lisätä rivi χ_j , siten, että valittu rivi maksimoi funktion $d(\chi_x)$ arvon (ks. yhtälö 15). Vastaavasti matriisista D voidaan poistaa rivi χ_j siten, että poistettava rivi minimoi funktion $d(\chi_x)$ arvon (ks. yhtälö 16).

Mikäli edellä kuvatut operaatiot eivät johda hajontamatriisin determinantin suppenemiseen, lisätään kaikki ekskursion aikana luodut mallimatriisit D joukkoon F , joka siis sisältää ns. epäonnistuneet koeasetelmat (*a set of "failure designs"*, ks. Mitchell, 2000, luku 2.2). Uuden ekskursion lähtökohtana on poimittu joukkoon F kuuluva matriisi, jota vastaavan hajontamatriisin determinantti on pienin mahdollinen.

Mitchell (2000, luku 3) tarkasteli DETMAX-algoritmin suorituskykyä soveltamalla sitä useisiin kaksitasoiseen koesuunnitelmiin. Näiden tulosten perusteella tehtiin seuraavat johtopäätökset: 1) sallimalla mallimatriisien rivien määrän vaihdella tilapäisesti (edellä mainitut ekskursiot), voitiin eräissä tapauksissa välttää se, että algoritmi löytää

vain lokaalin ratkaisun optimointiongelmaan, ja 2) DETMAX-algoritmin suorituskyky katsottiin tyydyttäväksi tapauksissa, joissa tekijöiden lukumäärä on pienempi kuin kahdeksan.

7.2 Fedorovin algoritmi

Fedorovin algoritmossa (de Aguilar et al., 1995, luku 6) optimoitavan mallimatriisin D rivien lukumäärä n pidetään vakiona, toisin kuin DETMAX-algoritmossa. Fedorovin algoritmiin liittyvä pseudokoodi on esitetty alla (algoritmi 2).

Fedorovin algoritmossa käydään läpi rivipareja (χ_i, χ_j) missä χ_i on matriisin D rivi ja χ_j on matriisin S sisältyvä rivi. Kullekin riviparille lasketaan sitä vastaavan delta-funktion arvo (ks. yhtälö 18) ja pareista valitaan se, jota vastaava delta-funktio on arvoltaan suurin. Mikäli tällaisia pareja on useita, valitaan niistä satunnaisesti yksi. Seuraavaksi näin valitut rivit χ_i ja χ_j vaihdetaan poistamalla matriisista D rivi χ_i ja lisäämällä siihen rivi χ_j . Lopuksi päivitetään hajontamatriisin determinantti yhtälön 17 perusteella. Tätä prosessia toistetaan, kunnes ei enää löydetä sellaista paria, jolle pätee $\Delta(\chi_i, \chi_j) > \Delta_\epsilon$, missä Δ_ϵ on valittu kynnsarvo (esimerkiksi 10^{-6}).

```
Alusta matriisi  $D$  poimimalla satunnainen mallimatriisi
Laske vastaava hajontamatriisi  $H$ 
Luo ehdokasmatriisia vastaava mallimatriisi  $X_M$ 
Aseta  $\Delta_{max} = 0$ 
while löytyy pari  $(\chi_i, \chi_j)$ , jolle pätee  $\Delta(\chi_i, \chi_j) > \Delta_\epsilon$  do
  for matriisin  $D$  rivi  $\chi_i$  do
    for matriisin  $S$  rivi  $\chi_j$  do
      laske  $\Delta(\chi_i, \chi_j)$ 
      if  $\Delta(\chi_i, \chi_j) \geq \Delta_{max}$  then
        tallenna pari  $(\chi_i, \chi_j)$ 
        päivitä  $\Delta_{max}$  tarvittaessa
      end
    end
  end
end
if  $\Delta_{max} > 0$  then
  if useammalla parilla  $(\chi_i, \chi_j)$  on sama  $\Delta_{max}$  then
    | valitse pari satunnaisesti
  end
  vaihda rivit  $\chi_i$  ja  $\chi_j$ 
  päivitä hajontamatriisin determinantti
  aseta  $\Delta_{max} = 0$ 
end
end
```

Algoritmi 2. Fedorovin algoritmi.

de Aguilar et al. (1995) tarkastelivat Fedorovin algoritmin suorituskykyä tapauksessa, jossa ehdokasmatriisin koko oli 257×2 . Lisäksi luvun n annettiin ensin vaihdella välillä 7-14. Tässä tapauksessa algoritmi tuotti parhaan arvion D-optimaaliselle koeasetelmalle

kun luvun n arvo oli 10. Toisaalta kun n oli suurempi kuin 14 todettiin, että algoritmin käyttäminen ei ole enää mielekäästä johtuen kasvaneista laskenta-ajoista.

Cook & Nachtsheim (1980) vertailivat keskenään viittä eri D-optimointiin liittyvää algoritmia (mukaan lukien Fedorovin algoritmi) ja totesivat, että kohtuuttomien laskenta-aikojen vuoksi Fedorovin algoritmia ei ole mielekäästä käyttää kun luku n on riittävän suuri. Toisaalta Fedorovin algoritmilla tuotti parempia arvioita D-optimaalisille koeasetelmille verrattuina vastaaviin DETMAX-algoritmin tuottamiin koeasetelmiin (Cook & Nachtsheim, 1980, taulukko 6).

7.3 Muunnettu Fedorovin algoritmi

Cook & Nachtsheim (1980, luku 2.4) esittivät muunnoksen Fedorovin algoritmiin (ks. algoritmi 3 alla). Tämä algoritmi perustuu siihen, että matriisia D ja hajontamatriisin H determinanttia päivitetään heti sen jälkeen, kun matriisin D yksittäistä riviä χ_j vastaavat parit (χ_i, χ_j) on käyty läpi.

```
Alusta matriisi  $D$  poimimalla satunnainen mallimatriisi
Laske vastaava hajontamatriisi  $H$ 
Luo ehdokasmatriisia vastaava mallimatriisi  $X_M$ 
Aseta  $\Delta_{max} = 0$ 
while löytyy pari  $(\chi_i, \chi_j)$ , jolle pätee  $\Delta(\chi_i, \chi_j) > \Delta_\epsilon$  do
  for matriisin  $D$  rivi  $\chi_i$  do
    for matriisin  $S$  rivi  $\chi_j$  do
      laske  $\Delta(\chi_i, \chi_j)$ 
      if  $\Delta(\chi_i, \chi_j) \geq \Delta_{max}$  then
        tallenna pari  $(\chi_i, \chi_j)$ 
        päivitä  $\Delta_{max}$  tarvittaessa;
      end
      if  $\Delta_{max} > 0$  then
        if useammalla parilla  $(\chi_i, \chi_j)$  on sama  $\Delta_{max}$  then
          valitse pari satunnaisesti
        end
        vaihda rivit  $\chi_i$  ja  $\chi_j$ 
        päivitä hajontamatriisin determinantti
        aseta  $\Delta_{max} = 0$ 
      end
    end
  end
end
end
```

Algoritmi 3. Muunnettu Fedorovin algoritmi.

Havainnollistetaan Fedorovin ja muunnetun Fedorovin algoritmien eroa tapauksessa, jossa $n=5$ ja $M=20$. Nyt Fedorovin algorithmi käy läpi yhteensä $5 \times (20-5) = 5 \times 15 = 75$ riviparia, ennen kuin matriisia D päivitetään. Muunnettu Fedorovin algoritmi voi suorittaa ensimmäisen päivityksen kun on käyty läpi 15 riviparia, toisen päivityksen kun on käyty läpi yhteensä 30 riviparia ja niin edelleen. Muunnettu Fedorovin algoritmi voi siis johtaa

viiteen päivitykseen verrattuna Fedorovin algoritmin yhteen päivitykseen, kun 75 riviparia on käyty läpi.

Cook & Nachtsheim (1980) vertailivat myös näiden kahden algoritmin suorituskykyä. Vertailun lopputuloksena oli, että muunnettu Fedorovin algoritmi voi jopa puolittaa vaaditun laskenta-ajan. Lisäksi muunnettu Fedorovin algoritmi tuotti lähes yhtä tarkkoja arvioita D-optimaaliselle koeasetelmalle kuin Fedorovin (alkuperäinen) algoritmi.

7.4 *k*-vaihtoalgoritmi

Niin kutsuttu *k*-vaihtoalgoritmi (*k-exchange algorithm*) on muunnetun Fedorovin algoritmin yleistys (Johnson & Nachtsheim, 1983). *k*-vaihtoalgoritmissa valitaan luku *k*, missä $k \leq n$. Erona muunnettuun Fedorovin algoritmiin on se, että kaikkia matriisin **D** rivejä χ_j ei käydä silmukassa läpi (vrt. algoritmi 3), vaan ainoastaan matriisin **D** *k* riviä. Nämä *k* riviä χ_j on valittu siten, että niitä vastaavat funktion $d(\chi_j)$ arvot ovat pienimmät mahdolliset. Kirjallisuudessa luvulle *k* on suositeltu arvoja 3 tai 4 (Johnson & Nachtsheim, 1983) ja lisäksi ylärajaa $k \leq n/4$ (Meyer & Nachtsheim, 1995).

Havainnollistetaan *k*-vaihtoalgoritmin ja muunnetun Fedorovin algoritmin laskennallista eroa tapauksessa, jossa $n=5$, $M=20$ ja $k=3$. Nyt muunnettu Fedorovin algoritmi käy läpi yhteensä 5×15 riviparia ja suorittaa korkeintaan 5 päivitystä matriisin **D** kaksoissilmukan suorittamisen yhteydessä. Toisaalta *k*-vaihtoalgoritmi käy läpi vastaavassa kaksoissilmukassa yhteensä 3×15 riviparia ja suorittaa sen yhteydessä korkeintaan kolme päivitystä.

Johnson & Nachtsheim (1983, taulukko 4) suorittaman vertailun perusteella *k*-vaihtoalgoritmi on laskennallisesti tehokkaampi kuin muunnettu Fedorovin algoritmi. Toisaalta tutkimuksessa todettiin, että *k*-vaihtoalgoritmi ei tuota kaikissa tapauksissa yhtä tarkkoja arvioita D-optimaaliselle koeasetelmalle.

7.5 *kl*-vaihtoalgoritmi

Niin kutsutussa *kl*-vaihtoalgoritmissa poimitaan *k* riviä, missä $k \leq n$, matriisista **D** ja *l* riviä matriisista **S**, missä $l \leq M$ (Atkinson & Donev, 1989). Optimoitavaa matriisia **D** päivitetään samalla tavalla kuin Fedorovin algoritmissa (vrt. algoritmi 2 yllä), mutta algoritmiin liittyvissä silmukoissa huomioidaan vain valitut matriisien **D** ja **S** rivit.

Lukujen *k* ja *l* valintaan Atkinson & Donev (1989) esittivät kaksi ratkaisua. Ensimmäisessä ratkaisussa valitaan *k* riviä χ_i matriisista **D** siten, että näitä rivejä vastaavat funktion $d(\chi_i)$ arvot ovat pienimmät mahdolliset. Lisäksi valitaan *l* riviä χ_j matriisista **S** siten, että näitä rivejä vastaavat funktion $d(\chi_j)$ arvot ovat suurimmat mahdolliset. Toinen esitetty ratkaisu on se, että valitaan *k* riviä matriisista **D** ja *l* riviä matriisista **S** satunnaisesti.

Havainnollistetaan kl -vaihtoalgoritmin ja Fedorovin algoritmin eroa tapauksessa, jossa $n=5$, $M=20$ ja $k=l=3$. Tällöin edellinen algoritmi laskee vain $3 \times 3 = 9$ delta-funktion arvoa ennen matriisin D (mahdollista) päivitystä, siinä missä Fedorovin algoritmi laskee 75 delta-funktion arvoa.

Atkinson & Donev (1989) eivät tarkastelleet kl -vaihtoalgoritmin suorituskykyä suhteessa Fedorovin algoritmiin tai k -vaihtoalgoritmiin. On kuitenkin ilmeistä, että kl -vaihtoalgoritmin laskennallinen tehokkuus ja tarkkuus riippuvat lukujen k ja l arvoista.

8 Keskustelu

Kirjallisuudesta ei löytynyt kattavaa vertailua luvussa 7 esiteltyjen algoritmien suhteellisesta laskennallisesta tehokkuudesta ja/tai tarkkuudesta. On myös huomattava, että löydetty lähteet ovat suurilta osin 80-luvulta ja ovat siten suhteellisen vanhoja.

Tarkasteltujen algoritmien suorituskykyä tulisi testata kattavasti hyödyntäen nykyistä laskentakapasiteettia. Tällaisessa tarkastelussa olisi kiinnitettävä huomiota erityisesti siihen, että voidaanko k - ja kl -vaihtoalgoritmeihin (ks. luvut 7.4 ja 7.5) liittyvien vapaiden parametrien (k ja l) arvoille löytää optimaaliset, koeasetelmasta riippumattomat arvot. Kysymys siitä, että mikä tarkastelluista algoritmeista on suorituskyvyltään paras, jää siis toistaiseksi avoimeksi.

D-optimointiin liittyviä uudempia menetelmiä ovat esimerkiksi *coordinate-exchange algorithm* (Meyer & Nachtsheim, 1995), geneettisiin algoritmeihin perustuva lähestymistapa (Heredia-Langner et al., 2003), *cocktail algorithm* (Yu, 2011), ja *randomized exchange algorithm* (Harman et al., 2020). Näiden algoritmien käsittely ja niiden vertailu luvussa 7 esitettyihin algoritmeihin ovat tämän työn aihepiirin ulkopuolella, sillä mainitut algoritmit poikkeavat huomattavasti lähtökohdiltaan ja toteutuksiltaan tässä tutkielmassa käsitellyistä rivinvaihtoalgoritmeista.

Tässä työssä tarkastellut algoritmit perustuvat sellaisten optimiasetelmien löytämiseen, jotka toteuttavat D-optimaalisen koeasetelman kriteerin (ks. yhtälö 13). Muita optimiasetelman kriteerejä ovat esimerkiksi A-, E-, ja G- optimaalisuus (Atkinson & Donev, 1992, luku 10). Näiden kriteerien esittely on kuitenkin tämän työn aihepiiriin ulkopuolella. Tämä johtuu siitä, että eri kriteerien käsittely ja niihin liittyvien algoritmien esittely vaatisivat erillisen, tätä tutkielmaa huomattavasti laajemman tarkastelun.

9 Yhteenveto

D-optimaalisen koeasetelman etsiminen siten, että käydään läpi kaikki koesuunnitelman mukaiset mallimatriisit ja lasketaan näitä vastaavien hajontamatriisien determinantit, voi osoittautua haasteeksi laskennallisessa mielessä. Tarvitaan siis algoritmeja, jotka tuottavat likimääräisen arvion D-optimaaliselle koeasetelmalle.

Tässä työssä käytiin läpi algoritmeja, jotka perustuvat optimoitavan matriisin päivittämiseen rivi kerrallaan. Esitellyistä algoritmeista k - tai kl -vaihtomenetelmä (ks. luvut 7.4 ja 7.5) tarjoavat parhaan kompromissin tarkkuuden ja laskenta-ajan välillä tarkasteltujen lähteiden perusteella. Tämä johtopäätös perustuu osittain siihen, että mainittuihin algoritmeihin liittyy ns. vapaita parametreja (k ja l), joiden arvoja voidaan säätää tapauskohtaisesti. Lisäksi k - ja kl vaihtomenetelmät ovat yleistyksiä eräästä tutkielmassa esitellyistä algoritmeista (muunnettu Fedorovin algoritmi, ks. luku 7.3).

Kirjallisuudesta ei kuitenkaan löytynyt kattavaa vertailua työssä esiteltyjen algoritmien suhteellisesta suorituskyvystä. Johtopäätöksenä on, että tällainen vertaileva tutkimus toisi selvyyttä eri rivinvaihtoalgoritmien paremmuuteen.

Lähdeluettelo

- Alho, J., Arjas, E., Läärä, E., & Pere, P. (2021). *Tilastotieteen sanasto*. Suomen Tilastoseura ry.
- Atkison A. C., & Donev, A. N. (1989). The construction of exact D-optimum experimental designs with application to blocking response surface designs. *Biometrika*, 76(3), 515–526. <https://doi.org/10.1093/biomet/76.3.515>
- Atkinson, A. C., & Donev, A. N. (1992). *Optimum experimental designs*. Clarendon Press
- Cook, R. D., & Nachtsheim, C. J. (1980). A comparison of algorithms for constructing exact D-optimal designs. *Technometrics* 22(3), 315-324. <https://doi.org/10.1080/00401706.1980.10486162>
- de Aguiar, P. F., Bourguignon, B., Khots, M. S., Massart, D. L., & Phan-Thao-Luu, R. (1995). D-optimal designs. *Chemometrics and Intelligent Laboratory Systems*, 30(2), 199–210. [https://doi.org/10.1016/0169-7439\(94\)00076-X](https://doi.org/10.1016/0169-7439(94)00076-X)
- Durakovic, B. (2017). Design of experiments application, concepts, examples: State of the art. *Periodicals of Engineering and Natural Sciences (PEN)*, 5(3), 421–439. <https://doi.org/10.21533/pen.v5i3.145>
- Harman, R., Filová, L., & Richtárik, P. (2020). A randomized exchange algorithm for computing optimal approximate designs of experiments. *Journal of the American Statistical Association*, 115(529), 348–361. <https://doi.org/10.1080/01621459.2018.1546588>
- Heredia-Langner, A., Carlyle, W. M., Montgomery, D. C., Borrór, C. M., & Runger, G. C. (2003). Genetic algorithms for the construction of D-optimal designs. *Journal of Quality Technology*, 35(1), 28–46. <https://doi.org/10.1080/00224065.2003.11980189>
- Ilzarbe, L., Álvarez, M. J., Viles, E., & Tanco, M. (2008). Practical applications of design of experiments in the field of engineering: a bibliographical review. *Quality and Reliability Engineering International*, 24(4), 417–428. <https://doi.org/10.1002/qre.909>
- Johnson, M. E., & Nachtsheim, C. J. (1983). Some guidelines for constructing exact D-optimal designs on convex design spaces. *Technometrics*, 25(3), 271–277.

<https://doi.org/10.1080/00401706.1983.10487880>

Kutner, M., C. Nachtsheim, J. Neter, & W. Li. (2005). *Applied Linear Statistical Models*. McGraw-Hill, Irwin

Meyer, R. K. & Nachtsheim, C. J. (1995). The coordinate-exchange algorithm for constructing exact optimal experimental designs. *Technometrics*, 37(1), 60-69. <https://doi.org/10.1080/00401706.1995.10485889>

Mitchell, T. J. (2000). An algorithm for the construction of “D-Optimal” experimental designs. *Technometrics*, 42(1), 48–54. <https://doi.org/10.1080/00401706.2000.10485978>

Montgomery, D. C. (2013). *Design and analysis of experiments*. John Wiley & Sons Inc.

Yu, Y. (2011). D-optimal designs via a cocktail algorithm. *Statistics and Computing*, 21(4), 475–481. <https://doi.org/10.1007/s11222-010-9183-2>