TAMPEREEN TEKNILLINEN YLIOPISTO

JARKKO PEKKARINEN

Suuridimensioisen optimointiongelman ratkaiseminen numeerisesti kineettisten parametrien suurimman uskottavuuden estimaatin määrittämiseksi PET-sinogrammiaikasarjasta

Diplomityö

Tarkastaja: Ulla Ruotsalainen
Tarkastaja ja aihe hyväksytty
Tieto- ja sähkötekniikan tiedekuntaneuvoston kokouksessa
12.01.2011

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO
Signaalinkäsittelyn ja tietoliikennetekniikan koulutusohjelma
JARKKO PEKKARINEN: Suuridimensioisen optimointiongelman ratkaiseminen numeerisesti kineettisten parametrien suurimman uskottavuuden estimaatin määrittämiseksi PET-sinogrammiaikasarjasta
Diplomityö, 51 sivua, 14 liitesivua
Elokuu 2011
Pääaine: Signaalinkäsittely
Tarkastaja: professori Ulla Ruotsalainen
Ohjaajat: Harri Pölönen, Jari Niemi, Ferrante Neri
Avainsanat: positroniemissiotomografia, numeerinen optimointi, konjugaattigradientti

Positroniemissiotomografia (positron emission tomography, PET) on lääketieteellinen kuvantamismenetelmä, jonka mittausdatan analysoinnissa perinteisesti käytettyjen lähestymistapojen yhteydessä tapahtuu informaatiohävikkiä eikä ole mahdollista saavuttaa optimaalisia tuloksia. Tässä diplomityössä keskitytään uuteen, mallintamiseen perustuvaan PET-mittausdatan analysointimenetelmään, joka tuottaa estimaatin kuvattavan kudoksen heterogeenisuudesta hyödyntäen mittausten kaiken informaatiosisällön. Työn tavoitteena on menetelmän käytännön toteutuksen kehittäminen ja nopeuttaminen sekä menetelmän toimivuuden tutkiminen simulaatioiden avulla tietyissä koeasetelmissa.

Merkittävin laskenta-aikaa kuluttava vaihe menetelmän toteutuksessa on kudoksen ominaisuuksia kuvaavien mallin parametrien suurimman uskottavuuden estimointi, joka joudutaan suorittamaan numeerisella optimointialgoritmilla. Menetelmän teoria edellyttää tämän suuridimensioisen optimointiongelman ratkaisemisen, jotta löydettäisiin suurimman uskottavuuden estimaatti. Useiden, keskenään erilaisten optimointialgoritmien suorituskykyä vertaillaan optimointiongelman ratkaisemiseksi ja päätelmien tekemiseksi optimointiavaruuden ominaisuuksista.

Tärkeäksi työssä tehtäväksi havainnoksi muodostui konjugaattigradienttialgoritmien potentiaali, kun yksinkertainen vakioaskelta käyttävä Fletcher-Reeves -algoritmi suoriutui optimointimenetelmien vertailuissa parhaiten. Aiemmin oletusarvoisesti käytetyn Barzilai-Borwein -gradienttioptimointimenetelmän suorituskykyä onnistuttiin myös parantamaan huomattavasti tämän diplomityön yhteydessä kehitettyjen modifiointien avulla. Toinen merkittävä havainto koskee parametriestimaattien käyttäytymistä optimoinnin edetessä. Havainto mahdollistaa merkittäviä laskenta-ajallisia säästöjä, kun optimointi voidaan keskeyttää jo ennen suurimman uskottavuuden estimaatin saavuttamista ilman, että tuloksiin muodostuu merkittävää virhettä.

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Programme in Signal Processing and Communications Engineering
JARKKO PEKKARINEN: Numerically solving a large-scale optimization problem to acquire maximum likelihood estimate of kinetic parameters from PET sinogram time series
Master of Science Thesis, 51 pages, 14 Appendix pages
August 2011
Major: Signal Processing
Examiner: professor Ulla Ruotsalainen
Instructors: Harri Pölönen, Jari Niemi, Ferrante Neri
Keywords: positron emission tomography, numerical optimization, conjugate gradient

Positron emission tomography is a medical imaging method, where with the traditionally applied approaches in data analysis a loss of information occurs making it impossible to achieve optimal results. This thesis focuses on a new, modeling-based approach to PET data analysis that makes use of the whole information content of the measurements to produce a heterogeneity estimate of the tissue that is being studied. The aim of this thesis is to improve and make faster the implementation of this method and to study its performance under certain test conditions by using simulations.

The most time consuming phase in the implementation of the method is the maximum likelihood estimation of the model parameters that describe the tissue properties. Numerical optimization algorithms have to be utilized to carry out this estimation. The theory of the method requires the exact global solution to the high-dimensional optimization problem in order to determine the maximum likelihood estimate. The performance of many different types of optimization algorithms is compared in order to solve the optimization problem and to make observations of the properties of the optimization space.

An important discovery made in the thesis is the potential of conjugate gradient methods, as a simple Fletcher-Reeves algorithm using constant step size performed best in the comparisons between optimization methods. Also the performance of the Barzilai-Borwein optimization algorithm, which has been used earlier as the standard optimization method, is managed to be boosted noticeably by using modifications developed for this thesis. Another important discovery is about the behavior of the parameter estimates during the optimization procedure. The discovery allows huge computational savings as optimization can be interrupted even before reaching the maximum likelihood estimate without causing any meaningful error.

ALKUSANAT

Tämä diplomityö on tehty Tampereen teknillisen yliopiston signaalinkäsittelyn laitoksen M^2oBSI-ryhmässä Suomen Akatemian huippuyksikkörahoituksella 213462.

Haluaisin erityisesti kiittää ohjaajiani Harri Pölöstä ja Jari Niemeä, jotka ovat sekä työ- että vapaa-ajallaan nähneet vaivaa opastaakeen minua työn aihepiiriin, kysymyksiini vastaamiseen ja työn väliversioiden lukemiseen. Useimmat diplomityön sisältöön vaikuttaneista ideoista ovat ilmaantuneet ohjaajieni kanssa käydyissä keskustelutuokioissa tai suoraan ohjaajieni ehdottamina. Hyviä neuvoja diplomityöni sisältöön liittyen olen saanut myös työni tarkastajalta Ulla Ruotsalaiselta ja Ferrante Neriltä, jonka kanssa yhteistyössä tuotimme konferenssijulkaisun, jossa esitellään osittain samoja tuloksia kuin tässä diplomityössä.

Suuri kiitos kuuluu myös kihlatulleni Saralle ja koirallemme Netalle. Heidän seuransa on tuonut rentouttavaa vastapainoa työlle ja opiskelulle.

Tampereella 12.07.2011

Jarkko Pekkarinen

SISÄLLYS

1. Johdanto . . . 1
2. PET-kuvantaminen . . . 4
2.1 Radon-muunnos . . . 5
2.2 Lokeromalli . . . 7
3. Virhekorjatun heterogeenisuusestimaatin laskenta PET-mittauksista . . . 9
3.1 Mittausmalli . . . 9
3.2 Suurimman uskottavuuden estimointi . . . 10
3.3 Toistokoemenetelmä estimointivirheen korjaamiseksi . . . 12
4. Vertailtavat optimointimenetelmät . . . 15
4.1 Nollannen asteen menetelmät . . . 16
4.1.1 Monilentoratahaku . . . 17
4.1.2 Itseohjautuva differentiaalinen evoluutio . . . 19
4.1.3 Kattavasti oppiva hiukkasparvioptimointi . . . 21
4.2 Ensimmäisen asteen menetelmät . . . 24
4.2.1 Jyrkimmän laskun menetelmä . . . 24
4.2.2 Fletcher-Reeves . . . 25
4.2.3 Barzilai-Borwein . . . 25
4.2.4 Barzilai-Borwein modifiointi: Useampi askelpituus . . . 26
4.2.5 Barzilai-Borwein modifiointi: Polynomisovitteella ennustaminen . . . 27
5. Kokeet . . . 30
5.1 Koeasetelma . . . 30
5.1.1 Simuloitu mittausdata ja sen mallintaminen . . . 30
5.1.2 Optimointimenetelmien vertailemisesta . . . 32
5.2 Koe 1: Optimointimenetelmien vertailu . . . 34
5.2.1 2K-lokeromallia käyttäen . . . 36
5.2.2 3K-lokeromallia käyttäen . . . 36
5.3 Koe 2: Nopeimman käytännöllisen optimointialgoritmin määrittäminen 38
5.4 Koe 3: Estimointivirheen korjaaminen, kun ML-estimaattia ei saavuteta 42
5.5 Koe 4: Sijainnin vaikutus estimointivirheeseen . . . 44
6. Yhteenveto . . . 47
Lähteet . . . 49
A.Liite: Monilentoratahaun lähistöhakumenetelmät . . . 52
B.Liite: Kokeiden tuloskuvaajat . . . 56

TERMIT JA SYMBOLIT

∇	Gradientti
α	Gradienttioptimointimenetelmien askelpituus
μ	Odotusarvo
σ	Keskipoikkeama
θ	Parametrivektori
θ^{mle}	Parametrivektorin suurimman uskottavuuden estimaatti
Λ	Virtuaalinen aktiivisuuskonsentraatiokuva
λ	Radioaktiivinen hajoamisvakio
BB	Barzilai-Borwein -gradienttioptimointimenetelmä
C	Radioaktiivisuuskonsentraatio lokeromallin lokerossa
$CLPSO$	Kattavasti oppiva hiukkasparvioptimointi
F	Plasman tilavuusosuus kudoksessa
FR	Fletcher-Reeves -gradienttioptimointimenetelmä
jDE	Itseohjautuva differentiaalinen evoluutio
M	Mallinnettu sinogrammiaikasarja, joka huomioi virhelähteet
MTS	Monilentoratahaku
PET	Positroniemissiotomografia
S	Ideaalinen mallinnettu sinogrammiaikasarja
sc	Simuloidun kuvantamisen hyötysuhdevakio
SD	Jyrkimmän laskun gradienttioptimointimenetelmä
W	Systeemimatriisi PET-kuvantamisjärjestelmän mallintamiseksi Radon-muunnoksella

Bin	Sinogrammimatriisin alkio, yhden detektoriparin mittaama data yhdestä kulmasta.
Etsintäsuunta	Suunta johon optimoitavaa parametrivektoria muutetaan askelpituuden verran.
Evolutiivinen algoritmi	Optimointialgoritmi, joka perustuu johonkin biologisen evoluution mekanismiin.
Fantomi	Ominaisuuksiltaan tunnettu mittauskohde, jota käytetään referenssinä kuvantamisjärjestelmän toimintaa analysoitaessa.
Heterogeenisuus	Sekakoosteisuus, epäyhtenäisyys.
Homogeenisuus	Tasakoosteisuus, yhtenäisyys.
Kohdefunktio	Funktio, jonka parametreja optimoidaan.
Konjugaattisuunta	Etsintäsuunta, joka määritellään gradienttisuunnan ja edellisen iteraation suunnan painotettuna summana. Painon määritelmä riippuu optimointimenetelmästä.
Lokeromalli	Malli merkkiaineen fysiologisen käyttäytymisen mallintamiseksi.
Merkkiaine	Kehon sisäiseltä käyttäytymiseltä tunnettu yhdiste, johon on liitetty β^+-aktiivinen isotooppi. PET:n mittaaman gammasäteilyn aiheuttaja.
ML-estimointi	Suurimman uskottavuuden estimointi. Menetelmä, jota käytetään tilastollisen mallin sovittamiseksi mittausaineistolle.
Radon-muunnos	Muunnos, joka mallintaa tomografiamittauksen.
Sinogrammi	PET-mittauksen tulosten matriisiesitysmuoto, jossa eri kulmista mitattu data on esitetty omilla riveillään.
Uskottavuusfunktio	Nimitys tiheysfunktiolle ML-estimoinnin yhteydessä, kun aineisto oletetaan kiinnitetyksi ja tiheysfunktion parametrit muuttujiksi.
Vokseli	Tilavuusalkio, kolmiulotteinen pikseli.

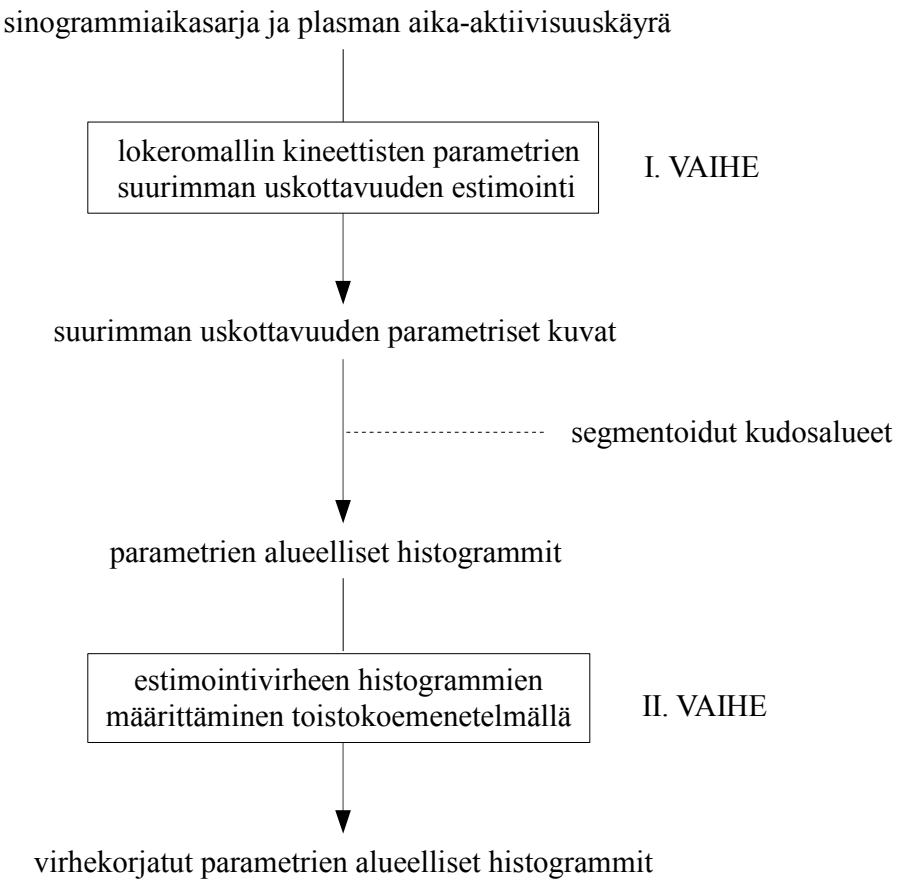
1. JOHDANTO

Heterogeenisuudella tarkoitetaan sekakoosteisuutta tai epäyhtenäisyyttä. Joissain tapauksissa syöpäkudos on heterogeenisempaa kuin vertailukohtana käytettävä terve kudos ja siten kudoksen heterogeenisuuden luotettavasta määrittämisestä voisi olla hyötyä mm. diagnostiikassa.

Positroniemissiotomografia on laajalti käytetty funktionaalisen kuvantamisen menetelmä, jossa potilaaseen injektoidaan radioaktiivista merkkiainetta liitettynä molekyyleihin, joiden biologinen käyttäytyminen oletetaan tunnetuksi. Merkkiaineen kulkeutumista seuraamalla saadaan informaatiota kehon toiminnasta. Kohteesta saatavat mittaukset ovat merkkiaineen gamma-aktiivisuuden projektioita, jotka mitataan kohteen ympärillä eri kulmissa sijaitsevien detektoreiden avulla. Mikäli mitatulle informaatiolle halutaan paikallistaa sijainti kohteen sisällä, esimerkiksi muodostamalla poikkileikkauskuvia kohteen aktiivisuudesta, törmätään perinteisillä, kuvarekonstruktioon perustuvilla lähestymistavoilla (esim. [1, s. 397]) inversio-ongelmaan, jonka yhteydessä on turvauduttu mittausdatassa esiintyvän kohinan regularisointiin mielekkäiden tulosten saavuttamiseksi. Regularisointi vaikuttaa datan informaatiosisältöön eikä optimaalisia tuloksia siten ole mahdollista saavuttaa.

Jotta PET-tutkimuksen kaikki mittausinformaatio saataisiin datan analysointivaiheessa hyödynnettyä on kehitetty menetelmiä, jotka esimerkiksi tuottavat estimaatin kudoksen heterogeenisuudesta suoraan mitatuista projektioista ilman minkäänlaista regularisointi- tai rekonstruktiovaihetta. Eräs tällainen tunnettu menetelmä perustuu EM-algoritmiin (expectation-maximization) [2]. Suoraan projektioita hyödyntää myös Tampereen teknillisen yliopiston signaalinkäsittelyn laitoksen M^2oBSI-ryhmässä kehitetty, mallintamiseen ja suurimman uskottavuuden estimointiin perustuva menetelmä [3][4], johon tässä diplomityössä keskitytään. Menetelmä ja sen jakautuminen kahteen päävaiheeseen on havainnollistettu kuvassa 1.1.

Menetelmän syötteinä on nk. sinogrammiaikasarja, joka koostuu PET-laitteiston tuottamista aktiivisuusprojektioista, ja kuvantamisen ohessa potilaan verestä mitattu plasman aika-aktiivisuuskäyrä. Menetelmän ensimmäisessä päävaiheessa määritetään syötteiden perusteella suurimman uskottavuuden estimaatit lokeromallin kineettisille parametreille, jotka mallintavat kudoksen ominaisuuksia ja joiden alueellista keskihajontaa käytetään menetelmän yhteydessä heterogeenisuuden mittarina. Suurimman uskottavuuden estimointi muodostaa erittäin suuridimensioisen op-



Kuva 1.1: *Tässä diplomityössä käsiteltävän, heterogeenisuusestimaatin suoraan sinogrammiaikasarjasta määrittävän menetelmän toiminta. Kaaviokuvassa parametreilla tarkoitetaan aina lokeromallin kineettisiä parametreja.*

timointiongelman, joka ratkaistaan numeerista optimointialgoritmia käyttäen. Keskeisin tavoite tässä diplomityössä on juuri suurimman uskottavuuden estimaatin määrittäminen mallin parametreille kohtuullisessa ajassa, jotta menetelmä olisi käytännöllinen mahdollisimman useilla sovellusalueilla. Oletusarvoisesti tähän asti käytetylle optimointimenetelmälle tutkitaan sen suorituskykyä mahdollisesti parantavia modifiointeja ja lisäksi tutkitaan useiden erilaisten optimointimenetelmien soveltuvuutta optimointiongelman ratkaisemiseksi vertailemalla niiden suorituskykyä.

Menetelmän seuraavana vaiheena on muodostuneiden parametristen kuvien jakaminen tarkasteltaviin alueisiin. Aluejaon määrittäminen eli segmentointi on rajattu tämän diplomityön ulkopuolelle. Menetelmän toisena päävaiheena on suurimman uskottavuuden parametriestimaatteihin muodostuneen estimointivirheen alueellisten histogrammien määrittäminen nk. toistokoemenetelmää käyttäen. Virhehistogram-

mien tunteminen edelleen mahdollistaa estimointivirheen vaikutuksen korjaamisen alueellisista heterogeenisuusestimaateista.

Toissijaisina tavoitteina diplomityössä on menetelmän toteutusyksityiskohtien nopeuttaminen ja menetelmän toimivuuden tutkiminen sekä yksinkertaisempaa 2K- että etenkin monimutkaisempaa 3K-lokeromallia käyttäen, jonka toimintaa menetelmän yhteydessä tutkittiin ensimmäistä kertaa tätä diplomityötä tehtäessä. Alustavia 3K-lokeromallilla saatuja tuloksia esiteltiin konferenssijulkaisussa [5], mutta ensimmäiset kattavat tulokset esitellään tässä työssä.

2. PET-KUVANTAMINEN

Positroniemissiotomografia on funktionaalisen kuvantamisen menetelmä eli se mittaa informaatiota kehon sisäisistä prosesseista. PET perustuu yhdisteiden kehon sisäisen kulkeutumisen tarkkailemiseen. Tiettyjen yhdisteiden kulkeutuminen kehossa oletetaan tunnetuksi, joten poikkeamat mittauksissa voivat viitata esimerkiksi syöpäkasvaimen läsnäoloon. Yhdisteiden kulkeutumista saadaan mitattua kehon ulkopuolelta liittämällä niihin jokin lyhyen puoliintumisajan omaava β^+-aktiivinen isotooppi, joka emittoi hajotessaan positronin. Liittämisen jälkeistä yhdistettä kutsutaan merkkiaineeksi. Potilaaseen injektoidaan merkkiainetta ja kehon sisällä hajoamisen seurauksena syntyvät positronit annihiloituvat kohdatessaan elektronin lähellä hajoamiskohtaa. Annihiloitumisesta syntyy kaksi vastakkaisiin suuntiin kulkevaa suurienergistä gammakvanttia, jotka voidaan mitata kehon ulkopuolella detektorien avulla.

Kuvantamisen edellytyksenä on näiden vastakkaisiin suuntiin kulkevien gammakvanttien mittaaminen, joten detektoreilla on oltava vastinparit kuvattavan kohteen molemmin puolin. Sellaiset detektorien havainnot hylätään, joissa ei vastindetektorissa ole tapahtunut havaintoa. Tällä menettelyllä saadaan huomattavasti vähennettyä mm. sironnan ja taustasäteilyn aiheuttamaa kohinaa mittauksissa. Yksittäiseksi detektoriparin rekisteröimäksi hyväksytyksi mittaukseksi voidaan tulkita vastejana (line of response, LOR), jonka varrella annihilaatio on tapahtunut. Yksittäinen tällainen vastejana ei sisällä syvyysinformaatiota, joten esimerkiksi poikkileikkauskuvien muodostamista varten on kohteesta tehtävä mittauksia eri suunnista ja ratkaistava inversio-ongelma syvyysinformaation selvittämiseksi. Mittaustuloksia suoraan tarkastelemalla saadaan vain yksittäisten kulmien perspektiivistä mitattuja 2D-säteilyprofiileja.

Moderneissa PET-järjestelmissä detektoriparit pystyvät rekisteröimään havaintoajanhetket sellaisella tarkkuudella, että annihilaatio voidaan rajata tapahtuneeksi näiden välisen vastejanan segmentillä. Tarkkuus ei kuitenkaan riitä hyödylliseen syvyysinformaation tulkitsemiseen suoraan mittauksista, vaan sitä voidaan käyttää useista suunnista mitatun datan pohjalta lasketun poikkileikkauskuvan resoluution kasvattamiseen. Resoluutioon vaikuttavia tekijöitä ovat mm. detektorien määrä ja ominaisuudet, mittauskulmien määrä, merkkiaineen aktiivisuus ja kuvattavan kohteen gamma-absorptio-ominaisuudet.

Staattiseksi PET-tutkimukseksi kutsutaan tutkimusta, jossa tulokset, esimerkiksi 3D-malli päästä mitatusta säteilystä, muodostetaan kaikkia mittausaikana tehtyjä havaintoja hyödyntäen. Näin menetellen saadaan suuri määrä havaintoja tulosten muodostamista varten ja saavutetaan mahdollisimman hyvä signaali-kohina -suhde. Staattisessa tutkimuksen haittapuolena on se, että menetetään mittausajanhetkeen sidottu informaatio kuten merkkiaineen kulkeutumisvaiheet kohteessa.

Dynaamisessa tutkimuksessa havaintoja voidaan tarkastella esimerkiksi eripituisiin aikaikkunoihin (frame) jaettuna ja saada merkkiaineen kulkeutumisvaiheiden perusteella tarkempaa informaatiota merkkiaineena käytetyn yhdisteen fysiologisesta käyttäytymisestä kohteessa.

2.1 Radon-muunnos

PET-laitteiston tuottamaa mittausdataa voidaan mallintaa diskreettiä Radon-muunnosta käyttäen. Kaksiulotteisessa tapauksessa jatkuva Radon-muunnos kuvaa funktion $f(x, y)$ yksiulotteisiksi projektioiksi $p(\tau, \beta)$. Yksittäisen projektion alkio $p(\tau_0, \beta_0)$ on kuvan 2.1 mukaisesti pisteen (τ_0, β_0) kautta kulkevan, r-akselin suuntaisen suoran R_{τ_0,β_0} yli laskettu $f(x, y)$:n viivaintegraali,

$$p(\tau_0, \beta_0) = \int\limits_{R_{\tau_0,\beta_0}} f(x, y)\, dr, \tag{2.1}$$

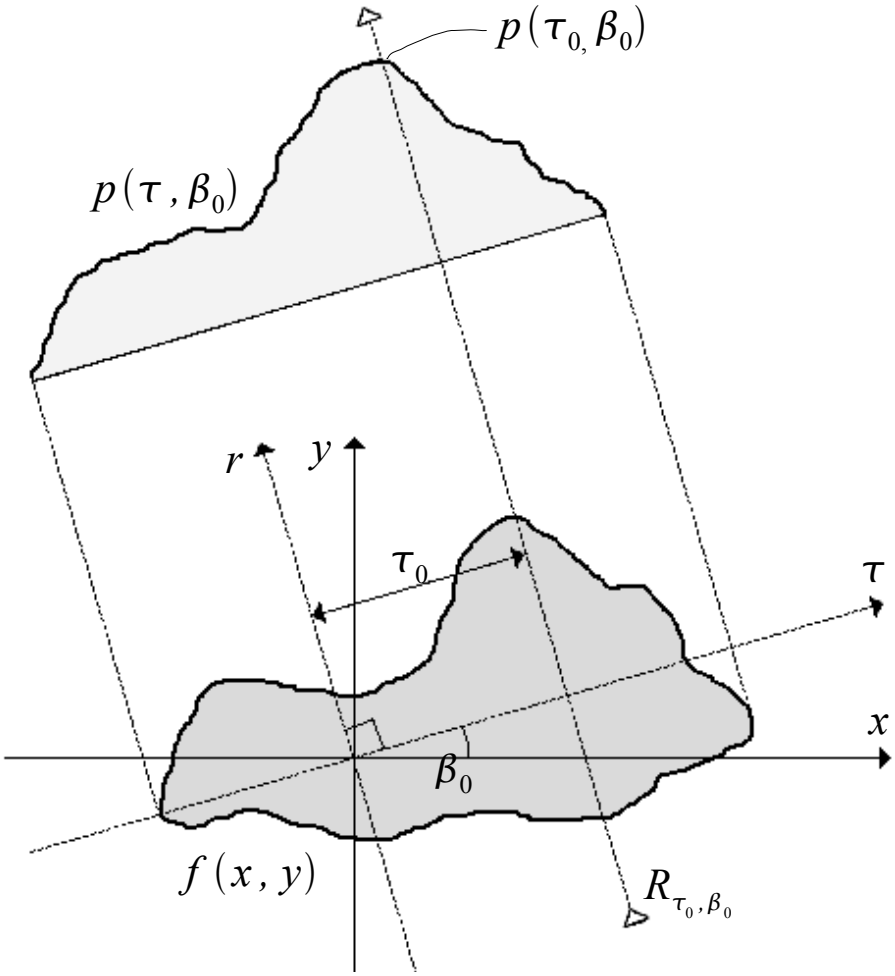
missä r ja τ määräävät koordinaatiston, joka on kulman β_0 kallistuneena suhteessa (x, y) -koordinaatistoon. [6]

Yksittäinen viivaintegraali vastaa yhden äärettömän ohuen detektoriparin mittausdataa yksittäisestä kulmasta. Radon-muunnos saadaan, kun viivaintegraali lasketaan kaikkien kulmien $\beta \in [0, \pi)$ ja siirtymien $\tau \in (-\infty, \infty)$ yli

$$p(\tau, \beta) = \int\limits_{R_{\tau,\beta}} f(x, y)\, dr. \tag{2.2}$$

Käänteinen Radon-muunnos (2.3) [6] palauttaa alkuperäisen kaksiulotteisen datan yksikäsitteisesti ideaalisista yksiulotteisista projektioista. Sen sijaan käytännön sovelluksissa, esimerkiksi PET-laitteiston mittaamalle datalle, voi syntyä käänteismuunnoksessa merkittävääkin vääristymää mittausdatassa esiintyvän kohinan seurauksena.

$$f(x, y) = \int_0^{\pi} \int_{-\infty}^{\infty} \frac{\partial p(\tau, \beta)}{\partial \tau} \frac{1}{(x \cos\beta + y \sin\beta - \tau)}\, d\tau d\beta. \tag{2.3}$$



Kuva 2.1: *Projektion muodostuminen yksittäisistä viivaintegraaleista.*

Diskreetissä Radon-muunnoksessa projektiot lasketaan äärellisellä kulmien β määrällä ja siten, että siirtymien τ ääriarvot on rajoitettu. Siirtymien ääriarvojen muodostama väli on lisäksi jaettu äärelliseen määrään tasapituisia välejä eli binejä. Yksittäinen projektion alkio saadaan nyt integroimalla yksittäisen tällaisen siirtymävolyymin yli yksittäisessä kulmassa.

Sinogrammiksi kutsutaan sellaista diskreetin Radon-muunnoksen tai PET-mittauksen tulosten matriisiesitysmuotoa, jossa eri kulmista kerätty data on esitetty omilla riveillään [1]. Kulmia vastavat rivit ovat samassa järjestyksessä kuin missä ne on laskettu tai mitattu. Sinogrammin sarakkeilla on esitetty saadut binien arvot. Todellisessa PET-laitteistossa yksi bini vastaa yhtä detektoriparia. Mikäli mittaukset ovat jakautuneet useampaan aikaikkunaan niin jokaiselle aikaikkunalle muodostetaan oma sinogramminsa ja näiden sinogrammien aikajärjestyksessä olevaa joukkoa kutsutaan sinogrammiaikasarjaksi.

2.2 Lokeromalli

Dynaamisissa PET-mittauksissa saatua informaatiota merkkiaineen käyttäytymisestä kohteessa voidaan hyödyntää kudoksen fysiologisten ominaisuuksien parametriseksi mallintamiseksi merkkiaineen suhteen. Parametrisella mallilla tarkoitetaan jonkin ilmiön todellista käyttäytymistä jäljittelevää prosessia, joka on kuvailtu mallin määritelmän ja sen sisältämien parametrien avulla. Parametrisella mallintamisella puolestaan tarkoitetaan niiden parametrien määrittämistä jotka selittävät mallinnettavat havainnot parhaiten kun oletetaan, että ne ovat syntyneet mallin määrittelemällä prosessilla.

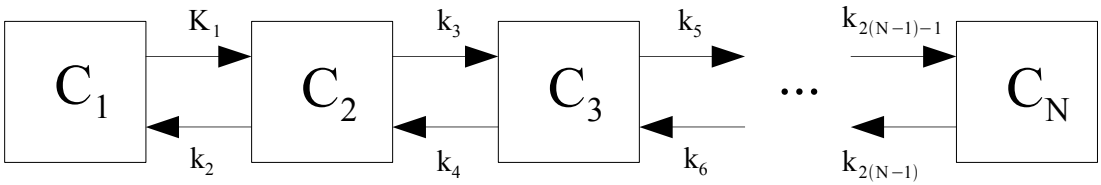
Yleinen lähestymistapa merkkiaineen fysiologisen käyttäytymisen mallintamiseksi on nk. lokeromalli [7], joka mallintaa merkkiaineen käyttäytymistä yksittäisessä kohteen tilavuusalkiossa differentiaaliyhtälöiden ja lokeroiden (compartment) avulla. Usein lokeroista ensimmäisen voidaan nähdä mallintavan merkkiaineen konsentraatiota plasmassa, toisen konsentraatiota kudoksessa ja mahdollisesti sen jälkeen tulevien lokeroiden merkkiaineen metabolisoitumista tai sitoutumista [7, s. 348]. Lokeromallin parametreina ovat kineettiset parametrit, jotka kuvaavat merkkiaineen kulkeutumisnopeutta lokeroiden väleillä. PET-laitteiston muutaman kuutiomillimetrin luokkaa oleva erotuskyky asettaa rajan sille kuinka tiheälle tilavuusalkiojaolle mallinnus on mielekästä toteuttaa. Tilavuusalkiosta käytetään nimitystä vokseli.

Lokeromallin yleisessä muodossa merkkiaineen konsentraatioiden muutoksia lokeroissa kuvataan differentiaaliyhtälöillä [3]

$$\left\{ \frac{\partial C_s(t, K_1^s, \ldots, K_N^s)}{\partial t} = \sum_{l=1}^{N} K_l^s C_l(t) \;\middle|\; s = 1, ..., N \right\}, \tag{2.4}$$

missä, C_s ja C_l ovat lokeroiden konsentraatioita ja K_l^s:t ovat kineettisiä parametreja. Yleisen nimeämiskäytännön mukaisesti kineettiset parametrit $K_2, \ldots, K_N$ on kirjoitettu tässä työssä pienellä kirjaimella muotoon $k_2, \ldots, k_N$. Lokeromalli on havainnollistettu kuvassa 2.2.

Tässä diplomityössä on tehty yleisesti käytetyt oletukset, että kineettiset parametrit eivät merkittävästi muutu ajan suhteen PET-mittausten aikana, C_1 on mallin



Kuva 2.2: *Lokeromalli, jossa N lokeroa*

sisääntulo eli plasman tunnettu aika-aktiivisuuskäyrä ja ajanhetkellä $t = 0$ kaikkien lokeroiden konsentraatio on nolla. Työn yhteydessä käytetään lokeromallin yleisestä muodosta johdettuja nk. 2K- ja 3K-lokeromalleja. 2K-lokeromallissa on kaksi lokeroa, C_1 ja C_2, ja kaksi kineettistä parametria, K_1 ja k_2. Lokeron C_2 konsentraation derivaatta ajan suhteen on 2K-mallissa muotoa

$$\frac{\partial C_2}{\partial t} = K_1 C_1 - k_2 C_2, \tag{2.5}$$

ja konsentraatioksi saadaan

$$C_2(t) = K_1 \int_0^t \exp(k_2(x-t)) C_1(x) dx. \tag{2.6}$$

3K-malliksi puolestaan kutsutaan lokeromallia, jossa on kolme lokeroa, C_1, C_2 ja C_3, ja kolme nollasta poikkeavaa kineettistä parametria, K_1, k_2 ja k_3. Tässä työssä oletetaan, että parametri $k_4 = 0$. 3K-mallissa lokeroiden konsentraatioiden derivaatat ovat

$$\begin{cases} \frac{\partial C_2}{\partial t} = K_1 C_1 - k_2 C_2 - k_3 C_2 \\ \frac{\partial C_3}{\partial t} = k_3 C_2 \end{cases} \tag{2.7}$$

ja konsentraatiot

$$\begin{cases} C_2(t) = K_1 \int_0^t \exp((k_2 + k_3)(x-t)) C_1(x) dx \\ C_3(t) = k_3 \int_0^t C_2(x) dx \end{cases} \tag{2.8}$$

3. VIRHEKORJATUN HETEROGEENISUUSESTIMAATIN LASKENTA PET-MITTAUKSISTA

Tämän diplomityön yhteydessä kudoksen heterogeenisuuden mittarina käytetään lokeromallin kineettisten parametrien keskihajontaa tarkasteltavalla alueella ja valitulla resoluutiolla. Heterogeenisuuden määrittämiseksi käytetään tavanomaisesta poikkeavaa, mallintamiseen perustuvaa lähestymistapaa [3], jossa keskihajonnat estimoidaan suoraan sinogrammiaikasarjasta suurimman uskottavuuden estimointia hyödyntäen. Menetelmä sisältää oleellisena osana myös vaiheen, jossa pyritään minimoimaan estimaatteihin muodostuvan esimointivirheen vaikutus. Tässä luvussa kuvaillaan tämän menetelmän toiminta.

3.1 Mittausmalli

Koska pyrkimyksenä on mallintaa PET-mittauksista saatua sinogrammimuotoista dataa niin hyvän mallin edellytyksenä on pystyä mallintamaan mittausten taustalla olevan prosessin kaksi päävaihetta. Ensimmäisenä vaiheena on merkkiaineen fysiologinen käyttäytymisen mallintaminen kohteessa ja sen seurauksena muodostuva virtuaalinen aktiivisuuskonsentraatiokuva Λ. Tämä vaihe mallinnetaan luvussa 2.2 kuvatulla lokeromallilla. Koska rajoitumme tämän työn yhteydessä yhteen todellisuudessa kolmiulotteisen kohteen viipaleeseen niin Λ on kiinteädimensioinen matriisi, jonka vokseliksi i ajanhetkellä t saadaan [3]

$$\Lambda_i(t) = \int_{\triangle t} \Big((1 - F_i)(\sum_{s=2}^{N} C_{si}(t)) + C_1(t) F_i \Big) \, \mathrm{e}^{-\lambda t} dt, \tag{3.1}$$

missä F_i on plasman tilavuusosuus kudoksessa vokselissa i, λ on käytetyn merkkiaineen hajoamisvakio, $\triangle t$ on aikaikkuna t:n ympärillä ja lokeroiden konsentraatioiden $C_{si}(t)$ määritelmät riippuvat käytettävästä lokeromallista.

Toisena mallinnettavana vaiheena on Λ:n kuvantamisen mallintaminen, mikä toteutetaan luvussa 2.1 esitellyn Radon-muunnoksen diskreetillä tapauksella. Mikä tahansa PET-kuvantamisjärjestelmä voidaan mallintaa Radon-muunnoksella yksilöllisen systeemimatriisin W avulla siten, että yksittäinen ideaalisen sinogrammin bin $S_j(t)$ saadaan systeemimatriisin alkioiden ja virtuaalisen kuvan Λ vokseleiden

lineaarikombinaatioina

$$S_j(t) = \sum_{i=1} W_{j_i} \Lambda_i(t), \tag{3.2}$$

missä i käy läpi kaikki virtuaalisen kuvan pikselien ja j_i vastaavien systeemimatriisin alkioiden indeksit. Koko ideaalinen sinogrammiaikasarja S voidaan esittää matriisitulona

$$S = W\Lambda, \tag{3.3}$$

missä W on replikoitu ja Λ vektoroitu sisältäen kaikki $\Lambda(t)$:t.

Tällainen ideaalinen aktiivisuuden mittaaminen ei kuitenkaan ole realistista, sillä radioaktiivisen hajoamisen satunnaisluonne aiheuttaa vaihtelua mitatuissa hajoamistapahtumamäärissä. Vaihtelun vaikutusta voidaan mallintaa Poissonin jakauman avulla. Tämän lisäksi PET-laitteiston detektorit ja muut virhelähteet aiheuttavat mittaustulokseen virhettä, jota mallinnetaan normaalijakautuneella satunnaismuuttujalla. Nämä virhelähteet mallintavan, nk. mitatun sinogrammiaikasarjan M yksittäiseksi biniksi j saadaan [3]

$$M_j \sim \mathrm{P}(S_j) + \mathrm{N}(0, \sigma^2), \tag{3.4}$$

missä P on Poissonin satunnaismuuttuja, jonka odotusarvo on ideaalinen sinogrammin binin arvo S_j ja N normaalijakautunut satunnaismuuttuja, jonka odotusarvo on 0 ja keskihajonta σ. Normaalijakauman keskihajonta σ oletetaan samaksi jokaiselle sinogrammin binille, mutta jokaisella binillä on oma realisoitunut virheensä. Normaalijakautunut virhekomponentti oletetaan pieneksi Poisson-virheeseen verrattuna.

3.2 Suurimman uskottavuuden estimointi

Seuraavana ongelmana on sellaisten mittausmallin (3.4) parametrien määrittäminen, jotka parhaiten selittävät PET-mittausdatan, kun sen oletetaan muodostuneen mittausmallin kuvailemalla prosessilla. Tällaisten parametrien määrittämiseen valittiin menetelmäksi suurimman uskottavuuden estimointi (maximum likelihood estimation) eli ML-estimointi. Suurimman uskottavuuden estimointi on tilastollinen menetelmä, jota käytetään tilastollisen mallin sovittamiseen mittausaineistolle $x = [x_1, x_2, \ldots, x_n]$ ja mallin parametriestimaattien $\hat{\theta}$ löytämiseen.

Oletetaan, että mittausaineisto koostuu riippumattomista, identtisesti jakautuneista näytteistä tuntemattomasta tiheysfunktiosta $p_0(\cdot)$. Mikäli myös oletetaan, että p_0 kuuluu tiettyyn jakaumaperheeseen niin voidaan pyrkiä määrittämään tällaisen jakauman p määritteleville parametreille estimaatti $\hat{\theta}$, joka on mahdollisimman

lähellä tuntemattomia, todellisia parametreja θ^{tosi}, kun $p(\cdot \mid \theta^{\text{tosi}}) = p_0$. Yhteiseksi tiheysfunktioksi aineistolle x saadaan

$$p(x \mid \theta) = p(x_1 \mid \theta)p(x_2 \mid \theta) \cdots p(x_n \mid \theta). \tag{3.5}$$

Mikäli aineisto oletetaan kiinnitetyksi ja parametrit θ muuttujiksi niin tiheysfunktiota kutsutaan myös nimellä uskottavuusfunktio. Uskottavuusfunktion maksimoivaa parametrivektoria θ^{mle} kutsutaan suurimman uskottavuuden estimaatiksi eli ML-estimaatiksi

$$\theta^{\text{mle}} = \arg\max_{\theta} p(x \mid \theta). \tag{3.6}$$

Uskottavuusfunktiota sovelluskohteeseen muodostettaessa kaavan (2.4) kineettiset parametrit K_l^s ja (3.4):n normaalijakautuneen virheen poikkeama σ oletetaan satunnaismuuttujiksi. Näistä muuttujista koostetaan parametrivektori $\theta = (K_1^1, \ldots, K_1^s, k_2^1, \ldots, k_N^s, \sigma)$. Mitatun sinogrammiaikasarjan M ja parametrivektorin θ välinen uskottavuusfunktio muodostetaan mittausmallista (3.4) Poisson-jakauman pistetodennäköisyysfunktion ja normaalijakauman tiheysfunktion konvoluutiona

$$p(M \mid \theta) = \prod_{j=1}^{m} \sum_{k=0}^{\infty} \left(\frac{(S_j)^k \exp(-S_j - \frac{1}{2}(M_j - k)^2/\sigma^2)}{k!\sqrt{2\pi\sigma^2}} \right), \tag{3.7}$$

missä S on parametreja θ käyttäen mallin kaavoilla (2.4) ja (3.3) muodostettu kohinaton sinogrammiaikasarja ja alaindeksi j käy läpi kaikki sinogrammiaikasarjan S binit, joita on yhteensä m.

On todistettu [4], että ML-estimaatti θ^{mle} on olemassa, kun mittauksia mallinnetaan (3.4):n mukaisesti Poisson-kohinaa käyttäen. Tästä kuitenkin seuraa, että uskottavuusfunktiota (3.7) on äärettömän summan ja kertoman takia vaikea estimoida numeerisesti. Ongelman yksinkertaistamiseksi approksimoimme Poisson-jakaumaa normaalijakaumalla. Tämä on mahdollista, sillä $\mathrm{P}(S_j) \approx \mathrm{N}(S_j, S_j)$ [3]. Nyt M_j on muotoa

$$M_j \sim \mathrm{N}(S_j, S_j) + \mathrm{N}(0, \sigma^2) = \mathrm{N}(S_j, S_j + \sigma^2). \tag{3.8}$$

Approksimoinnin seurauksena uskottavuusfunktio yksinkertaistuu moniulotteisen normaalijakauman tiheysfunktioksi

$$p(M \mid \theta) = \frac{1}{(2\pi)^{m/2}\sqrt{\det(\Sigma)}} \exp\big((M - S)^{\mathrm{T}}\, \Sigma^{-1}(M - S)\big), \tag{3.9}$$

missä Σ on diagonaalimatriisi, jonka diagonaalin elementit ovat $(S + \sigma^2)$ ja m on yhteenlaskettu binien määrä kaikissa sinogrammiaikasarjan aikaikkunoissa ja projektioissa.

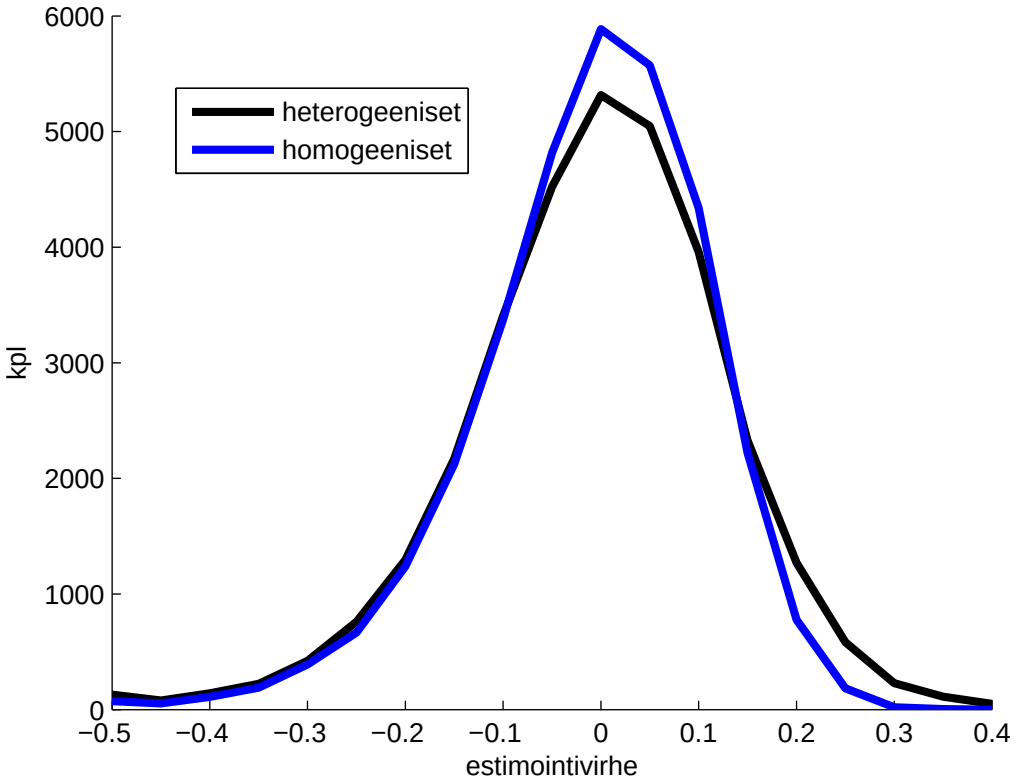
On hyvä huomata, että minkäänlaista regularisointia ei käytetä, joten saavutetaan aidot, teorian mukaiset estimointivirheen sisältävät ML-estimaatit. Tämä on tärkeää, sillä nykyiset regularisointia käyttävät menetelmät tuhoavat heterogeenisuusinformaatiota pyrkiessään minimoimaan kohinan aiheuttamaa variaatiota. Toisaalta ML-estimaatissa oleva estimointivirhe on niin suurta, ettei sitä voida suoraan käyttää heterogeenisuuden määräämiseen. Varsinainen ML-estimointi, eli ML-estimaatin määrittäminen, suoritetaan numeerisia optimointimenetelmiä uskottavuusfunktiolle käyttäen, koska analyyttistä ratkaisua ei ole saatavilla. Optimoinnissa voidaan käyttää perusmuotoisen uskottavuusfunktion $p(M \mid \theta)$ sijasta negatiivista ja logaritmista uskottavuusfunktiota $-\log(p(M \mid \theta))$, sillä logaritmi on jatkuva ja aidosti kasvava funktio. Logaritmin ansiosta uskottavuusfunktio on laskennallisesti yksinkertaisempi kertolaskujen korvautuessa yhteenlaskuilla ja tämän seurauksena myös tietokoneella käytettävissä olevan lukuesityksen laajuus riittää huomattavasti paremmin uskottavuusfunktion arvojen esittämiseen.

3.3 Toistokoemenetelmä estimointivirheen korjaamiseksi

Suurimman uskottavuuden estimointi valittiin parametrien θ estimointimenetelmäksi, sillä ML-estimaatin tilastollisia ominaisuuksia voidaan hyödyntää sinogrammidatan kohinaisuudesta aiheutuvan estimointivirheen korjaamiseksi alueellisista heterogeenisuusestimaateista. Alueella tarkoitetaan todellisesta kuvattavasta kohteesta määriteltyä osaa, jolle virhekorjattu heterogeenisuusestimaatti on tavoitteena määrittää.

ML-estimaatti θ^{mle} on kaikille parametreille asymptoottisesti harhaton ja noudattaa normaalijakaumaa todellisten parametriarvojen θ^{tosi} ympärillä [8]. Tästä johtuen alueelliset kineettisten parametrien histogrammit ovat todellisten kineettisten parametrien histogrammien ja virherealisaatioiden histogrammien konvoluutioita. Mikäli estimointivirheen histogrammi tunnetaan on sen vaikutus mahdollista korjata ja palauttaa todellinen parametrien alueellinen histogrammi tämän muodosta riippumatta. Yksittäisistä parametreista estimointivirhettä on kuitenkin mahdotonta korjata, sillä yksittäistä realisoitunutta virhettä ei voida saada selville.

Estimointivirheen estimoimiseksi on kehitetty Monte Carlo -lähestymistapaa hyödyntävä menetelmä [3], jossa ML-estimointi toistetaan useasti virtuaalisella, alueellisesti homogeenisella kohteella, joka imitoi aluejaollaan todellisen kohteen fysiologiaa. Tämän virtuaalisen kohteen alueille r määritetään kineettisten parametrien θ_r^{vir} tosiarvoiksi todelliselle sinogrammidatalle suoritetun ML-estimoinnin tuloksista vastaavilta alueilta lasketut parametrien keskiarvot, jotka harhattomuudesta johtuen vastaavat likimain odotusarvoja. Tämä on tarpeen, sillä ML-estimointiteorian [8] mukaan muodostuvan estimointivirheen suuruus on vahvasti riippuvainen parametrien tosiarvojen odotusarvosta, mutta lähestulokoon riippumaton näiden hajon-



Kuva 3.1: *20:lle heterogeeniselle ja homogeeniselle simuloidulle datalle suoritetuista ML-estimoinneista määritetyt k_2:n estimointivirheen histogrammit kohteen suurimmalta alueelta, kun käytettävänä lokeromallina on ollut 2K, fantomin dimensiot ovat olleet* 64×64 *ja datan generoinnissa käytetyt parametriarvot on ilmoitettu taulukon 5.1 kahdella ensimmäisellä rivillä.*

nasta.

Kun tällaisella alueittain homogeenisella datalla suoritetaan ML-estimointi, niin kaikki ML-estimaatissa $\theta_r^{\mathrm{mle|vir}}$ esiintyvä parametrien varianssi on estimointivirheestä aiheutuvaa ja antaa estimaatin estimointivirheen histogrammista. Mikäli alueella r sijaitsee pieni määrä vokseleita, on hyvän estimaatin saamiseksi toistettava ML-estimointi usealla eri kohinarealisaatiolla homogeenisesta datasta. Tämä johtuu siitä, että estimointivirhe yksittäisessä kineettisessä parametrissa kaikissa toistoissa antaa vain yksittäisen realisaation tuntemattomasta alueen virhehistogrammista ja näitä realisaatioita on oltava riittävän suuri määrä luotettavan estimaatin määrittämiseksi.

Tässä aliluvussa kuvattua toistokoemenetelmää hyvin pienten alueiden kanssa käytettäessä on hyvä tiedostaa muutaman, laajempien alueiden yhteydessä epäoleellisen virhelähteen vaikutus. Ensinnäkin todelliselle mittausdatalle suoritetusta ML-estimoinnista saatavat alueen kineettisten parametrien histogrammit ovat jos-

sain määrin ongelmallisia, sillä ne sisältävät vain hyvin vähän alkioita ja yksittäisten virherealisaatioiden vaikutus histogrammiin voi kasvaa suureksi. Toiseksi, näistä histogrammeista ei myöskään saada luotettavaa estimaattia alueen kineettisten parametrien odotusarvoille, joita käyttäen homogeeninen data muodostetaan. Tästä aiheutuu homogeeniselle datalle suoritetuista toistoista saatuun virhehistogrammin estimaattiin virhettä, joka on riippumatonta suoritettujen toistojen määrästä.

Kuvaajassa 3.1 on havainnollistettu eräät realisoituneen estimointivirheen histogrammit sekä usealla alueellisesti heterogeenisella, todellista mittauskohdetta simuloivalla datalla että edellisistä muodostetulla, alueellisesti homogeenisella datalla. Havaitaan, että molemmissa tapauksissa realisoituneen estimointivirheen histogrammit ovat teorian mukaisesti hyvin samankaltaiset ja siten estimointivirheen vaikutusta on mahdollista korjata. Kuvaaja on muodostettu erään tämän diplomityön teon yhteydessä suoritetun testin tuloksiin pohjautuen. Testien toteutukseen liittyvät yksityiskohdat on esitelty luvussa 5.1.

4. VERTAILTAVAT OPTIMOINTIMENETELMÄT

Optimoinnilla tarkoitetaan parhaimman valinnan tekemistä joukosta mahdollisia vaihtoehtoja. Usein tämä tarkoittaa kohdefunktion $f(\mathbf{X})$ globaalin- tai lokaalin minimi- tai maksimikohdan etsimistä, kun $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ on nk. parametrivektori, joka sisältää optimoitavat muuttujat. Globaalissa minimikohdassa $\mathbf{X_{glob}}$ pätee $f(\mathbf{X_{glob}}) \leq f(\mathbf{Y}), \forall\ \mathbf{Y}$. Minimikohta voidaan tunnistaa, mikäli kohdefunktio $f(\mathbf{X})$ on derivoituva, sillä voidaan laskea gradientti, joka saa minimikohdissa arvon nolla. Gradientti saa arvon nolla myös satula- ja maksimikohdissa, mutta nämä voidaan erottaa minimikohdista lähiympäristöä tarkastelemalla. Globaalin minimikohdan tunnistaminen muista minimikohdista on sen sijaan tavallisesti vaikeaa tai mahdotonta. [9]

Käytännön optimointiongelmissa parametrivektorin muuttujille asetetaan toisinaan rajoitteita. Esimerkiksi auton moottorin tehoa optimoitaessa voidaan asettaa sallittu arvoväli moottorin massalle, joka on yksittäinen parametrivektorin muuttuja. Rajoitteiden asettamisen etuna on se, että optimointiavaruus pienenee rajoittamattomaan tapaukseen verrattuna. Toisaalta rajoittamisen seurauksena voi olla, että globaali minimi- tai maksimikohta sijaitsee rajojen ulkopuolella ja jää löytämättä.

Optimointiongelma, eli kohdefunktion minimin löytäminen, voidaan ratkaista joko analyyttisesti tai numeerisesti. Analyyttisesti tämä voidaan tehdä esimerkiksi evaluoimalla kohdefunktio kaikissa derivaatan nollakohdissa ja valitsemalla näistä ääriarvokohdista se, jossa funktion evaluointi tuotti pienimmän arvon. Kohdefunktio voi olla niin monimutkainen, ettei derivaatan nollakohtia ole mahdollista ratkaista tai derivaatta ei edes ole saatavilla ja joudutaan tyytymään numeeriseen optimointiin. Numeerinen optimointi toteutetaan algoritmeilla, jotka sisältävät jonkinlaisen iteratiivisen logiikan kohdefunktion toivottujen ääriarvojen löytämiseksi.

Tässä diplomityössä pyritään minimoimaan kohdefunktiota, joka on negatiivinen ja logaritminen uskottavuusfunktio (3.9), jonka globaalissa minimikohdassa sijaitsee suurimman uskottavuuden estimaatti. Kohdefunktio on suuridimensioinen ja epälineaarinen, mikä tuo haasteita numeeriseen optimointiin. Parametrivektorin muuttujien arvoja ei myöskään rajoiteta, sillä suurimman uskottavuuden uskottavuusfunktion käyttäminen parametrivektorin optimoimiseksi tuo parametrien estimaatteihin

estimointivirhettä, jonka vaikutus voi olla suuri. Estimointivirheestä johtuen parametrit voivat esimerkiksi saada negatiivisia arvoja, joille ei ole fysiologisesti mielekästä tulkintaa. Kohdefunktio on toisaalta derivoituva, mikä mahdollistaa myös gradienttia hyödyntävien optimointimenetelmien soveltamisen.

4.1 Nollannen asteen menetelmät

Nollannen asteen menetelmillä tarkoitetaan optimointimenetelmiä, jotka käyttävät optimointiprosessissaan vain kohdefunktion evaluoimista. Nollannen asteen menetelmien käyttö voi olla hyödyllistä mikäli kohdefunktio ei ole derivoituva, sillä tällöin gradienttia ole välttämättä mahdollista tai järkevää hyödyntää. Toinen usein hyödyllinen käyttökohde on multimodaalisten kohdefunktioiden optimointi. Nollannen asteen menetelmille on tyypillistä, että ne tarvitsevat hyvin suuren määrän kohdefunktion evaluointeja yksinkertaistenkin ongelmien ratkaisemiseksi. [9, s.94]

Alkeellisimmat nollannen asteen menetelmät perustuvat kohdefunktion kattavaan evaluoimiseen koko optimointiavaruuden laajuudella. Evaluoinnit suoritetaan menetelmästä riippuen joko jotain ennaltamäärättyä järjestelmällistä tai satunnaisuuteen perustuvaa etsintäkuviota noudattaen. Tällaiset optimointimenetelmät ovat useimmiten käyttökelpoisia vain mikäli optimoitavia parametreja on vähän ja ne saavat vain diskreettejä arvoja rajoitetulla arvovälillä. Kehittyneemmät nollannen asteen menetelmät ovat adaptiivisia eli osaavat muuttaa toimintaansa aiempien evaluointien perusteella. Kaikki tässä luvussa esiteltävät menetelmät ovat adaptiivisia, mutta niistä jokaisessa adaptiivisuus on toteutettu erilaisella logiikalla.

Monet kehittyneemmät menetelmät ovat populaatiopohjaisia, mikä tarkoittaa sitä, että ne eivät optimoi vain yhtä parametrivektoria, vaan parametrivektorijoukkoa. Populaatiopohjaisuuden seurauksena käytettävissä oleva optimointi-iteraatiomäärä jakautuu populaation jäsenten kesken, mutta toisaalta optimointia edistää mm. se, että hyvän alkuarvauksen todennäköisyys kasvaa, kun jokaisella populaation jäsenellä on oma alkuarvauksensa. Populaatiopohjaisuutta hyödynnetään eri menetelmissä eri tavoin. Esimerkiksi niin sanotut evolutiiviset algoritmit (evolutionary algorithm, EA) hyödyntävät sitä biologiseen evoluutioon perustuvilla mekanismeilla: mutaatiolla, rekombinaatiolla ja valinnalla. Tämä tarkoittaa sitä, että uusia parametrivektorikandidaatteja muodostetaan nykyisten populaation jäsenten pohjalta näitä mekanismeja hyödyntäen.

Tässä luvussa esiteltävät menetelmät ovat menestyneet hyvin optimointimenetelmien välisissä vertailuissa ja soveltuvat erilaisten ongelmatyyppien ratkaisemiseen [10][11][12]. Menetelmät on valittu projektin yhteistyökumppanin, dosentti Ferrante Nerin, ehdottamina.

4.1.1 Monilentoratahaku

Monilentoratahaku (multiple trajectory search, MTS) [13] on populaatiopohjainen optimointimenetelmä, joka sisältää kolme erillistä lähistöhakumenetelmää (local search, LS), jotka myös ovat itsessään optimointialgoritmeja. MTS:n pseudokoodi on esitelty algoritmissa 3.1. Joka kerta kun populaation jäsen tulee optimointivuoroon, sille kokeillaan tietty määrä ($TestiKierrokset$) iteraatioita jokaisella kolmesta lähistöhakumenetelmästä. Näistä jokainen antaa jäsenelle pisteitä sen mukaan miten hyvin ne toimivat ja eniten pisteitä saanut menetelmä otetaan käyttöön varsinaiseen lähistöhakuun, jossa suoritetaan ($HakuKierrokset$) lisää iteraatioita. Edellä kuvatun pisteytysmenetelmän tarkoituksena on saada valittua parhaiten populaation jäsenen optimointiavaruuden lähistön topografiaan sopiva menetelmä. Pisteytys ei perustu kohdefunktion arvon minimoitumismääriin, joten valituksi voi tulla lähistöhakumenetelmä, joka ei ole saavuttanut suurinta kohdefunktion minimoitumista.

Populaatio alustetaan nk. simuloitua ortogonaalirakennetta (simulated orthogonal array, SOA) ja parametrikohtaista ylärajavektoria $YRAJA$ ja alarajavektoria $ARAJA$ hyödyntäen. Kun populaatiokoko on M niin SOA-matriisin riviksi d saadaan

$$SOA_d = randperm(0, 1, ..., M - 1), \tag{4.1}$$

kun käydään läpi kaikki optimointiavaruuden dimensiot d ja $randperm$ on funktio, joka muodostaa satunnaisen permutaation sille annetusta syötteestä. Nyt populaation jäsen x_i saadaan alustettua kaavalla

$$x_i[d] = ARAJA[d] + (YRAJA[d] - ARAJA[d])\frac{SOA_d[i]}{(M-1)}. \tag{4.2}$$

Tällaisen alustusmenettelyn tarkoituksena on, että jokaisesta parametrivektorin dimensiosta löytyy populaatiossa arvot tasavälein alarajalta ylärajalle. On hyvä huomata, että ala- ja ylärajavektorit eivät rajoita optimointiavaruutta, vaan ovat vain vaikuttamassa alustusvälien laajuuteen ja lähtistöhakumenetelmien puolitushaussa käyttämien populaation jäsenkohtaisten hakurajojen ($Hakurajat$) pituuksiin.

Populaation jäsenistä osa on ensimmäistä iteraatiota lukuunottamatta etu- ja osa taka-alalla siitä riippuen kuinka hyvän pisteytyksen ne ovat saavat edellisessä lähistöhaussa. Vain etualalla olevia jäseniä optimoidaan seuraavalla monilentoratahaun kierroksella, joka käy kaikki populaation jäsenet läpi. Tällaisen kierroksen jälkeen alkuperäisen pseudokoodin [13] mukaan tulee vielä suorittaa tietty määrä ($ParhaanKierrokset$) optimointi-iteraatioita LS1-lähistöhakumenetelmällä populaation parhaalle jäsenelle. Myöhemmässä julkaisussa [14] tällainen parhaan jäsenen optimointi on jätetty pois, mutta tämän työn yhteydessä se on käytössä.

```
1.  Generoi populaatio X, jossa M yksilöä
2.  for i = 1 to M
3.      Salli[i] = TOSI
4.      Parannus[i] = TOSI
5.      Hakurajat[i] = (YRAJA − ARAJA)/2
6.  end
7.  While lopetusehto ei täyty
8.      for i = 1 to M
9.          if Salli[i] == TOSI
10.             Pisteet[i] = 0
11.             for j = 1 to TestiKierrokset
12.                 LS1_pisteet = LS1_pisteet + LS1(X_i)
13.                 LS2_pisteet = LS2_pisteet + LS2(X_i)
14.                 LS3_pisteet = LS3_pisteet + LS3(X_i)
15.             end
16.             Pisteiden perusteella valitaan paras LS
17.             for j = 1 to HakuKierrokset
18.                 Pisteet[i] = Pisteet[i] + LS_paras(X_i)
19.             end
20.         end
21.     end
22.     for i = 1 to ParhaanKierrokset
23.         LS1(X_paras)
24.     end
25.     for i = 1 to M
26.         if Pisteet[i] parhaan #etuala:n joukossa
27.             Salli[i] = TOSI
28.         else
29.             Salli[i] = EPÄTOSI
30.         end
31.     end
32. end
```

Algoritmi 4.1. *MTS:n pseudokoodi.*

Lähistöhakumenetelmistä ensimmäinen, LS1, käy optimoitavan parametrivektorin dimensioita läpi järjestyksessä yksi kerrallaan eräänlaista puolitushakua tarkasteluvälillä tehden ja tarvittaessa tarkasteluväliä siirtäen. LS2 puolestaan tekee yh-

dessä iteraatiossa samankaltaisen puolitushaun joukolle parametreja, ja näin ollen saa suoritetuksi enemmän operaatioita yhtä kohdefunktion evaluointia kohden. Toisaalta kohdefunktion arvon mahdollinen parantuminen ei kerro mitään siitä mitkä yksittäiset parametrit tekivät hyödyllisen siirtymän ja mitkä eivät. LS1:n tavoin LS3 käy parametrivektorin järjestyksessä läpi parametri kerrallaan, mutta pyrkii löytämään parannuksen tekemällä parametrille pieniä muutoksia lisäämällä ja vähentämällä siihen pieniä vakiolukuja ja tekemällä näiden muutosten tulosten pohjalta heuristisen arvauksen uudeksi parametrin arvoksi. Kaikkien edellä mainittujen lähistöhakumenetelmien pseudokoodit löytyvät liitteestä A.

4.1.2 Itseohjautuva differentiaalinen evoluutio

Differentiaalinen evoluutio (differential evolution, DE) on populaatiopohjainen algoritmi, joka jäljittelee evoluutioprosessia ja kuuluu siten evolutiivisiin algoritmeihin. DE jakautuu kolmeen vaiheeseen: mutaatioon, risteytykseen ja valintaan, jotka suoritetaan tässä järjestyksessä. Populaatio X alustetaan tasajakautuneista satunnaisista pisteistä optimointiavaruuden kaikkien dimensioiden suhteen. Populaatiokoon n suositus on kymmenen kertaa optimoitavien parametrien määrä dim. [15]

Mutaatiota varten arvotaan satunnaisesti kolme populaation jäsenen indeksiä t,r ja s, joista yksikään ei saa olla käsiteltävän jäsenen indeksi i. Mutaatiossa

$$x_{jm} = x_t + F(x_r - x_s), \tag{4.3}$$

x_{jm} kuvaa mutaatiossa muodostettua jälkeläistä ja F skaalauskerrointa, joka kontrolloi eksploratiivisen vektorin $(x_r - x_s)$ pituutta [11]. Binaarisessa risteytyksessä

$$x_j[d] = \begin{cases} x_{jm}[d] & \text{if } rand \leq CR \\ x_i[d] & \text{else} \end{cases} \tag{4.4}$$

käydään läpi kaikki dimensiot $d = 1$ to dim ja muodostetaan jälkeläinen x_j käsiteltävän jäsenen x_i ja mutaation tuottaman jälkeläisen x_{jm} alkioista risteytysherkkyyden (crossover rate, CR) painottamana. Luku $rand$ generoidaan satunnaisesti väliltä $[0, 1]$ jokaiselle dimensiolle d. DE:n käyttämä ahne valinta (4.5) korvaa populaation jäsenen jälkeläisellä, vain mikäli tämä tuottaa parannuksen kohdefunktion arvoon.

$$x_i = x_j \quad \text{if } f(x_j) < f(x_i) \tag{4.5}$$

Perusmuotoisen DE:n ongelma on, että skaalauskerroin F ja risteytysherkkyys CR ovat ennaltamäärättäviä vakioita, joiden optimiarvot riippuvat kohdefunktiosta. Skaalauskertoimen ja risteytysherkkyyden valinta voi vaikuttaa merkittävästi optimoinnin nopeuteen ja toteutetaan usein kokeilemalla. Eräs tehokkaaksi todetuista [11] DE-varianteista on algoritmissa 4.2 esitelty itseohjautuva differentiaalinen evo-

$\mathbf{x}_1$	F_1	CR_1
$\mathbf{x}_2$	F_2	CR_2
$\vdots$		
$\mathbf{x}_n$	F_n	CR_n

Kuva 4.1: *jDE:n populaatio X, jonka koko on n.*

luutio, jDE [16], joka tuo vain pieniä lisäyksiä perusmuotoiseen DE:hen. jDE:ssä F ja CR ovat jäsenkohtaisia ja muuttuvat optimoinnin aikana adaptiivisesti olemalla mukana evoluutioprosessissa parametrivektorin loppuun liitettyinä kuvan 4.1 mukaisesti.

Algoritmissa 4.2 esiintyvät jDE:n käyttämät vakiot ovat: $\tau_1 = 0.1$ ja $\tau_2 = 0.1$, jotka määrittävät riveillä 4 ja 6 kuinka usein F:lle ja CR:lle generoidaan uudet arvot, ja $F_l = 0.1$ ja $F_u = 0.9$, jotka määrittävät generoitavan F:n ala- ja ylärajan. Jokainen algortmissa esiintyvä *rand* edustaa itsenäisesti generoitua satunnaislukua väliltä $[0, 1]$. Mikäli uusi F tai CR generoidaan niin ne pääsevät vaikuttamaan mutaatio- ja risteytysprosessiin jo kyseisellä iteraatiolla.

Yksinkertaisuutensa ja tehokkuutensa ansiosta DE ja sen variantit ovat laajalti insinöörityössä käytettyjä optimointimenetelmiä. On havaittu, että varianteista erityisesti jDE on menestyksekäs monenlaisisissa eri ongelmatyypeissä, mutta koska parhaan menetelmän valinta on hyvin ongelmakohtaista ei mitään yksittäistä varianttia voida nostaa parhaaksi. [11]

```
1.  Generoi populaatio X, jossa n yksilöä
2.  While lopetusehto ei täyty
3.      for i = 1 to n
4.
            F_i = { F_l + F_u · rand_1   if  rand_2 < τ_1
                  {               F_i   else

5.          Standardi DE:n mutaatio, missä F = F_i
            ja mutatoidaan myös osapuolten CR ja F
6.
            CR_i = { rand_3   if  rand_4 < τ_2
                   {  CR_i   else

7.          Standardi DE:n risteytys, missä CR = CR_i
            ja risteytetään myös osapuolten CR ja F
8.          Standardi DE:n valinta
9.      end
10. end
```

Algoritmi 4.2. *jDE:n pseudokoodi.*

4.1.3 Kattavasti oppiva hiukkasparvioptimointi

Hiukkasparvioptimointi (particle swarm optimization, PSO) on populaatiopohjainen algoritmi, joka matkii eliöiden parvikäyttäytymistä ja kuuluu siten evolutiivisiin algoritmeihin. PSO:ssa populaation jäsenet, hiukkaset, ovat pisteitä optimointiavaruudessa ja jokaisella niistä on nopeus ja muisti parhaimmasta sijainnistaan, jossa kohdefunktio on saanut pienimmän arvonsa. Lisäksi aina on tiedossa paras koko populaation löytämä sijainti eli globaalisti paras sijainti. Hiukkasille määritetään jokaisella PSO:n iteraatiolla uusi nopeus, joka lasketaan kyseisen hiukkasen parasta löytämää sijaintia, globaalisti parasta sijaintia ja aiempaa nopeutta hyödyntäen. Hiukkasen uusi sijainti saadaan summaamalla nopeus vanhaan sijaintiin.

Tämän perusmuotoisen PSO-algoritmin [17] suurin ongelma on lokaaleihin minimeihin juuttuminen, sillä kaikki hiukkaset hakeutuvat aina koko populaation löytämää parasta sijaintia kohti. PSO:n pohjalta on kehitetty lukuisia variantteja, jotka sopivat globaalin minimikohdan löytämiseen paremmin. Eräs näistä varianteista on kattavasti oppiva hiukkasparvioptimointi (comprehensive learning particle swarm optimizer, CLPSO) [18], jossa merkittävimpänä muutoksena perusmuotoiseen PSO-

algoritmiin hiukkaset hakeutuvat kaikkien hiukkasten löytämiä parhaita sijainteja kohti.

Algoritmissa 3.3. on esitetty CLPSO:n pseudokoodi, joka alkaa populaation X, nopeuksien V, oppimistodennäköisyyksien Pc_i ja oppimiskohteiden f_i alustamisella. Populaatiokoon n suuruusluokaksi suositellaan kymmenen kertaa optimoitavien parametrien määrä dim. CLPSO on alkupopulaation sijaintien suhteen robusti, mutta mikäli optimoitavasta kohdefunktiosta ja sen parametreista ei ole ennakkokäsitystä kannattaa populaatio alustaa tasajakautuneesti optimointiavaruuden kaikkien dimensioiden suhteen mahdollisimman laajalla vaihteluvälillä. Vaihteluväliä voidaan kaventaa optimoinnin nopeuttamiseksi, mikäli on ennakkotietoa parametrien arvoskaalasta kuten tämän työn yhteydessä ja alkupopulaation yksittäisen hiukkasen yksittäiseksi dimensioksi saadaan

$$x_i[d] \sim U(b_{lo}, b_{up}), \tag{4.6}$$

kun käydään läpi kaikki dimensiot d ja b_{lo} on vaihteluvälin alaraja ja b_{up} yläraja. Näitä rajoja ei tule sekoittaa optimoinnin rajoittamiseksi. Yksittäisen hiukkasen i alkunopeus yksittäisen dimension suhteen alustetaan vastaavasti väliltä

$$v_i[d] \sim U(-|b_{up} - b_{lo}|, |b_{up} - b_{lo}|). \tag{4.7}$$

Jokaiselle partikkelille määritetään oppimistodennäköisyys

$$Pc_i = 0.05 + 0.45\ \frac{\exp(\frac{10(i-1)}{ps-1}) - 1}{\exp(10) - 1}, \tag{4.8}$$

jonka suuruus vaikuttaa siihen kuinka herkästi partikkeli ottaa mallia muiden partikkelien löytämistä parhaista ratkaisuista. Myöhemmissä tutkimuksissa [19] on havaittu, että oppimistodennäköisyyden adaptiivinen valitseminen toimii edellä esitettyä alkuperäistä, staattista määritelmää paremmin useimmissa tapauksissa, mutta tästä huolimatta tämän diplomityön yhteydessä valitsimme käytettäväksi alkuperäisen määritelmän.

Oppimiskohdevektorit f_i kertovat mistä toisesta hiukkasesta hiukkasen i yksittäinen dimensio ottaa mallia. Prosessi on kuvattuna yksityiskohtaisemmin pseudokoodin riveillä 6-17. Oppimiskohdevektorit määritetään aina uudelleen, mikäli hiukkanen ei ole virkistämisväliä m edustavan iteraatiomäärän jälkeen onnistunut löytämään uutta omakohtaista parasta sijaintia.

```
1.  Generoi populaatio X, alusta V, Pc ja f_i
2.  for k = 1 to maxGen
3.      w[k] = w_0 - ((w_0 - w_1)k) / maxGen
4.      for i = 1 to n
5.          if lippu[i] ≥ m
6.              for d = 1 to dim
7.                  if rand < Pc_i
8.                      Generoi kaksi satunnaista
                        partikkelin indeksiä f1 ja f2,
                        jotka eivät saa olla i
9.                      if f(hparas_f1) < f(hparas_f2)
10.                         f_i[d] = f1
11.                     else
12.                         f_i[d] = f2
13.                     end
14.                 else
15.                     f_i[d] = i
16.                 end
17.             end
18.             lippu[i] = 0
19.         end
20.         for d = 1 to dimensio
21.             V_i[d] = w · V_i[d] + c · rand · (paras_f_i[d][d] - X_i[d])
22.             V_i[d] = min(V_max[d], max(-V_max[d], V_i[d]))
23.             X_i[d] = X_i[d] + V_i[d]
24.         end
25.         if f(X_i) < f(hparas_i)
26.             hparas_i = X_i
27.             if f(X_i) < f(paras)
28.                 paras = X_i
29.             end
30.         else
31.             lippu[i] = lippu[i] + 1
32.         end
33.     end
34. end
```

Algoritmi 4.3. *CLPSO:n pseudokoodi.*

4.2 Ensimmäisen asteen menetelmät

Ensimmäisen asteen menetelmillä tarkoitetaan iteratiivisia optimointimenetelmiä, jotka hyödyntävät gradientti-informaatiota. Gradientin hyödyntämisen ansiosta ensimmäisen asteen menetelmät ovat usein nopeampia kuin nollannen asteen menetelmät minimikohdan löytämiseksi, mutta hakeutuvat gradientin osoittamaa minimiä kohti riippumatta siitä onko tämä lokaali vai globaali minimi. Useimpien ensimmäisen asteen menetelmien voidaan ajatella jakautuvan kahteen osa-ongelmaan: etsintäsuunnan valitsemiseen, jossa käytetään hyväksi gradientti-informaatiota, ja askelpituuden valitsemiseen, eli siihen kuinka paljon optimoitavien parametrien arvoja muutetaan etsintäsuuntaa kohti. Optimointi lopetetaan gradientin saadessa arvon nolla, mikä tapahtuu vain ääriarvo- ja satulakohdissa.

4.2.1 Jyrkimmän laskun menetelmä

Eräs yksinkertaisimmista ensimmäisen asteen menetelmistä on jyrkimmän laskun menetelmä (steepest descend method), joka on muotoa

$$x_{k+1} = x_k + \alpha_k d_k, \tag{4.9}$$

missä etsintäsuunta $d_k \in \mathbb{R}^n$ valitaan kohdefunktion negatiiviseksi gradientiksi kohdassa x_k

$$d_k = - \bigtriangledown f(x_k) \tag{4.10}$$

ja askelpituudeksi α_k valitaan se, joka minimoi kohdefunktion arvon etsintäsuunnassa gradientin suuntaisen siirtymän jälkeen

$$\alpha_k = \arg\min_{\alpha} f(x_k + \alpha_k d_k). \tag{4.11}$$

Negatiivisen gradientin suunta on lokaalisesti paras etsintäsuunta kohdefunktion arvon minimoimiseksi, mutta koko optimointiavaruutta ajatellen se ei ole välttämättä edes kovinkaan hyvä valinta. Vaihtoehtoisina etsintäsuuntina ovat nk. konjugaattigradienttisuunnat, joissa etsintäsuunta määritetään uuden gradienttisuunnan ja yhden tai useamman edellisen etsintäsuunnan painotettuna summana. [9]

Jyrkimmän laskun menetelmän jokaisella iteraatiolla kohdataan uusi optimointiongelma askelpituuden määrittämiseksi. Käytännössä optimaalisen tai lähellä optimaalista olevan askelpituuden etsintä on työlästä, koska sen löytämiseksi vaaditaan useita iteraatioita ja kohdefunktion evaluoiminen voi olla hyvinkin aikaa vievää. Optimin askelpituuden etsimisen sijasta askelpituuden valitsemiseksi on kehitetty usei-

ta menetelmiä, joista eräs on tässä työssä esiteltävä Barzilai-Borwein [20]. On myös mahdollista esimerkiksi käyttää koko optimoinnin ajan samaa, ennalta määrättyä vakioaskelpituutta.

4.2.2 Fletcher-Reeves

Fletcher-Reeves [21] on epälineaariseen optimointiin kehitetty algoritmi, joka on johdettu alkuperäisestä, lineaarisesta konjugaattigradienttimenetelmästä [22]. Fletcher-Reevesin ainoa muutos jyrkimmän laskun menetelmään on se, että etsintäsuuntana käytetään gradientin osoittaman jyrkimmän laskun suunnan $- \bigtriangledown f(x)$ sijasta nk. konjugaattisuuntaa S, joka muodostetaan uuden gradienttisuunnan ja edellisen etsintäsuunnan painotettuna summana

$$S_{k+1} = - \bigtriangledown f(x_k) + \beta_{k+1} S_k, \tag{4.12}$$

missä paino β on skaalari, joka kuvaa kohdefunktion gradienttien pituuksien suhdetta kahdessa viimeisimmässä iteraatiossa ja $S_1 = - \bigtriangledown f(x_0)$. β saadaan kaavasta

$$\beta_{k+1} = \frac{\bigtriangledown f(x_k)^T \bigtriangledown f(x_k)}{\bigtriangledown f(x_{k-1})^T \bigtriangledown f(x_{k-1})}. \tag{4.13}$$

On hyvä huomata, että painolle β on kehitetty useita vaihtoehtoisia määritelmiä, joista Fletcher-Reevesin määritelmä on yksi pelkistetyimmistä. Konjugaattisuunnan käyttämisen etuna jyrkimmän laskun suuntaan verrattuna on, että se tuottaa usein vähemmän kulmikkaan etsintäkuvion optimoinnin aikana, kuten kuvassa 4.2 on visualisoitu. Tätä voi tapahtua esimerkiksi häiriöalttiuden (ill-conditioned) aiheuttamissa kapeissa optimointimointiavaruuden laaksoissa. [9]

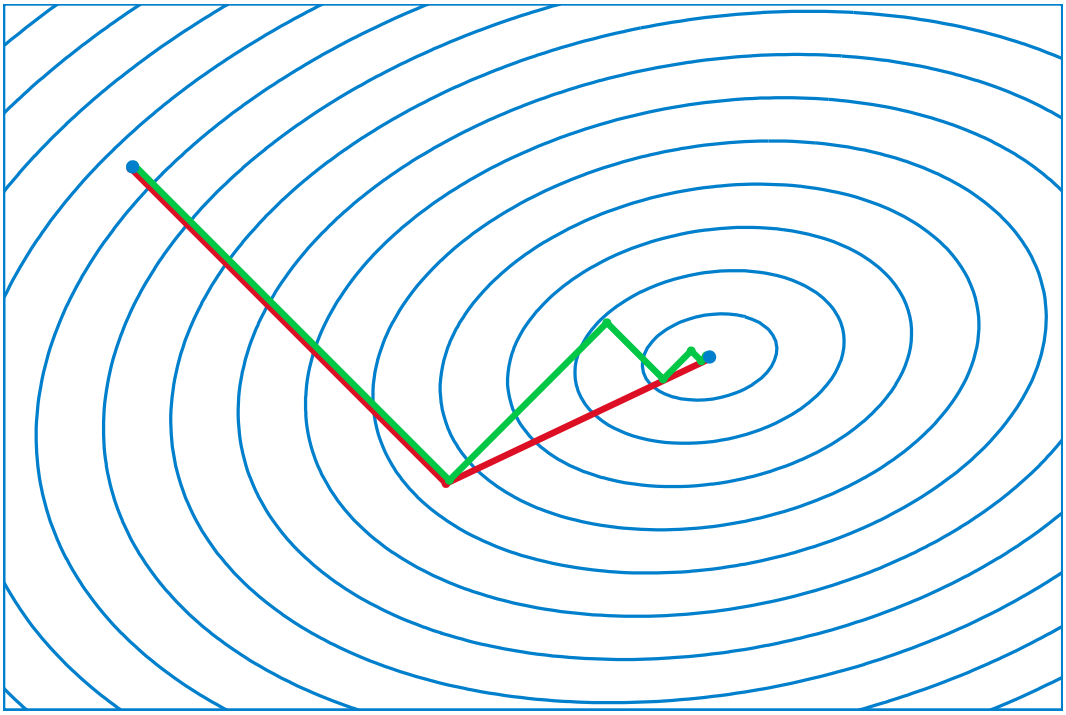
4.2.3 Barzilai-Borwein

Barzilai-Borwein on gradienttioptimointimenetelmä, joka poikkeaa perusmuotoisesta jyrkimmän laskun menetelmästä vain askelpituuden valinnan suhteen. Barzilai-Borweinin askelpituus on jyrkimmän laskun menetelmän askelpituuteen verrattuna laskennallisesti helpompi ratkaista ja tuo siten tavallisesti huomattavan nopeutuksen optimointiin. Tämän lisäksi Barzilai-Borwein näyttäisi olevan vähemmän herkkä ongelman häiriöalttiudelle kuin jyrkimmän laskun menetelmä. [20]

Barzilai-Borweinin askelpituudelle on kaksi vaihtoehtoista määritelmää

$$\alpha_k = \frac{\Delta x^T \Delta x}{\Delta x^T \Delta g} \tag{4.14}$$

ja



Kuva 4.2: *Kuvan esimerkissä [23] optimointi konjugaattisuuntaan (punainen) tuottaa vähemmän kulmikkaan etsintäkuvion kuin jyrkimmän laskun suuntaan (vihreä).*

$$\alpha_k = \frac{\Delta x^T \Delta g}{\Delta g^T \Delta g}, \tag{4.15}$$

missä $\Delta x = x_k - x_{k-1}$, $\Delta g = g_k - g_{k-1}$ ja $g_k = \bigtriangledown f(x_k)$. Molemmat määritelmät on johdettu alkuperäisessä julkaisussa [20] ja ne perustuvat kahden pisteen approksimaatioon sekanttiyhtälöstä, joka on kvasi-Newton menetelmien taustalla. Vaihtoehtoisten askelpituuksien välinen suorituskykyero on havaittu pieneksi.

4.2.4 Barzilai-Borwein modifiointi: Useampi askelpituus

Tämän diplomityön yhteydessä on kehitetty lähestymistapa, jossa lasketaan kaikille parametreille yhteisen Barzilai-Borwein -askelpituuden (4.14) sijasta useampi Barzilai-Borwein -askelpituus jakamalla parametrivektori osajoukkoihin ja laskemalla oma askelpituus jokaiselle osajoukolle. On hyvä huomata, että tällaiselle menettelylle ei ole johdettu teoreettista perustaa. Mikäli osajoukot ovat yksittäisen parametrivektorin alkion kokoisia niin askelpituuksiksi saadaan

$$\alpha_k[d] = \frac{\Delta x[d] \Delta x[d]}{\Delta x[d] \Delta g[d]}, \tag{4.16}$$

kun käydään läpi kaikki parametrivektorin dimensiot d.

Osajoukkojen mielekäs valitseminen jollain muulla tavalla riippuu suuresti opti-

mointiongelman luonteesta. On mahdollista, että hyvä valinta yksittäiseksi osajoukoksi on joukko parametreja joiden optimiarvojen tiedetään tai voidaan perustellusti olettaa olevan arvoiltaan samaa suuruusluokkaa.

Emme esitä, että edes optimaaliseen kahden tai useamman osajoukon jakoon pohjautuvaa, useamman askelpituuden Barzilai-Borwein -optimointia olisi perusteltua olettaa toimivammaksi kuin perusmuotoista, yhden askelpituuden Barzilai-Borwein -optimointia. On kuitenkin hyvä huomata, että useamman askelpituuden lähestymistapa toimii optimaalisilla askelpituusvalinnoilla aina vähintään yhtä tehokkaasti kuin yhden optimaalisen askelpituuden lähestymistapa. Helppo todistus tälle väittämälle on se, että sama suorituskyky saavutetaan, kun jokainen useista askelpituuksista valitaan samaksi kuin vertailukohtana käytettävä yksittäinen askelpituus.

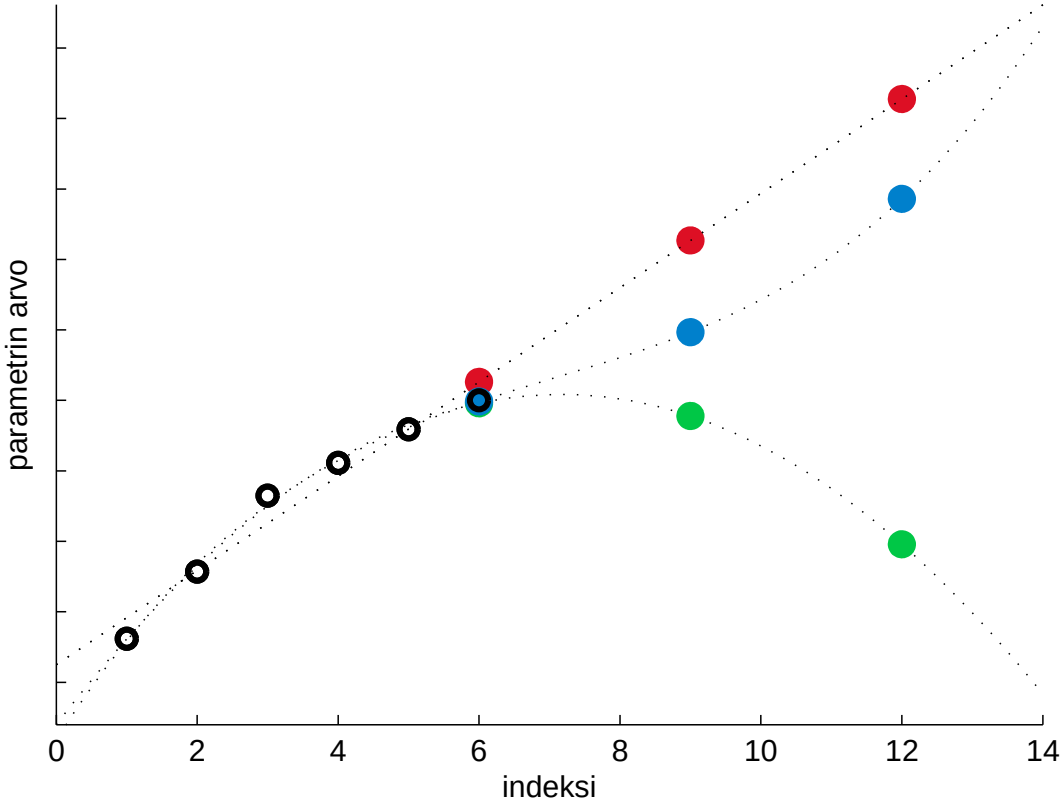
4.2.5 Barzilai-Borwein modifiointi: Polynomisovitteella ennustaminen

Tarkasteltuamme yksittäisten parametrien kehittymistä Barzilai-Borwein -optimoinnin aikana havaitsimme, että näiden kehityshistoriakäyrät olivat usein muodoiltaan säännönmukaisia kaaria, jotka alkoivat vakioitumaan optimoinnin edetessä. Tästä saimme idean polynomien sovittamiseksi parametrien kehityshistoriaan uusien parametriarvojen ennustamiseksi. Tässä aliluvussa kuvataan tähän ideaan perustuva menetelmä, jonka kehitimme. Menetelmä sopii käytettäväksi minkä tahansa optimointialgoritmin ohessa, mutta hyötyä siitä ei ole havaittu olevan kaikkien algoritmien kanssa.

Parametrien kehityshistorialla X_{hist} tarkoitetaan käytettävän optimointimenetelmän tuottamaa järjestettyä joukkoa parametrivektoreita, jotka ovat tuoneet parannuksen kohdefunktion arvoon optimoinnin edetessä verrattuna aiempaan pienimpään löydettyyn kohdefunktion arvoon. Joukon alkioina olevat parametrivekorit ovat samassa järjestyksessä kuin missä ne optimoinnin aikana on luotu. Mikäli kehityshistoria X_{hist} esitetään matriisimuodossa, jossa parametrivektorit on sijoitettu järjestyksessä sarakkeisiin niin matriisin rivi i kuvastaa i:nnen parametrin kehitystä.

Kehityshistorian taaksepäin ulottuvalle pituudelle *pituus* tulee määrittää jokin arvo, jonka suuruus vaikuttaa toisaalta polynomin sovittamiseen ja toisaalta siihen kuinka monta kertaa polynomin sovittaminen voidaan määrätyn mittaisen optimoinnin aikana suorittaa. Polynomin sovittamiseen kehityshistorian pituus vaikuttaa siten, että useampi kehityshistorian piste antaa enemmän informaatiota parametrien pidemmän aikavälin kehitystrendeistä.

Polynomin $f(x, \beta)$, jonka aste on n, kertoimiksi β valitaan ne jotka tuottavat pienimmän neliösumman (least squares) virheen S yksittäisen parametrin kehityshistorialle y, kun



Kuva 4.3: *Polynomeja on sovitettu parametrin kehityshistoriaan, jonka pituus = 6. Visualisoituna on ennusteet, kun ennustekerroin ek = 1, ek = 1.5 ja ek = 2. Ensimmäisen asteen polynomin ennusteet on visualisoitu punaisilla, toisen asteen ennusteet vihreillä ja kolmannen asteen ennusteet sinisillä palloilla.*

$$S = \sum_{x=1}^{n} r[x]^2, \tag{4.17}$$

missä $r[x] = y[x] - f(x, \beta)$ on polynomin ennustaman arvon ja kehityshistorian arvon välinen poikkeama. Mikäli polynomin aste $n = 1$, niin

$$f(x, \beta) = \beta[1] + \beta[2]x. \tag{4.18}$$

Polynomin kertoimien, β, selvittämisen jälkeen muodostetaan parametrille ennustettu arvo $y_{ennuste}$ suhteessa kehityshistorian pituuteen *pituus* siten, että

$$y_{ennuste} = f(round(pituus \cdot ek), \beta), \tag{4.19}$$

missä *ek* on ennustekerroin ja *round* pyöristysoperaatio lähimpään kokonaislukuun. Ennusteen tekeminen on havainnollistettu kuvassa 4.3.

Olemme havainneet, että polynomin sovittamista ei usein kannata tehdä aivan

optimoinnin alussa eikä myöskään heti edellisen sovittamisen jälkeen, sillä parametrien käyttäytyminen näissä kohdissa voi olla pidemmän aikavälin trendistä poikkeavaa. Näissä tapauksissa tehtävien asettumisiteraatioiden määrän ja hyvän kehityshistorian pituuden ja polynomin asteen valinnan jätämme avoimiksi, sillä uskomme niiden olevan hyvin tapauskohtaisia.

Tässä aliluvussa esitelty polynomisovitteella ennustaminen muistuttaa splini-ekstrapolointia. Merkittävänä erona on se, että splinifunktio kulkee kaikkien syötteenä annettavien pisteiden kautta. Splinifunktio on paloittain määritelty polynomien avulla näiden pisteiden välillä ja polynomit määritetään siten, että splinifunktio on jatkuva ja derivoituva syötepisteiden kohdalla. Ekstrapolointi splinifunktiolla toteutetaan yksinkertaisesti kahden ensimmäisen tai viimeisen syötepisteen välille sovitettua polynomia käyttäen.

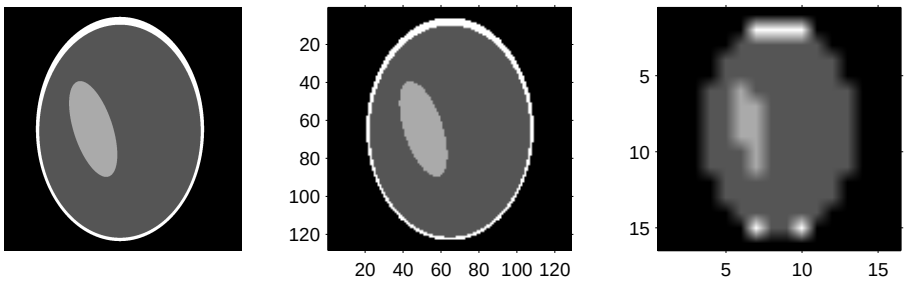
5. KOKEET

5.1 Koeasetelma

5.1.1 Simuloitu mittausdata ja sen mallintaminen

Tämän diplomityön yhteydessä tehdyissä kokeissa on käytetty todellisen PET-laitteiston tuottaman mittausdatan sijasta lokeromallin ja diskreetin Radon-muunnoksen avulla generoitua simuloitua mittausdataa. Simuloitu data on generoitu yhtä viipalekuvaa esittävän, neljä aluetta sisältävän virtuaalisen fantomin avulla, joka voidaan diskretisoida mielivaltaiseen resoluutioesitykseen. Fantomilla tarkoitetaan ominaisuuksiltaan tunnettua mittauskohdetta, jota käytetään referenssinä kuvantamisjärjestelmän toimintaa analysoitaessa. Käyttämämme virtuaalinen fantomi ja siitä diskretisoidut 128×128 ja 16×16 vokselikoon fantomit on esitelty kuvassa 5.1. Fantomin jokainen vokseli kuuluu vain yhteen neljästä alueesta. Fantomissa musta alue kuvaa kohteen ulkopuolista tausta-aluetta ja muilla sävyillä esitetyt alueet kohteen sisällä olevia alueita.

Fantomin jokaiselle vokselille i luotiin lokeromallia käyttäen fysiologista käyttäytymistä kuvaava kohinaton aika-aktiivisuuskäyrä $\Lambda_i(t)$. Hajoamisvakiota λ ei huomioitu joten n-lokeromallia käyttäen (3.1) yksinkertaistui muotoon



Kuva 5.1: *Fantomi ja siitä diskretisoidut* 128×128 *ja* 16×16 *-vokselikoon fantomit. Alue 1 on alueista suurin ja on esitettynä tummanharmaalla sävyllä, alue 2 sijaitsee kohteen reunalla valkoisella sävyllä esitettynä ja alue 3 sijaitsee lähellä kohteen keskiosaa ja on esitetty vaaleanharmaalla sävyllä.*

$$\Lambda_i(t) = \int_{\triangle t} (1 - F)(\sum_{s=2}^{N} C_{si}(t)) + C_1(t)F dt, \tag{5.1}$$

missä F on plasman tilavuusosuus kudoksessa, joka voitaisiin estimoida kaikille vokseleille, mutta joka on tämän diplomityön yhteydessä määritetty vakioksi 0.05. Lokeroiden konsentraatiot $C_{si}(t)$ saadaan käytettävästä mallista riippuen kaavasta (2.6) tai (2.8), jotka tarvitsevat vokselille i määritetyt kineettiset parametrit. Vokselikohtaiset kineettisten parametrien arvot, joiden yksikkö on $\big(ml/(g \cdot s)\big)$, on poimittu satunnaisesti yksiulotteisista normaalijakaumista, joiden aluekohtainen odotusarvo, μ, ja keskihajonta, σ, on esitetty taulukossa 5.1. Kun tämän luvun kokeiden yhteydessä puhutaan heterogeenisesta datasta niin tarkoitetaan, että käytetään taulukossa ilmoitettua keskihajontaa kineettisille parametreille. Homogeenisella datalla puolestaan tarkoitetaan dataa, jossa käytetty keskihajonta on kaikilla alueilla 0. C_1 on käyttämämme paloittain määritelty radioaktiivisuuskonsentraatiota plasmassa ajan suhteen kuvaava syötekäyrä, joka on muodostettu erään todellisen mittauksen pohjalta ja on muotoa

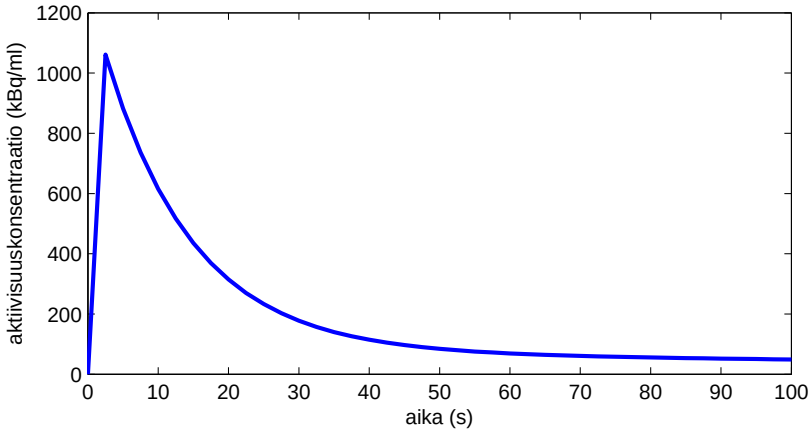
$$C_1 = (co \cdot t)(t \leq tp) + (t > tp)(A \cdot \exp(a \cdot t) + B \cdot \exp(b \cdot t) + C \cdot \exp(c \cdot t)), \tag{5.2}$$

missä $A = -2.0957$, $a = -4.3291$, $B = 80$, $b = -0.005$, $C = 1200$, $c = -0.08$, $tp = t(2)$, $co = (A \cdot \exp(a \cdot tp) + B \cdot \exp(b \cdot tp) + C \cdot \exp(c \cdot tp))/tp$ ja vektori t sisältää mittauksessa käytetyttyjen aikaikkunavälien loppuajanhetket. Tasalevyisiä aikaikkunavälejä on 40kpl ja niiden loppuajanhetket saavat arvot $t = 2.5, 5, \ldots, 100$. Sisääntulokäyrä on visualisoituna kuvassa 5.2.

Fantomille muodostettu kaikkien yksittäisten vokseleiden aika-aktiivisuuskäyristä koostuva aika-aktiivisuuskuvasarja kerrottiin kuvantamisen hyötysuhdetta simuloivalla vakiolla sc, jona käytettiin tapauksesta riippuen joko 0.05 tai 0.004. Kuvantamisen hyötysuhteella tarkoitetaan sitä osuutta merkkiaineen aiheuttamasta gammasäteilystä, jonka PET-laitteisto onnistuu mittaamaan. Tähän lukemaan vaikuttaa sekä detektorien asemointi potilaan ympärillä että detektorien ominaisuudet. Edellä il-

	Alue 1		Alue 2		Alue 3	
	μ_1	σ_1	μ_2	σ_2	μ_3	σ_3
K_1	0.7000	0.0700	0.2500	0.0250	0.9000	0.0900
k_2	0.7368	0.0737	0.2632	0.0263	0.9474	0.0947
k_3	0.0200	0.0020	0.0100	0.0010	0.0400	0.0040

Taulukko 5.1: *Alueittaiset kineettisten parametrien odotusarvot, μ, ja keskihajonnat, σ.*



Kuva 5.2: *Syötekäyrä C_1.*

moitetut, tässä työssä käytetyt hyötysuhdelukemat ovat samaa suuruusluokkaa kuin todellisten PET-laitteistojen lukemat. Sinogrammiaikasarja muodostettiin tekemällä diskreetti Radon-muunnos jokaiselle aika-aktiivisuuskuvasarjan 40:stä aikaikkunasta. Radon-muunnos tehtiin käyttäen 180 eri kulmaa ja simuloidun fantomin dimensioista ja tapauskohtaisista syistä riippuvaa määrää binejä. 128×128 -fantomin kanssa käytettiin joko 367:ää tai 185:ttä biniä ja 16×16 -fantomin kanssa 49:ää tai 27:ää biniä. Suurempia binimääriä käytettiin useimmissa tapauksissa, sillä haluttiin välttyä liian pienen binimäärän takia mahdollisesti muodostuvalta, käytännölliseltä alimääräytyneisyysongelmalta (ill-posed), joihin alustavat empiiriset testit pienemmillä binimäärillä viittasivat. Saatuihin sinogrammin binien arvoihin lisättiin lopuksi mallin (3.4) mukaista Poisson-luonteista ja normaalijakautunutta virhettä. Normaalijakautuneen virheen keskihajontana käytettiin kaikissa kokeissa arvoa $\sigma = 2$.

Testidatan generoinnissa käytettyä lokeromallia ja fantomia käytetään myös testidatan mallintamisessa. Mallinnettaessa oletetaan radioaktiivisen hajoamisen aiheuttama virhe normaalijakautuneeksi toisin kuin testidatan generoinnissa, jossa se oletettiin Poisson-jakautuneeksi. Erittäin suurivarianssinen Poisson-kohina kuitenkin lieventää tätä approksimaatiovirhettä merkittävästi. Tässä diplomityössä ei tutkita tätä epäideaalisemman mallin tai epätäydellisen aluesegmentoinnin käyttämisen vaikutusta mallintamiseen. Plasman tilavuusosuus kudoksessa, F, oletetaan mallinnettaessa tunnetuksi.

5.1.2 Optimointimenetelmien vertailemisesta

Optimoitavan parametrivektorin oletusalkuarvaus on, ellei muuta mainita, ollut jokaisessa kokeessa sama kaikilla vertailtavilla menetelmillä ja jokaisen yksittäisen menetelmän jokaisella erillisellä optimointiajolla mikäli näitä on tehty useita. Alkuarvaukseksi yksittäiselle parametrille poimittiin taulukosta 5.1 tätä vastaava testida-

tan generoinnissa käytetty, alueellinen kineettisen parametrin odotusarvo μ. Tässä työssä alkuarvauksena käytettävät odotusarvot oletetaan tunnetuiksi optimoinnin alussa, mutta todellisuudessa ne estimoidaan etukäteen konventionaalisilla menetelmillä. Olemme havainneet [24], että optimointi on alkuarvauksen suhteen melko robusti joten alkuarvauksen valitseminen ei ole merkittävä ongelma.

Optimointimenetelmien suorituskykyvertailut on toteutettu kohdefunktion evaluointimääriin perustuen. Yhtenä vaihtoehtoisena mittarina olisi ollut vertailla optimointimenetelmien kuluttamaa laskenta-aikaa. Funktion evaluointien määrä mittarina mahdollistaa ohjelmistoympäristöstä, laitteistosta ja laitteiston muusta kuormituksesta riippumattoman mittarin, jonka mittaamat tulokset on toistettavissa täsmällisesti. Laskenta-ajallinen mittari puolestaan tuottaa selvästi paremman arvion optimointimenetelmien laskennallisesta raskaudesta tietyssä laitteisto- ja ohjelmistoympäristössä, mutta tulokset eivät yleensä ole täsmällisesti toistettavissa ympäristöriippuvaisuuden takia. Esimerkiksi tässä työssä käytetyt ensimmäisen asteen optimointimenetelmät laskevat jokaisella iteraatiolla kohdefunktiolle evaluoinnin lisäksi myös gradientin, ja näin ollen käyttävät yhdessä iteraatiossa selvästi enemmän laskenta-aikaa kuin useimmat nollannen asteen menetelmät, jotka suorittavat vain funktion evaluoinnin.

Luvussa 4 esitellyistä optimointimenetelmistä käytetään tämän diplomityön kokeissa seuraavia lyhenteitä:

- SD: Jyrkimmän laskun menetelmä
- FR: Fletcher-Reeves
- BB: Barzilai-Borwein
- MTS: Monilentoratahaku
- jDE: Itseohjautuva differentiaalinen evoluutio
- CLPSO: Kattavasti oppiva hiukkasparvioptimointi

Mikäli jyrkimmän laskun menetelmä tai Fletcher-Reeves on vertailussa mukana, niin näiden suorituskykyä tarkastellaan vain vakioaskelpituutta käyttäen sen sijaan, että jokaisella iteraatiolla etsittäisiin optimaalinen askelpituus. Vertailuissa ilmoitetut askelpituudet ovat askelpituuden α käänteislukuja, sillä askelpituudet ovat hyvin pieniä. Askelpituuden merkintapaa on lisäksi lyhennetty siten, että e:llä tarkoitetaan kymmenen potensseja. Vertailuissa ilmoitettu askelpituus $1e8$ vastaa siis todellisuudessa askelpituutta $\alpha = 10^{-8}$

5.2 Koe 1: Optimointimenetelmien vertailu

Tässä kokeessa vertaillaan optimointimenetelmien nopeutta ja käyttäytymistä kohdefunktiota minimoitaessa suurimman uskottavuuden estimaatin löytämiseksi sovelluskohteessamme. Tämän kokeen yhteydessä ei pyritä analysoimaan saavutettuja tuloksia heterogeenisuuden estimoinnin näkökulmasta. Pyrkimyksenä on sen sijaan tehdä tulosten pohjalta päätelmiä optimointiavaruuden ominaispiirteistä, kun mittausdataa mallinnettaessa käytössä on tietty lokeromalli. Optimointimenetelmiä vertaillaan tässä kokeessa yhtä simuloitua homogeenista dataa, mutta useita eri aloituspisteitä käyttäen.

Koska optimointiongelma on suuridimensioinen niin voidaan olettaa todennäköiseksi, että mikä tahansa tietyn lokeromalin käytön seurauksena muodostuva, yhteen mittausdataan liittyvä yksittäinen optimointiavaruus omaa myös muilla mahdollisilla mittausdatoilla esiintyvät optimointiavaruuden tyypilliset ominaispiirteet. Näin ollen tulosten pohjalta voidaan tehdä todennäköisesti paikkansa pitäviä, lokeromallikohtaisia päätelmiä ainakin optimointiavaruuden mahdollisesta multimodaalisuudesta, eli siitä omaako se tyypillisesti useita minimi- tai maksimikohtia vai ei. Lisäksi kokeessa valittiin käytettäväksi alueellisesti homogeenista dataa, sillä oletettiin, että siihen liittyvä optimointiavaruus on optimointimointimenetelmien suorituskyvyn kannalta todennäköisesti edullisempi kuin heterogeenista dataa käytettäessä. On hyvä tiedostaa, että homogeenisen datan yhteydessä parametrien ML-estimaatti ei ole estimointivirheestä johtuen homogeeninen.

Useiden aloituspisteiden käyttö tuo informaatiota siitä kuinka robusteja eri optimointialgoritmit ja kuinka robusti menetelmämme on aloituspisteen valinnan suhteen. Saman datan ja eri aloituspisteiden käyttämisen etuna optimointialgoritmien vertailemiseen usean datan käyttämiseen verrattuna on se, että kaikkien optimointiajojen tapauksissa kohdefunktion arvot ovat vertailukelpoisia. Pienemmän kohdefunktion arvon tuottamat parametrit voidaan olettaa paremmaksi arvaukseksi suurimman uskottavuuden estimaatista kuin suuremman kohdefunktion arvon tuottavat parametrit.

Samojen aloituspisteiden käyttäminen useasti ei ole vertailun kannalta mielekästä, sillä käytetyt gradienttimenetelmät ovat deterministisiä ja siten käyttäytyisivät kaikissa tällaisissa ajoissa identtisesti. Satunnaisuutta sisältäville nollannen asteen menetelmille samojen aloituspisteiden käyttäminen toisi informaatiota algoritmien sisäisen satunnaisuuden vaikutuksesta optimointiin, mutta pitkäkestoisessa optimoinnissa voidaan olettaa että yksittäisten hyvien ja huonojen arvausten tuoma vaikutus tasoittuu.

Luvussa 4 esiteltyjen optimointimenetelmien lisäksi tämän luvun kokeita varten on kehitetty yhdistelmämenetelmiä, joissa optimointi aloitetaan Barzilai-Borwein -

algoritmilla, mutta tietyn kohdefunktion evaluointimäärän jälkeen vaihdetaan johonkin esitellyistä nollannen asteen menetelmistä. Tällaisista yhdistelmämenetelmistä käytetään tämän kokeen yhteydessä lyhenteitä BB+MTS,BB+jDE ja BB+CLPSO.

Vertailtavilla optimointimenetelmillä käytettiin seuraavia asetuksia

- jDE: $F_l = 0.1$, $F_u = 0.9$ ja $\tau_1 = \tau_2 = 0.1$, kuten alkuperäisessä julkaisussa [16] on suositeltu ja populaatiokooksi valittiin alustaviin testiajoihin perustuen 20.
- MTS: populaatiokoko $M = 5$, $TestiKierrokset = 3$,$HakuKierrokset = 100$,$ParhaanKierrokset = 150$, $BONUS1 = 10$, $BONUS2 = 1$, $a = rand[0.4, 0.5]$, $b = rand[0.1, 0.3]$ ja $c = rand[0, 1]$ alkuperäisen julkaisun [13] suositusten mukaisesti. Parametrikohtaisiksi rajoiksi määriteltiin $YRAJA[d] = x_o[d]+0.25$ ja $ARAJA[d] = x_o[d]-0.25$, kun x_o on luvussa 5.1.2 kuvailtu oletusalkuarvaus.
- CLPSO: $m = 7$, $w_0 = 0.9$, $w_1 = 0.4$ ja $c = 1.49445$ valittiin algoritmin kehittäjien ohjeistukseen [18] perustuen. Populaatiokooksi ps valittiin 20, alkunopeudet valittiin satunnaisesti hieman suositeltua kapeammalta tasajakautuneelta väliltä $[-0.02, 0.02]$ ja maksiminopeudeksi V_{max} asetettiin 0.02.

Optimointialgoritmien suorituskykyvertailu toteutettiin siten, että suoritettiin tietty määrä optimointiajoja jokaisella menetelmällä ja tietyllä budjetilla kohdefunktion evaluointeja. Parhaimmaksi valittiin se algoritmi, joka oli saavuttanut näistä erillisajoista laskettuna parhaimman keskimääräisen kohdefunktion arvon ajojen lopussa. Yksittäisen erillisajon saavuttamaksi kohdefunktion arvoksi katsotaan parhaan populaation jäsenen omaava kohdefunktion arvo.

Aloituspisteitä generoitiin 20 kappaletta jokaista ajorypästä varten. Ajoryppäällä tarkoitetaan jokaisen vertailtavan optimointimenetelmän suorittamista yhden kerran ajorypästä varten generoituja aloituspisteitä populaation alustamisessa hyödyntäen. CLPSO:n ja jDE:n populaatiokoko on 20, joten ne hyödyntävät kaikkia generoituja aloituspisteitä populaation alustamiseksi. MTS alustaa viiden suuruisen populaationsa viidellä ensimmäiseksi generoidulla aloituspisteellä ja kaikki gradienttimenetelmät alustavat parametrivektorin ensimmäisellä aloituspisteellä. Tällaisen alustusmenettelyn seurauksena jokaisessa ajoryppäässä kaikilla optimointimenetelmillä on mahdollisimman tasavertainen lähtötilanne.

Yhdistelmämenetelmillä alustaminen toteutettiin BB:n jälkeen uuteen menetelmään siirryttäessä siten, että BB:n optimoima parametrivektori sijoitettiin populaation ensimmäiseksi jäseneksi ja loput jäsenet poimittiin aloituspistejoukon toisesta jäsenestä alkaen.

5.2.1 2K-lokeromallia käyttäen

Tässä aliluvussa vertaillaan optimointimenetelmien suorituskykyä, kun sekä datan generoinnissa että sen mallintamisessa käytetään 2K-lokeromallia. Kokeessa käytetyn fantomin koko on 16×16, binien määrä 49, hyötysuhde $sc = 0.004$, optimointialgoritmien budjetti 10^6 kohdefunktion evaluointia ja yhdistelmämenetelmien vaihtokohta 25000 evaluointia. Jyrkimmän laskun menetelmä ja Fletcher-Reeves käyttivät empirisesti hyväksi valinnaksi havaittua vakioaskelpituutta $\alpha = 5e5$.

Jokaisella optimointimenetelmällä suoritettiin 15 toistoa eri aloituspisteistä aloittaen. Tarvittavat 20 aloituspistettä jokaista toistoa varten generoitiin siten että jokaiselle alueella i sijaitsevalle vokselille K_1:n alkuarvaus poimittiin satunnaisesti väliltä [$\mu_i(K_1)$-0.1 $\mu_i(K_1)$+0.1] ja k_2:n väliltä [$\mu_i(k_2)$-0.1 $\mu_i(k_2)$+0.1]. Keskihajonnat μ on ilmoitettu taulukossa 5.1. Virheparametrin σ alkuarvaus poimittiin väliltä [1.99 2.01].

Kokeen tulokset

Jokaiselle optimointimenetelmälle laskettiin kaikista erillistoistoista keskiarvoistettu kohdefunktion arvon kehittyminen kohdefunktion evaluointimäärän suhteen. Tulokset löytyvät visualisoituna liitteen kuvaajista B.1 ja B.2. Optimointimenetelmien saavuttamat keskimääräiset kohdefunktion arvot ja erillisajoista lasketut keskihajonnat saavutettujen kohdefunktion arvojen välillä on esitelty taulukossa 5.2.

Tulosten pohjalta havaitaan, että Barzilai-Borwein, Fletcher-Reeves ja BB+jDE -menetelmien erillisajot konvergoituvat todennäköisesti kohti samaa pistettä. Koska lisäksi erillisajot ovat aloittaneet satunnaistetuista aloituspisteistä niin voidaan päätellä, että optimointiavaruus saattaa olla unimodaalinen.

Voidaan myös havaita, että konjugaattisuuntaa käyttävä Fletcher-Reeves suoriutuu huomattavasti jyrkimmän laskun suuntaa käyttävää jyrkimmän laskun menetelmää paremmin ja että puhtaat nollannen asteen menetelmät ovat varsin selvästi hitaimpia.

5.2.2 3K-lokeromallia käyttäen

Tässä aliluvussa vertaillaan optimointimenetelmien suorituskykyä, kun sekä datan generoinnissa että sen mallintamisessa käytetään 3K-lokeromallia. Kokeessa käytetyn fantomin koko on 16×16, binien määrä 49, hyötysuhde $sc = 0.004$, optimointialgoritmien budjetti $1.685 \cdot 10^6$ kohdefunktion evaluointia ja yhdistelmämenetelmien vaihtokohta $2 \cdot 10^5$ evaluointia. Jyrkimmän laskun menetelmä ja Fletcher-Reeves käyttivät vakioaskelpituutta $\alpha = 1e8$.

Alkuarvaukset σ:aa, K_1:stä ja k_2:sta edustaville parametreille generoitiin kuten edellisessä aliluvussa 2K-mallia käyttäen ja k_3:lle väliltä [$\mu_i(k_3)$-0.05 $\mu_i(k_3)$+0.05].

Optimointialgoritmi	Kohdefunktion arvo	Keskihajonta
FR	7.88873474284e5	1.12e-10
BB	7.88873474284e5	2.73e-10
BB+jDE	7.88873474288e5	1.09e-6
SD	7.88873474675e5	7.51e-6
BB+MTS	7.88873478311e5	1.42e-4
BB+CLPSO	7.88873484179e5	2.65e-4
MTS	7.88873622174e5	9.26e-2
jDE	7.88873828840e5	5.18e-2
CLPSO	7.88883632295e5	3.48e-1

Taulukko 5.2: *Optimointialgoritmien saavuttama kaikista erillisajoista laskettu keskimääräinen kohdefunktion arvo ja keskihajonta testiajojen lopussa 2K-lokeromallia käyttäen.*

Keskihajonnat $\mu_i(k_3)$ on ilmoitettu taulukossa 5.1.

Kokeen tulokset

Optimointimenetelmien keskimääräiset suorituskyvyt on visualisoitu liitteen kuvaajissa B.3 ja B.4 ja taulukosta 5.3 löytyvät optimointimenetelmien saavuttamat keskimääräiset kohdefunktion arvot ja erillisajojen välisten kohdefunktion arvojen keskihajonta.

Tulokset poikkeavat huomattavasti 2K-mallilta saaduista vastaavista tuloksista. Vaikka puhtaat nollannen asteen menetelmät ovat yhä hitaampia kuin gradienttimenetelmät niin BB+jDE ja BB+MTS yhdistelmämenetelmät menestyvät vertailussa parhaiten. Barzilai-Borwein ja Fletcher-Reeves -gradienttimenetelmät näyttävät konvergoituvan kohti sellaisia ratkaisuja, joita vastaavat kohdefunktion arvot ovat BB+jDE:n tulosten perusteella verrattaen etäällä globaalin minimikohdan kohdefunktion arvosta. Selityksenä ilmiölle voi olla se, että optimointiavaruus on multimodaalinen.

Yksikään gradienttimenetelmien erillisajo ei lähtenyt konvergoitumaan kohti sellaista lokaalia minimiä, jossa kohdefunktio saisi huomattavasti muista erillisajoista poikkeavan arvon, vaan kaikki erillisajot päätyivät kohdefunktion arvon osalta suhteellisen lähelle toisiaan. Tämä voisi viitata siihen, että joukko lokaaleja minimikohtia sijaitsee globaalin minimikohdan ympäristössä laajan attraktioaltaan (basin of attraction) pohjalla. Tämän tyyppinen optimointiavaruus myös selittäisi yhdistelmämenetelmien menestyksen, sillä nollannen asteen menetelmät eivät juutu lokaaleihin minimikohtiin.

Optimointialgoritmi	Kohdefunktion arvo	Keskihajonta
BB+jDE	8.3660449e5	4.87e-2
BB+MTS	8.3660530e5	1.06e-1
FR	8.3660549e5	3.44e-5
BB	8.3660559e5	3.47e-2
BB+CLPSO	8.3660581e5	9.09e-2
SD	8.3660889e5	7.92e-2
jDE	8.3661023e5	9.08e-1
CLPSO	8.3662968e5	4.92e-1
MTS	8.3678139e5	1.00e2

Taulukko 5.3: *Optimointialgoritmien saavuttama kaikista erillisajoista laskettu keskimääräinen kohdefunktion arvo ja keskihajonta testiajojen lopussa 3K-lokeromallia käyttäen.*

5.3 Koe 2: Nopeimman käytännöllisen optimointialgoritmin määrittäminen

Kokeen tarkoituksena on selvittää mikä vertailtavista optimointialgoritmeista on kaikkein käyttökelpoisin heterogeenisuuden estimointimenetelmämme kanssa käytettäväksi mahdollisimman lähellä todellista käyttötilannetta vastaavassa koeympäristössä. Käyttökelpoisuuden kriteerinä on ensisijaisesti algoritmin nopeus.

Tässä kokeessa vertailu suoritetaan sekä heterogeenista että homogeenista dataa käyttäen, sillä estimointivirheen korjausmenetelmämme vaatii ML-estimaatin määrittämisen näissä molemmissa tapauksissa. Käytettävä lokeromalli on 3K, sillä se kertoo kolmen kineettisen parametrin avulla enemmän kohdekudoksen ominaisuuksista kuin kahden parametrin 2K-malli ja on siten toivottavampi. Kokeessa käytetyn fantomin koko on lähestulkoon lääketieteelliseen tutkimukseen kelpaava 128×128 ja optimoitavan parametrivektorin pituudeksi muodostuu 24121.

Parametrien määrän ollessa huomattavasti suurempi kuin ensimmäisen kokeen 3K-vertailussa, jossa parametrivektorin pituus oli 337, eivät nollannen asteen menetelmät ole enää käytännöllisiä. MTS joutuu suorittamaan funktion evaluointeja yhdessä iteraatiossa keskimäärin parametrivektorin pituutta vastaavan määrän, eli lähes sata kertaa enemmän kuin ensimmäisessä iteraatiossa. Lisäksi on hyvä huomata, että yksittäisen funktion evaluoinnin laskenta-aika on pidentynyt merkittävästi. CLPSO ja jDE puolestaan olettavat, että populaation koko on lähellä samaa suuruusluokkaa kuin parametrivektorin pituus ja suorittavat yhdessä iteraatiossa jokaista populaation jäsentä kohti yhden funktion evaluoinnin. Evaluointimäärä yhdessä iteraatiossa saattaa siten jäädä MTS:ää pienemmäksi, mutta tästä saatava hyöty menetetään CLPSO:n ja jDE:n vaatiessa useita iteraatioita, eli sukupolvia, jotta evoluutiosta saatava hyöty alkaisi näkymään. Koska tämän kokeen yhteydessä eva-

luointibudjetti riittäisi nollannen asteen menetelmillä vain kymmeniin iteraatioihin, ne joudutaan pudottamaan pois vertailusta.

Ensimmäisen kokeen heikkoihin tuloksiin perustuen myöskin jyrkimmän laskun menetelmä jätettiin vertailun ulkopuolelle. Hyvin menestyneen Fletcher-Reevesin suorituskykyä sen sijaan tarkastellaan useita eri vakioaskelpituuksia käyttäen ja Barzilai-Borweinin suorituskykyä tutkitaan sekä perusmuotoisena että luvuissa 4.2.4 ja 4.2.5 esiteltyjen modifiointien kanssa.

Usean askelpituuden modifiointia kokeillaan kahta eri osajoukkojakoa käyttäen. Ensimmäissa jaossa tiettyä kineettistä parametria tietyllä kohteen alueella edustavat parametrivektorin alkiot muodostavat yksittäisen osajoukon. Toisessa jaossa jokainen parametrivektorin alkio muodostaa oman osajoukkonsa.

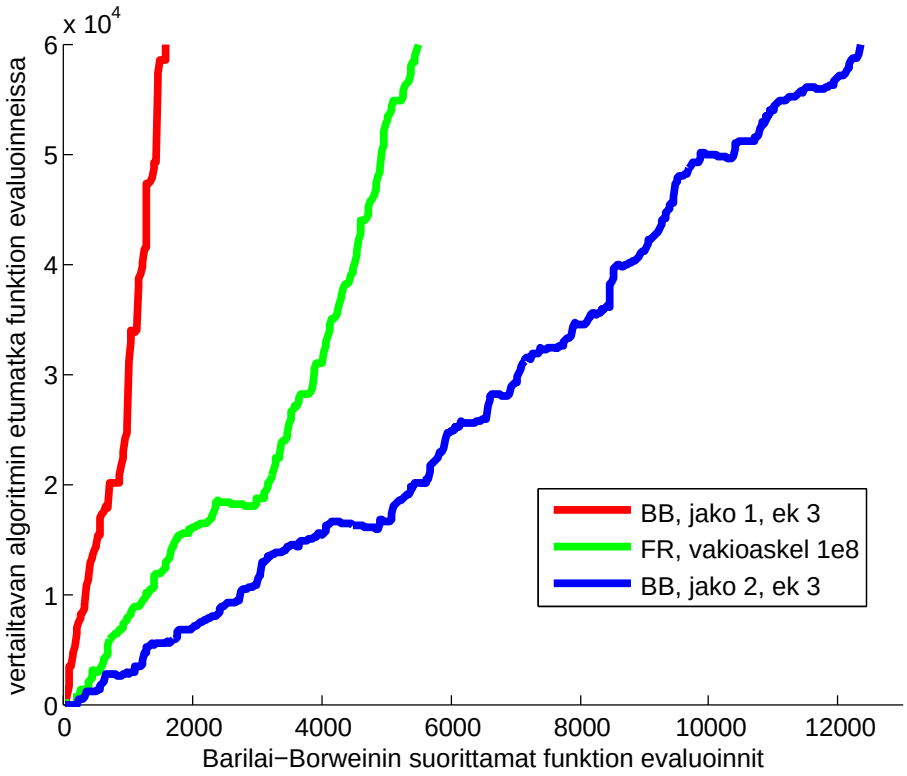
Polynomisovitteella ennustamista testataan vain ensimmäisen asteen polynomia eli suoraa käyttäen, mutta useilla eri ennustekertoimilla. Tähän valintaan päädyttiin, sillä suoran yhteydessä eri ennustekertoimien merkitystä on helpompi vertailla, kuin toisen ja sitä korkeamman polynomien yhteydessä. Näillä korkean ennustekertoimen käyttäminen tuottaa eksponentiaalisesti suuremman tai pienemmän ennusteen kuin pientä ennustekerrointa käytettäessä.

Eri menetelmillä suoritettiin vain yksittäiset optimointiajot oletusalkuarvauksesta aloittaen sekä yksittäistä heterogeenista että homogeenista dataa käyttäen. Toistojen puuttumisen selityksenä on puhtaasti laskenta-ajalliset syyt optimoinnin ollessa mm. kohdefunktion evaluoinnin ja gradientin laskennan suhteen huomattavasti raskaampaa kuin ensimmäisessä kokeessa. Ensimmäisen kokeen tulosten perusteella voimme toisaalta todeta, että FR- ja BB-algoritmien erillisajot suoriutuvat tavallisesti hyvin samankaltaisella suorituskyvyllä, kun alkuarvaukset eivät poikkea merkittävästi toisitaan.

Kokeessa binien määrä oli 367, hyötysuhde $sc = 0.05$ ja optimointialgoritmien budjetti $2e5$ kohdefunktion evaluointia. Polynomilla ennustamisessa käytettiin 50:n pituista parametrien kehityshistoriaa ja suoritettiin 25 asettumis-iteraatiota, ennenkö alettiin kokoamaan uutta kehityshistoriaa seuraavaa ennustetta varten. Testatut ennustekertoimet ek olivat 2,3 ja 4 ja FR:n askelpituudet $7e7$, $1e8$, $5e8$ ja $1e9$.

Kokeen tulokset

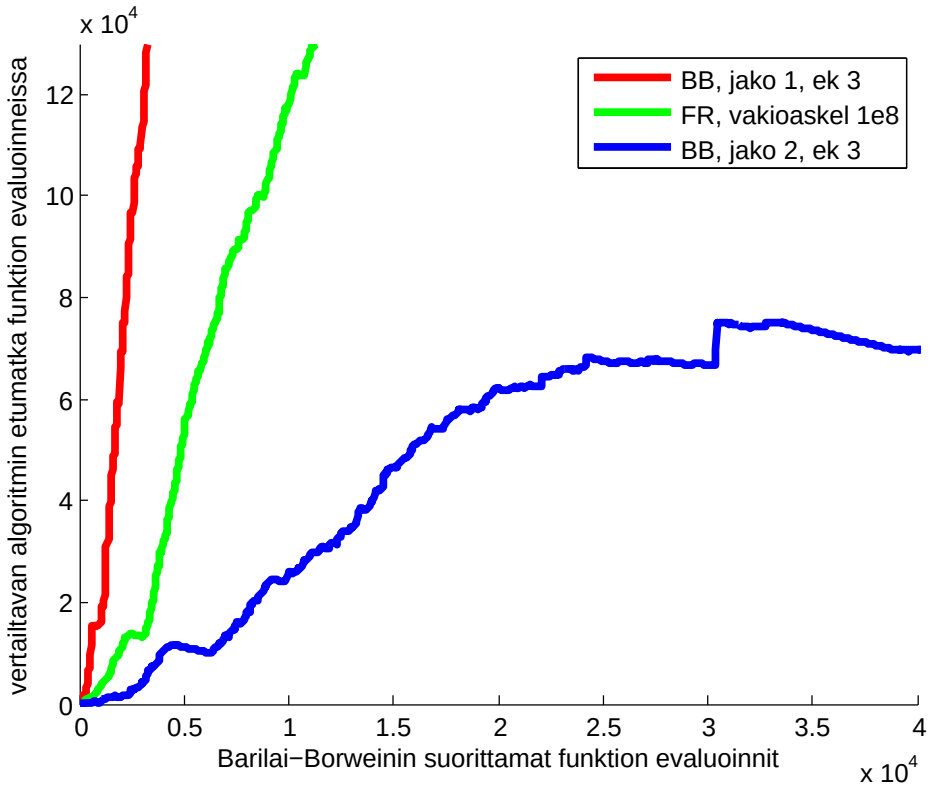
Kaikkien vertailtavien menetelmien suorituskykykuvaajat B.5 ja B.6 löytyvät liitteestä. Kuvaajissa 5.3 ja 5.4 puolestaan vertaillaan tulosten pohjalta muutaman lupaavan menetelmän suorituskykyä suhteessa Barzilai-Borweiniin, joka on tähän asti ollut oletusarvoisesti käytetty optimointimenetelmä. Suorituskykyä suhteessa Barzilai-Borweiniin tarkastellaan siltä kannalta kuinka monta kohdefunktion iteraatiota enemmän Barzilai-Borwein vaatii saavuttaakseen vertailtavien algoritmien saavuttaman kohdefunktion arvon.



Kuva 5.3: *Vakioaskelta käyttävän Fletcher-Reevesin ja kahden eri Barzilai-Borwein -modifioinnin etumatka funktion evaluoinneissa suhteessa perusmuotoiseen Barzilai-Borweiniin, kun ML-estimointi on suoritettu heterogeeniselle datalle.*

Vertailun tuloksista havaitaan, että molemmat testatut modifioinnit tuovat lisäsuorituskykyä perusmuotoiseen BB-optimointiin verrattuna sekä erikseen että yhdessä käytettyinä. Hieman yllättävää oli, että polynomilla ennustamisen yhteydessä kaikki testatut ennustekertoimet tuottivat hyvin samankaltaisen, mutta selkeän suorituskyvyn parannuksen. Koska suorituskyky ei ollut ennustekertoimen valinnan suhteen kovinkaan herkkä niin voidaan olettaa, että se ei ole sitä myöskään sovitetun suoran kulmakertoimen suhteen. On siis mahdollista, että ennustamisen tuomaa suorituskykylisää voitaisiin todennäköisesti parantaa mikäli ennusteita laskettaisiin tiheämmin, mutta epätarkemmin esimerkiksi pienentämällä asettumisiteraatioiden ja suoran sovittamiseen käytettyjen pisteiden määrää.

Usean askelpituuden modifioinnin tuloksia vertaillessa havaitaan, että alue- ja parametrityyppikohtainen jako toimi paremmin kuin parametrikohtainen jako. Parametrikohtaisen jaon merkittävä etu on toisaalta se ettei kohteen alueita tarvitse ylipäätään määrittää etukäteen. On myös hyvin todennäköistä, että todellisessa käyttötapauksessa näiden osajoukkojakojen välinen suorituskykyero kaventuu merkittävästi mm. siksi, että kohteen alueet eivät ole yhtä ideaalisesti määritettävissä ja ovat huomettavasti pienempiä. Vaatimus alueiden etukäteen määrittämisestä ja



Kuva 5.4: *Vakioaskelta käyttävän Fletcher-Reevesin ja kahden eri Barzilai-Borwein -modifioinnin etumatka funktion evaluoinneissa suhteessa perusmuotoiseen Barzilai-Borweiniin, kun ML-estimointi on suoritettu homogeeniselle datalle.*

edellä mainitut seikat saattavat tehdä tästä, tämän kokeen nopeimmasta menetelmästä todellisessa käyttötapauksessa epäkäytännöllisen.

Testissä havaittiin, että vakioaskelta käyttävän Fletcher-Reevesin suorituskyky on merkittävästi riippuvainen askelpituuden valinnasta. Testit myös vahvistivat jo aiemmin tämän diplomityön teon yhteydessä tehtyä havaintoa, että eri askelpituusvalintojen välinen suorituskykyero pysyy lähes vakiona eri iteraatiomäärillä. Tätä havaintoa on mahdollista hyödyntää tulevaisuudessa hyvän askelpituuden automaattiseksi määrittämiseksi esimerkiksi puolitushakua käyttäen suorittamalla pieniä määriä testi-iteraatioita eri askelpituuksilla.

Kuvaajista 5.3 ja 5.4 havaitaan, että ensimmäistä osajoukkojakoa ja polynomiennustetta käyttävä Barzilai-Borwein -modifiointi ja vakioaskelta käyttävä Fletcher-Reeves suoriutuvat sekä heterogeenisen että homogeenisen datan yhteydessä tasaisen hyvällä suorituskyvyllä suhteessa Barzilai-Borweiniin koko optimoinnin ajan. Ensiksi mainitulle nähdään kuvaajista noin 30-kertainen ja Fletcher-Reevesille noin 10-kertainen nopeus suhteessa Barzilai-Borweiniin. Toista aluejakoa ja polynomiennustetta hyödyntävä BB-modifiointi puolestaan lähentyy suorituskyvyltään perusmuotoista Barzilai-Borweinia optimoinnin suhteellisen menestyksekkäiden alkuite-

raatioiden jälkeen ainakin homogeenisen datan yhteydessä. Heterogeenisen datan tapauksessa optimointi-iteraatioita suoritettiin algoritmeilla pienempi määrä, joten on hyvä huomata, että kuvaaja ei ole evaluointimäärien suhteen yhtä kattava.

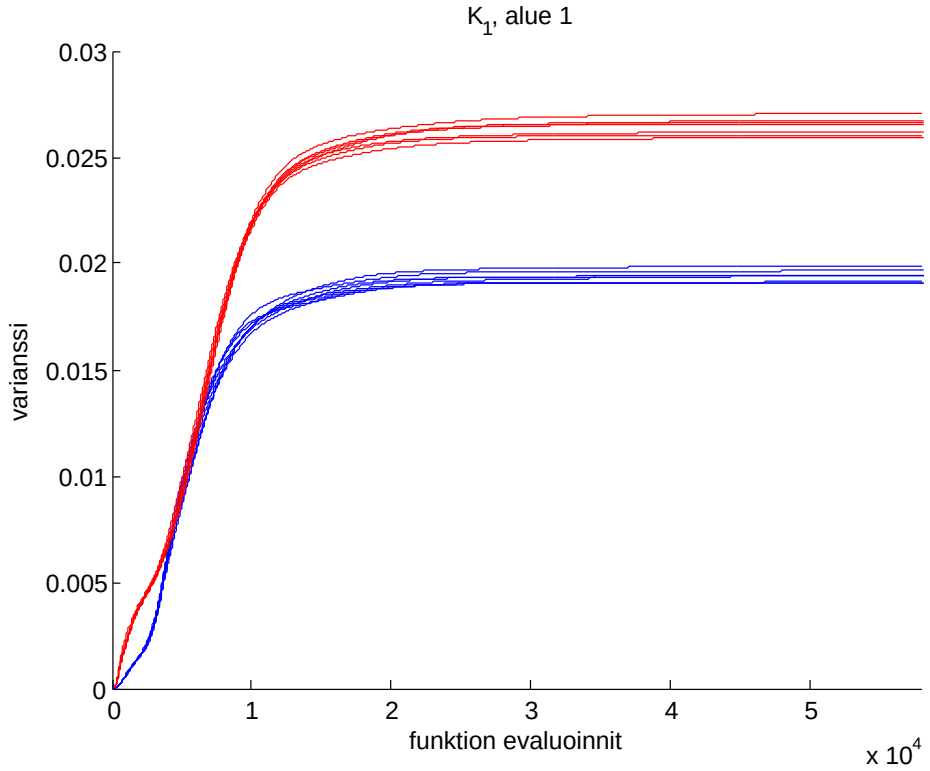
Vaikka tässä kokeessa parhaiten suoriutunut Barzilai-Borwein -modifiointi onkin todennäköisesti todellisessa käyttötilanteessa epäkäytännöllinen niin sen suhteellista suorituskykyä tämän kokeen asetelmassa voidaan pitää vertailukohtana sille kuinka paljon nopeuttamisen varaa algoritmin valinnalla ja hienosäätämisellä vähintäänkin on yhä jäljellä. Vertailun nopein käytännöllinen menetelmä, Fletcher-Reeves, jää kolmasosaan tästä suorituskyvystä.

5.4 Koe 3: Estimointivirheen korjaaminen, kun ML-estimaattia ei saavuteta

Kokeessa 1 havaittiin, että 3K-mallin yhteydessä ML-estimaattia ei saavuteta gradienttimenetelmiä käyttäen. Tässä kokeessa tarkastelemme kuinka hyvin luvussa 3.3 esitelty ML-estimointivirheen korjaaminen tästä seikasta huolimatta onnistuu, kun gradienttioptimointi-iteraatioiden määrä kasvaa. Ideaalisesti meillä olisi oltava ML-estimaatti vertailukohtana, sillä sitä käyttäen saavutetaan paras mahdollinen virheenkorjaustulos, mutta äärimmäisen laskennallisen raskauden takia tämän määrittäminen ja vertailuun mukaan ottaminen on rajattu tämän kokeen ulkopuolelle. Sen sijaan pyrimme tekemään päätelmiä virheenkorjauksen onnistumisesta mm. sillä perusteella kuinka hyviä heterogeenisuusestimaatteja saadaan verrattuna estimaatteihin, joista virhettä ei ole korjattu.

Todellisessa käyttötapauksessa käytettävissä on vain yksi ainoa mittausdata, mutta tämän kokeen yhteydessä generoimme samoilla asetuksilla seitsemän heterogeenista dataa, eli simuloitua mittausdataa, joissa kohina ja yksittäisten kineettisten parametrien arvot saavat omat realisaationsa. Tämän menettelyn ansiosta saadaan pienennettyä poikkeuksellisten kohina- ja parametrirealisaatioiden vaikutusta tuloksiin. ML-estimointi suoritetaan jokaiselle datalle erikseen ja alueittaiset kineettisten parametrien varianssit talletetaan koko optimoinnin ajalta. Näin saaduista seitsemästä varianssin kehittymiskäyrästä jokaista aluetta ja kineettistä parametria kohti lasketaan mediaani, jota käytetään virhekorjatun estimaatin laskemiseen estimointivirheen varianssi vähentämällä. Vastaavasti estimointivirheiden varianssit määritetään ottamalla mediaani kahdeksalle homogeeniselle datalle suoritetuista ML-estimoinneista. Kuvaajassa 5.5 on visualisoituna näistä yhteensä viidestätoista erillisajosta saadut varianssikäyrät parametrille K_1 alueella 1.

Koe suoritetaan samoilla datoilla muutamalla eri optimointimenetelmällä, jotta saataisiin informaatiota myös siitä kuinka eri tavoin alueittainen heterogeenisuusestimaatti kehittyy eri menetelmiä käyttäen optimoinnin edetessä. Menetelmiksi valit-



Kuva 5.5: *Fletcher-Reeves -algoritmilla suoritetun ML-estimoinnin tuottamat varianssien kehittymiskäyrät heterogeenisten (punainen) ja homogeenisten (sininen) simuloitujen mittausdatojen yhteydessä 3K-mallin kineettiselle parametrille K_1 alueella* 1.

tiin perusmuotoinen Barzilai-Borwein, molempia modifiointeja hyödyntävä Barzilai-Borwein sekä vakioaskelpituutta käyttävä Fletcher-Reeves.

Kokeessa käytetyn fantomin koko on 128×128, binien määrä 367 ja hyötysuhde $sc = 0.05$. Datan generoinnissa käytettiin taulukossa 5.1 ilmoitettuja kineettisten parametrien alueellisia keskiarvoja ja -hajontoja. Fletcher-Reevesin vakioaskelpituus on $1e8$ ja modifiointeja hyödyntävän BB:n ennustekerroin oli 3, sovituksessa käytetyn parametrien kehityshistorian pituus 50 ja asettumisiteraatioiden määrä 25. Useaa askelpituutta laskettaessa osajoukkojakona käytettiin parametrityyppi- ja aluekohtaista jakoa. Kaikki optimoinnit aloitettiin oletusalkuarvauksesta. Optimointiiteraatioita suoritettiin Fletcher-Reevesillä $5.8 \cdot 10^4$, Barzilai-Borweinilla $1.3 \cdot 10^5$ ja modifioidulla Barzilai-Borweinilla $3.2 \cdot 10^4$.

Kokeen tulokset

Suuremmasta optimointi-iteraatiomäärästä huolimatta perusmuotoinen Barzilai-Borwein on saavutetulla kohdefunktion arvolla tarkasteltuna vielä huomattavan kaukana kahdesta muusta vertailtavasta algoritmista ja siten myös ML-estimaatista.

Tästä syystä johtuen ML-estimointivirheen poistamisen onnistumista kommentoidaan vain Fletcher-Reevesin ja modifioidun Barilai-Borweinin tuloksien perusteella.

Kokeen tuloskuvaajat B.7, B.9 ja B.8 eri optimointimenetelmiä käyttäen löytyvät liitteestä. Tuloksista havaitaan, että sekä virhekorjaamaton että -korjattu heterogeenisuusestimaati alkaa vakioitua suhteellisen säännönmukaisella tavalla estimaattia vastaavan kohdefunktion arvon pienetessä, kun optimointi-iteraatiomäärä kasvaa. Tätä havaintoa on mahdollista hyödyntää esimerkiksi automaattisessa estimoinnin lopetuskohdan määrittämisessä asettamalla lopettamisen ehdoksi tietyn vakioitumiskriteerin.

Kokeen tuloksista havaitaan myös, että virhekorjattu heterogeenisuusestimaatti vakioituu kaikissa tapauksissa huomattavasti lähemmäksi todellista heterogeenisuutta kuin virhekorjaamaton estimaatti. Todellisen heterogeenisuuden ja virhekorjatun estimaatin välille tilastollisesti jäävä virhe johtuu ML-estimaatista jäämisen lisäksi kohinan ja muiden virhelähteiden vaikutuksesta eikä ML-estimaatista jäämisen osuutta ole mahdollista erotella tästä kokonaisvirheestä. Koska kokonaisvirhe havaitaan kokeen tuloksissa verrattaen pieneksi niin myös ML-estimaatista jäämisen vaikutus muodostuu enimmillään verrattaen pieneksi.

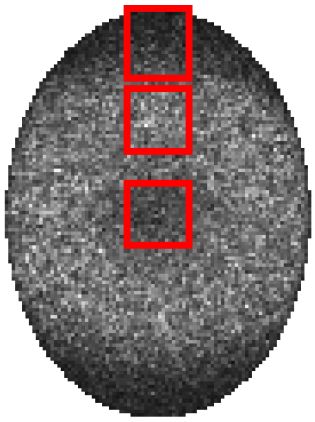
Fletcher-Reevesin ja modifioidun Barzilai-Borweinin tuottamat virhekorjatut heterogeenisuusestimaatit havaitaan lähes identtisiksi. Koska on todennäköistä, että eri menetelmät konvergoituvat kohti eri minimejä ML-estimaatin lähistöllä niin voidaan todeta, että näissä minimeissä heterogeenisuudet ovat todennäköisesti lähellä toisiaan.

5.5 Koe 4: Sijainnin vaikutus estimointivirheeseen

Tässä kokeessa tutkitaan kohteen sisäisen sijainnin mahdollista vaikutusta muodostuvaan estimointivirheeseen. Hypoteesina kokeessa on, että kohteen reuna-alueille syntyy vähemmän estimointivirhettä, koska reuna-alueiden yli kulkevien vastejanojen sisälle jää vähemmän vokseleita ja siten mitatut hajoamistapahtumat pystyttäisiin paikallistamaan paremmin.

Muodostuvan estimointivirheen suuruus on riippuvainen kohdekudoksen ominaisuuksista merkkiaineen suhteen eli simuloidun mittausdatan tapauksessa datan generoinnissa käytettyjen kineettisten parametrien arvoista. Jotta muodostuvan absoluuttisen estimointivirheen tilastolliset erot saatiin aiheutumaan vain sijainnista johtuen kohteen kaikki kolme aluetta yhdistettiin yhdeksi isoksi, ominaisuuksiltaan identtiseksi homogeeniseksi alueeksi. Samoilla asetuksilla generoitiin 15 tällaista homogeenista dataa ja suoritettiin niille ML-estimoinnit. Näistä toistoista laskettiin kineettisille parametreille vokselikohtaiset keskimääräiset absoluuttiset virheet.

Koe suoritettiin yksinkertaisempaa 2K-mallia käyttäen, sillä mallista riippumatta keskimääräiseen absoluuttiseen virheeseen muodostuu sijainnista aiheutuvaa virhet-



Kuva 5.6: *Neliöalueiden sijainnit fantomin suhteen on visualisoitu kuvan päälle, joka ilmaisee K_1:n keskimääräisen absoluuttisen virheen laskettuna ajoista, joissa binien määrä oli* 184 *ja sc* 0.004*. Ylintä neliötä kutsutaan reunaneliöksi, keskimmäistä rengasneliöksi ja alimmaista, fantomin keskellä sijaitsevaa neliötä keskineliöksi. Punaiset reunukset eivät ole mukana neliöalueissa, vaan ne rajaavat* 16×16 *vokselin suuruiset alueet.*

tä, mikäli tällaista virhettä ylipäätään syntyy. 2K-mallin etuna 3K-malliin verrattuna on myös laskenta-ajallinen hyöty sekä kokeen 1 tulosten perusteella suurempi varmuus ML-estimaatin saavuttamisesta.

Kokeessa käytetyn fantomin koko oli 128×128 ja koe suoritettiin kahta eri binien määrän ja hyötysuhteen *sc* yhdistelmää käyttäen. Datan generoinnissa käytettiin koko kohteen laajuiselle homogeeniselle alueelle taulukossa 5.1 ilmoitettuja kineettisten parametrien K_1 ja k_2 keskiarvoja alueella 1. ML-estimointi suoritettiin kaikissa tapauksissa Fletcher-Reeves -algoritmia ja vakioaskelpituutta $1e6$ käyttäen. ML-estimointia jatkettiin erillisajoille niin kauan kunnes ei oltu enää tuhanteen iteraatioon löydetty parannusta kohdefunktion arvoon tuottavaa ratkaisua. Ei kuitenkaan voida sanoa, että lopetuskohdassa oltaisiin saavutettu täsmällinen ML-estimaatti, vaan ollaan mahdollisesti esimerkiksi jumiuduttu tilanteeseen, jossa jäädään tekemään toistuvia edestakaisia gradienttiaskeleita globaalin minimikohdan lähiympäristössä tai ollaan saavutettu laskennallisen tarkkuuden raja.

Kokeen tulokset

Kun binien määrä oli 185 ja hyötysuhde *sc* 0.004 saatiin tulokseksi liitteen kuvan B.10 mukaiset parametrityyppikohtaiset keskimääräiset absoluuttiset virheet kohteen vokseleille ML-estimoinnin eri vaiheissa. Tuloksista havaitaan, että hypoteesin vastaisesti suurin virhe ei muodostu kohteen keskelle ja pienene kohteen reunojen suuntaan, vaan muodostuu rengasmaisesti kohteen keskikohdan ympärille ja pienenee reunoja ja keskikohtaa kohti. Toinen hypoteesin vastainen havainto on se, että

	binimäärä 185, *sc* 0.004		binimäärä 367, *sc* 0.05	
	keskiarvo	mediaani	keskiarvo	mediaani
K_1, reunaneliö	0.6167	0.5619	0.1262	0.1225
K_1, rengasneliö	1.3971	1.3531	0.1286	0.1259
K_1, keskineliö	0.9053	0.8784	0.1332	0.1334
k_2, reunaneliö	0.3402	0.2978	0.1185	0.1151
k_2, rengasneliö	0.5084	0.4821	0.1183	0.1146
k_2, keskineliö	0.3612	0.3470	0.1210	0.1193

Taulukko 5.4: *Kuvassa 5.6 esitellyiltä neliöalueilta lasketut keskiarvot ja mediaanit absoluuttisten virheiden keskiarvoista suhteessa todellisiin parametriarvoihin.*

suuremman virheen alue yltää kohteen vasemmassa ja oikeassa reunassa käytännössä reunalle asti toisin kuin kohteen ylä- ja alareunassa.

Se, että rengas muistuttaa muodoltaan enemmän ympyrän kehää kuin kohteen muotoisen soikion kehää voisi viitata siihen, että virhe on diskreetin Radon-muunnoksen toteuttavan Matlab-ohjelmiston funktion 'radon' aiheuttama. Funktion sisäistä toteutusta tuntematta on hyvin vaikea sanoa täsmällisemmin mikä virheen voisi aiheuttaa.

Koe toistettiin muuten identtisesti, mutta binien määrä nostettiin 367:ään ja hyötysuhde *sc* 0.05:een. Visualisoidut tulokset löytyvät liitteen kuvasta B.11. Tällä kertaa sijainnista ei muodostu silmämääräisesti havaittavaa virhettä. Luotettavamman arvion saamiseksi fantomista määriteltiin kolme, kuvassa 5.6 esiteltyä neliön muotoista aluetta, joiden sisältä laskettiin keskiarvo ja mediaani erillisajoista määritetyistä vokseleiden parametriestimaattien absoluuttisten virheiden keskiarvoista suhteessa parametrien todellisiin arvoihin. Nämä suhdeluvut on ilmoitettu taulukossa 5.4.

Taulukoitujen suhdelukujen perusteella todetaan, että suurempaa binimäärää ja hyötysuhdetta käyttäen havaittavaa sijainnista johtuvaa virhettä ei muodostu, vaikka käytetty kulmien määrä on ollut sama. On mahdollista, että tiettyä rajaa alempi binimäärä suhteessa fantomin dimensioihin aiheuttaa toistaiseksi määrittelemättömästä syystä johtuen 'radon'-funktion muodostamaan sijainnista riippuvaa virhettä. Havaittuun ilmiöön vaikuttaa todennäköisesti myös ns. näytteistysvirhe, joka voidaan nähdä riittämättömänä informaatiomääränä, ja josta seuraa se, että ongelma on lähellä olla tai on alimääräytynyt. On mahdollista, että näiden virhelähteiden yhteisvaikutus muodostaa havaitun alueellisesti varioivan virheen.

Tässä kokeessa havaittu sijainnista riippuvainen virhe on hyvä tiedostaa jatkotutkimuksissa etenkin mikäli pyritään toimimaan pieniä binimääriä käyttäen. Todellisissa käyttökohteissa pyritään mahdollisimman hyvään resoluutioon suuria bini- ja kulmamääriä käyttäen jolloin tästä ei muodostu ongelmaa.

6. YHTEENVETO

Tässä diplomityössä pyrittiin kehittämään uuden mallintamiseen perustuvan, kudoksen heterogeenisuuden suoraan PET-sinogrammista estimoivan menetelmän käytännön toteutusta ja tutkimaan menetelmän toimivuutta simulaatioiden avulla tietyissä koeasetelmissa. Merkittävin laskenta-aikaa kuluttava vaihe menetelmän toteutuksessa on mallin parametreille suoritettava suurimman uskottavuuden estimointi, joka teoriassa edellyttää suuridimensioisen optimointiongelman täydellisen ratkaisemisen, jotta löydettäisiin suurimman uskottavuuden estimaatti.

Tämän diplomityön kokeiden tuloksista havaitaan, että tähän asti ML-estimoinnin suorittamiseen oletusarvoisesti käytetyn Barzilai-Borwein -optimointimenetelmän suorituskykyä on mahdollista parantaa huomattavasti kahdella verrattaen yksinkertaisella, tätä diplomityötä varten kehitetyllä modifioinnilla. Lisäksi tehdään merkittävä havainto, että yksinkertainenkin konjugaattisuuntaa käyttävä optimointimenetelmä, poikkeuksellisesti vakioaskelpituutta käyttävä Fletcher-Reeves, menestyy erittäin hyvin suhteessa Barzilai-Borweiniin ja vakioaskelpituutta käyttävään jyrkimmän laskun menetelmään.

Fletcher-Reevesin yllättävänkin hyvään suorituskykyyn perustuen myös muita konjugaattisuuntaa käyttäviä algoritmeja on tulevaisuudessa syytä toteuttaa ja vertailla ML-estimoinnin suorittamiseksi. Fletcher-Reevesille vaihtoehtoisia, erilaisen määritelmän konjugaattisuunnan painolle esitteleviä algoritmeja ovat esimerkiksi Hestenes-Stiefel [22] ja Polak-Ribière [25]. Kuten Fletcher-Reeves -algoritmikin nämä menetelmät perustuvat optimaalisen askelpituuden käyttämiseen jokaisella iteraatiolla. Tästä huolimatta myös epäideaalisemman askelvalinnan käyttäminen voi joissain tilanteissa olla käytetyn laskenta-ajan huomioivan suorituskyvyn kannalta parempi vaihtoehto, sillä ideaalisen askeleen määrittäminen tuo mukanaan oman optimointiongelmansa.

Konjugaattimenetelmien kanssa käytettävää askelvalintaa onkin hyvä tutkia jatkotutkimuksissa, sillä ei ole mitään takeita siitä, että vakioaskeleen käyttäminen olisi paras tai edes hyvä ratkaisu. Voitaisiin esimerkiksi toteuttaa jonkinlaista logiikkaa käyttävä askelvalitsin, joka kokeilee jokaisella iteraatiolla useaa eri askelpituutta ja valitsee näistä aina parhaimman. Tällaisia epäideaaliin askelpituuteen tyytyviä ja sen jollain logiikalla määrittäviä konjugaattimenetelmiä on jo ennestään kehitetty lukuisia joten hyvä lähtökohta olisi seuraavaksi kokeilla tällaisten algoritmien vä-

lisissä vertailuissa hyvin menestynyttä, Hagerin ja Zhangin optimointimenetelmää [26].

Suoritetuissa kokeissa havaittiin myös, että ML-estimoinnin keskeyttäminen tietyin edellytyksin ennen globaalin optimikohdan saavuttamista ei todennäköisesti tuo merkittävää lisävirhettä heterogeenisuusestimaatteihin. Edellytyksenä tälle on vähintäänkin se, että löydetään ratkaisu, jonka voidaan nähdä sijaitsevan lähellä globaalia optimikohtaa. Tämä on hyödyllinen tieto, sillä kokeissa havaittiin, että 3K-lokeromallia käytettäessä uskottavuusfunktio on todennäköisesti multimodaalinen eikä siten gradienttimenetelmää käyttäen voida olettaa, että ML-estimaattia koskaan saavutettaisiin.

Laskenta-ajallisesti arvioiden tässä työssä tarkasteltu, 3K-mallia käyttävä heterogeenisuuden estimointimenetelmä on tällä hetkellä tämän diplomityön kokeiden havainnot huomioiden yhä liian hidas todellisissa kliinisissä tutkimuksissa käytettäväksi, kun parametristen kuvien resoluutiona käytetään kuvakokoa 128×128. Kokeen 3 heterogeenisuusestimaattien tuottamiseksi vakioaskelta käyttäneen Fletcher-Reevesin tarvitsema laskenta-aika oli noin yksi viikko nykyhetken keskitehokkaita tietokoneita (IBM System x3550) erillisajoissa hyödyntäen. On lisäksi odotettavissa, että parametristen kuvien resoluutioksi tarvittaisiin ainakin 256×256, jotta heterogeenisuusestimaatit olisivat luotettavampia ja jotta voitaisiin analysoida pienempiä kudosalueita kohteen sisällä. Tällainen resoluution kasvattaminen edelleen hidastaa ML-estimoinnin suorittamista arviolta nelinkertaisesti.

Eniten potentiaalia menetelmän nopeuttamiseksi jää todennäköisesti yhä optimointimenetelmän valinnan ja hienosäätämisen suhteen. Menetelmän ohjelmakoodin sisältämän kaiken muun toiminnallisuuden toteutusta on optimoitu sekä tämän diplomityön teon yhteydessä että sen ulkopuolella, eikä siihen panostamisessa nähdä tällä hetkellä yhtä suurta nopeutuspotentiaalia kuin edellä mainitussa.

LÄHTEET

[1] Gonzalez, R. C. & Woods, R.E. Digital Image Processing, 3rd edition. New Jersey 2010, Pearson Prentice Hall. 976 s.

[2] Carson, R.E. & Lange, K. The EM parametric image reconstruction algorithm. Journal of the American Statistical Association 80(1985), s. 20-22.

[3] Pölönen, H., Niemi, J. & Ruotsalainen, U. Error-corrected estimation of regional kinetic parameter histograms directly from PET projections. Physics in Medicine and Biology 55(2010)24, s. 7573-7586.

[4] Niemi, J., Ruotsalainen, U., Saarinen, A. & Ruohonen, K. Stochastic dynamic model for estimation of rate constants and their variances from noisy and heterogeneous PET measurements. Bulletin of mathematical biology 69(2007)2, s. 585-604.

[5] Pölönen, H., Niemi, J., Pekkarinen, J. & Ruotsalainen, U. Direct Estimation of Regional Kinetic Rate Constant Distributions from PET Sinograms and Correction of Statistical Estimation Error Distributions. Proceedings of 2010 Nuclear Science Symposium and Medical Imaging Conference. Knoxville, Tennessee, USA 2010.

[6] Jan, J. Medical Image Processing, Reconstruction and Restoration. Concepts and Methods. Boca Raton 2006, CRC Press. 730 s.

[7] van den Hoff, J. Principles of quantitative positron emission tomography. Amino Acids 29(2005)4, s. 341-353.

[8] Kay, S.M. Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory, 1st edition. Englewood cliffs, New Jersey, 1993, Prentice Hall. 625 s.

[9] Vanderplaats, G.N. Numerical Optimization Techniques for Engineering Design, 3rd edition. 3. painos. 1999. 441 s.

[10] Chen, S. An Analysis of Locust Swarms on Large Scale Global Optimization Problems. In: Korb, K., Randall, M. & Hendtlass, T. Artificial Life: Borrowing from Biology. Berliini 2009, Springer. s. 211-220.

[11] Neri, F. & Tirronen, V. Recent advances in differential evolution: a survey and experimental analysis. Artificial Intelligence Review 33(2010)1, s. 61-106.

[12] Liang, J.J., Qin, A.K., Suganthan, P.M. & Baskar, S. Particle swarm optimization algorithms with novel learning strategies. IEEE International Conference on Systems, Man and Cybernetics, vol 4. 2004. s. 3659-3664.

[13] Tseng, L-Y. & Chen, C. Multiple Trajectory Search for Large Scale Global Optimization. IEEE World Congress on Computational Intelligence. Hong Kong 2008, IEEE. s. 3052-3059.

[14] Tseng, L-Y. & Chen, C. Multiple Trajectory Search for Unconstrained/Constrained Multi-Objective Optimization. IEEE Congress on Evolutionary Computation, CEC '09. Trondheim, Norja, 2009, IEEE. s. 1951-1958.

[15] Storn, R. & Price, K. Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization 11(1997)4, s. 341-359.

[16] Brest, J., Greiner, S., Boŝković, B., Mernik, M. & Ẑumer, V. Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems. IEEE Transactions on Evolutionary Computation. 10(2006)6, s. 646-657.

[17] Eberhart, R.C. & Kennedy, J. A New Optimizer Using Particle Swarm Theory. Proceedings of the Sixth International Symposium on Micro Machine and Human Science. Nagoya, Japani 1995. s. 39-43.

[18] Liang, J.J., Qin, A.K., Suganthan, P.N. & Baskar, S. Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions. IEEE Transactions on Evolutionary Computation 10(2006)3, s. 281-295.

[19] Liang, J.J. & Suganthan, P.N. Adaptive Comprehensive Learning Particle Swarm Optimizer with History Learning. In: Wang, T-D., Xiaodong, L., Chen, S-H., Wang, X., Abbass, H., Iba, H., Chen, G. & Yao, X. Simulated Evolution and Learning. Berliini 2006, Springer. s. 213-220.

[20] Barzilai, J. & Borwein, J. Two-point step size gradient methods. IMA Journal of Numerical Analysis 8(1988), s. 141-148.

[21] Fletcher, R. & Reeves, C.M. Function minimization by conjugate gradients. The Computer Journal 7(1964)2, s. 149-154.

[22] Hestenes, M. & Stiefel, E. Methods of Conjugate Gradients for Solving Linear Systems. Journal of Research of the National Bureau of Standards 49(1952)6, s. 409-436.

[23] Alexandrov, O. Illustration of conjugate gradient method, Wikipedia [WWW]. [viitattu 29.11.2010]. Saatavissa: http://en.wikipedia.org/wiki/File:Conjugate_gradient_illustration.svg.

[24] Pekkarinen, J., Pölönen, H. & Neri, F. Advanced Metaheuristic Approaches and Population Doping for a Novel Modeling-Based Method of Positron Emission Tomography Data Analysis. Proceedings of the Main European Event on Evolutionary Computation. Torino, Italia 2011.

[25] Polak, E. & Ribière, G. Note sur la convergence de directions conjugées. Rev. Francaise Informat Recherche Opertionelle, 3e Année 16(1969), s. 35-43.

[26] Hager, W.W. & Zhang, H. A New Conjugate Gradient Method with Guaranteed Descent and an Efficient Line Search. SIAM Journal on Optimization 16(2005)1, s. 170-192.

A. LIITE: MONILENTORATAHAUN LÄHISTÖHAKUMENETELMÄT

```
1.  if Parannus[i] == EPÄTOSI
2.      Hakurajat[i] = Hakurajat[i]/2
3.      if Hakurajat[i] < 10^-15
4.          Hakurajat[i] = (YRAJA − ARAJA) · 0.4
5.      end
6.  end
7.  Parannus[i] = EPÄTOSI
8.  for j = 1 to N
9.      X_i[j] = X_i[j] − Hakurajat[i]
10.     if X_i paras löydetty ratkaisu
11.         pisteet = pisteet + BONUS1
12.         päivitä paras löydetty ratkaisu
13.     else
14.         if X_i huonompi kuin ennen muutosta
15.             palauta X_i[j] alkuperäiseen arvoonsa
16.             X_i[j] = X_i[j] + 0.5 · Hakurajat[i]
17.             if X_i paras löydetty ratkaisu
18.                 pisteet = pisteet + BONUS1
19.                 päivitä paras löydetty ratkaisu
20.             end
21.             if X_i huonompi kuin ennen muutosta
22.                 palauta X_i[j] alkuperäiseen arvoonsa
23.             else
24.                 pisteet = pisteet + BONUS2
25.                 Parannus[i] = TOSI
26.             end
27.         else
28.             pisteet = pisteet + BONUS2
29.             Parannus[i] = TOSI
30.         end
31.     end
32. end
```

Algoritmi A.1. *MTS:n ensimmäisen lähistöhakumenetelmän (LS1) pseudokoodi.*

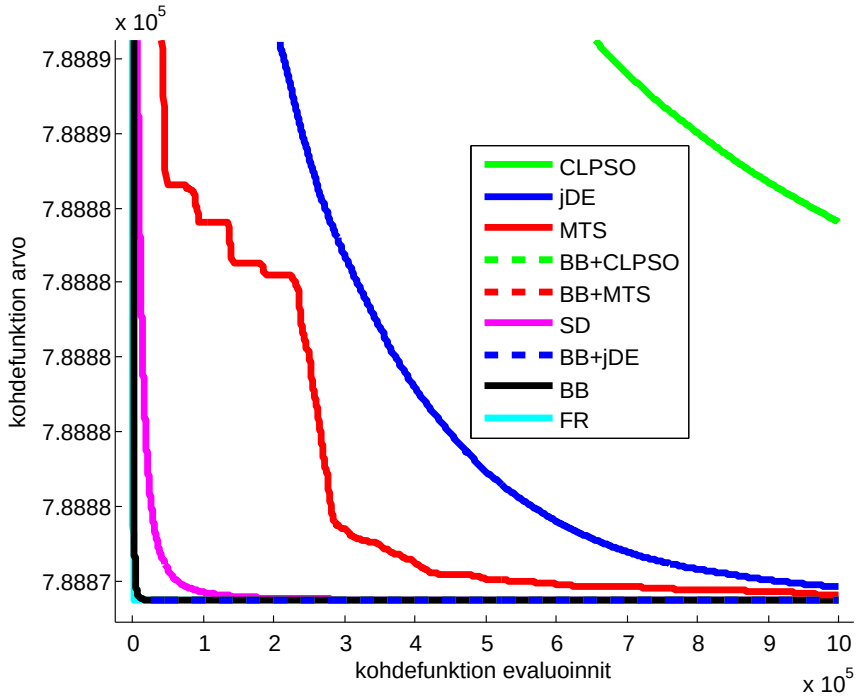
```
Rivit 1-7 kuten LS1:ssä
8.  for j = 1 to N
9.      for k = 1 to N
10.         r[k] = rand{0, 1, 2, 3}
11.         D[k] = rand{−1, 1}
12.         if r[k] == 0
13.             X_i[k] = X_i[k] − Hakurajat[i] · D[k]
14.         end
15.     end
16.     if X_i paras löydetty ratkaisu
17.         pisteet = pisteet + BONUS1
18.         päivitä paras löydetty ratkaisu
19.     else
20.         if X_i huonompi kuin ennen muutosta
21.             palauta X_i[j] alkuperäiseen arvoonsa
22.             for k = 1 to N
23.                 if r[k] == 0
24.                     X_i[k] = X_i[k] + 0.5 · Hakurajat[i] · D[k]
25.                 end
26.             end
27.             if X_i paras löydetty ratkaisu
28.                 pisteet = pisteet + BONUS1
29.                 päivitä paras löydetty ratkaisu
30.             end
31.             if X_i huonompi kuin ennen muutosta
32.                 palauta X_i[j] alkuperäiseen arvoonsa
33.             else
34.                 pisteet = pisteet + BONUS2
35.                 Parannus[i] = TOSI
36.             end
37.         else
38.             pisteet = pisteet + BONUS2
39.             Parannus[i] = TOSI
40.         end
41.     end
42. end
```

Algoritmi A.2. *MTS:n toisen lähistöhakumenetelmän (LS2) pseudokoodi.*

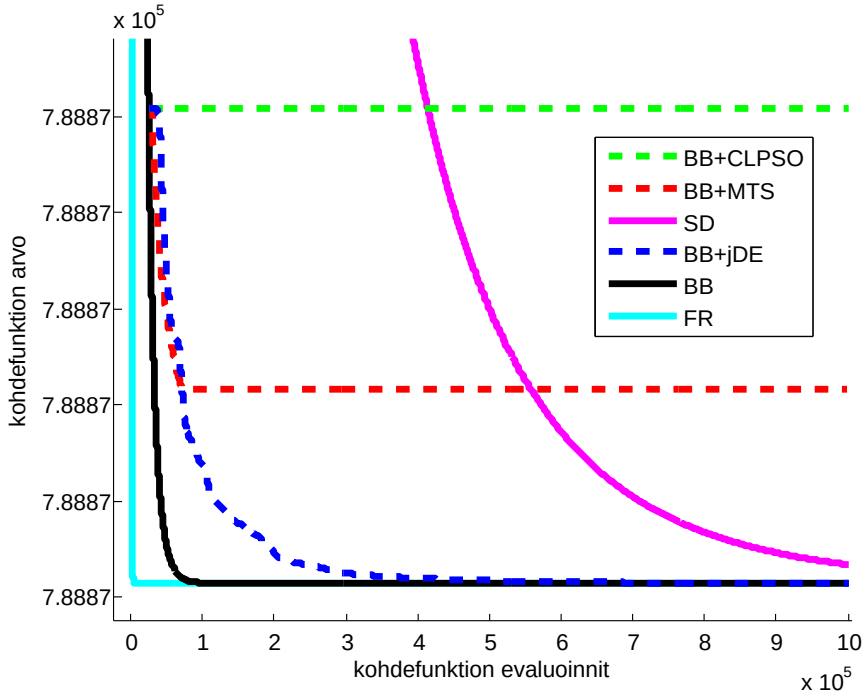
```
1.  for j = 1 to N
2.      X_1 = X_i, jonka k:s parametri +0.1
3.      Y_1 = X_i, jonka k:s parametri −0.1
4.      X_2 = X_i, jonka k:s parametri +0.2
5.      if X_1,Y_1 tai X_2 paras löydetty ratkaisu
6.          pisteet = pisteet + BONUS1
7.          päivitä paras löydetty ratkaisu
8.      end
9.      D_1 = f(X_i) − f(X_1)
10.     if D_1 > 0
11.         pisteet = pisteet + BONUS2
12.     end
13.     D_2 = f(X_i) − f(Y_1)
14.     if D_2 > 0
15.         pisteet = pisteet + BONUS2
16.     end
17.     D_3 = f(X_i) − f(X_2)
18.     if D_3 > 0
19.         pisteet = pisteet + BONUS2
20.     end
21.     a = rand[0.4, 0.5]
22.     b = rand[0.1, 0.3]
23.     c = rand[0, 1]
24.     X[j] = X[j] + a(D_1 − D_2) + b(D_3 − 2D_1) + c
25. end
26. if X_i ei parantunut LS3:n aikana
27.     palauta alkuperäinen X_i ennen LS3:a
28. else
29.     pisteet = pisteet + BONUS2
30. end
```

Algoritmi A.3. *MTS:n kolmannen lähistöhakumenetelmän (LS3) pseudokoodi.*

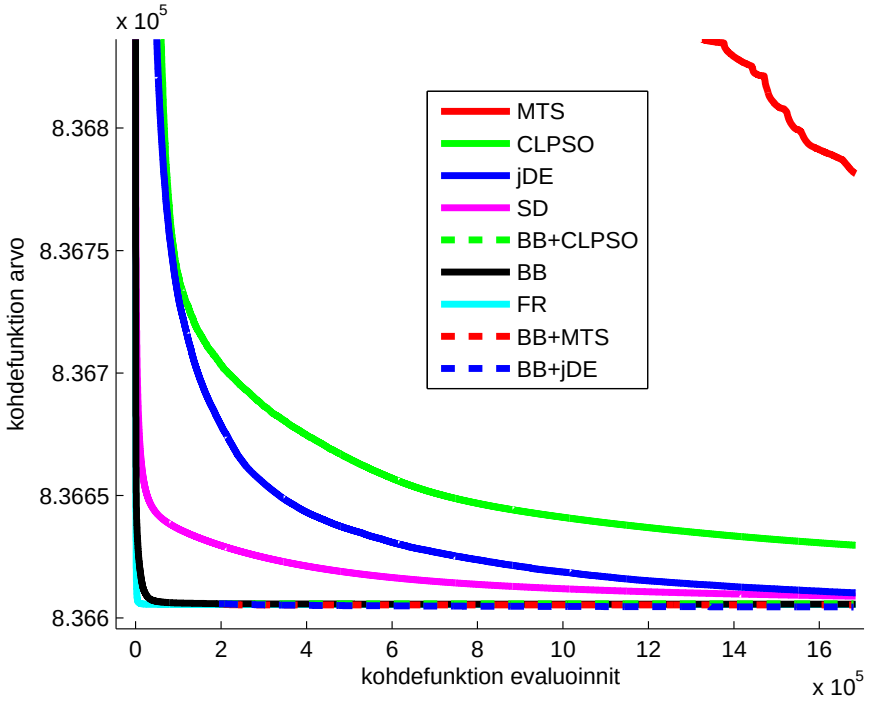
B. LIITE: KOKEIDEN TULOSKUVAAJAT



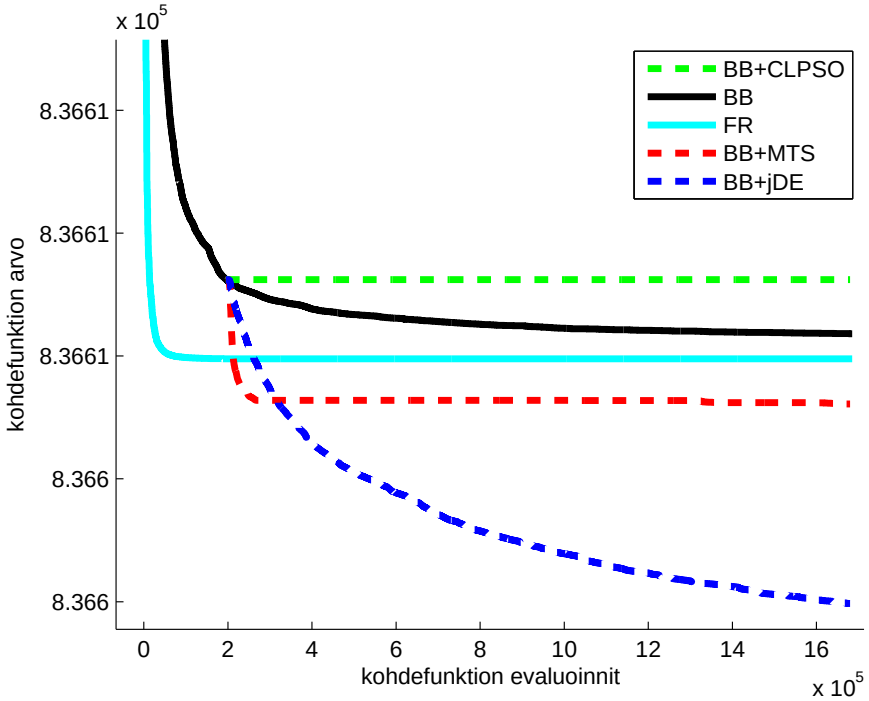
Kuva B.1: *Koe 1: Optimointialgoritmien erillisajoista lasketun keskimääräisen kohdefunktion arvon kehittyminen 2K-mallia käyttäen ja resoluutiolla* 16×16.



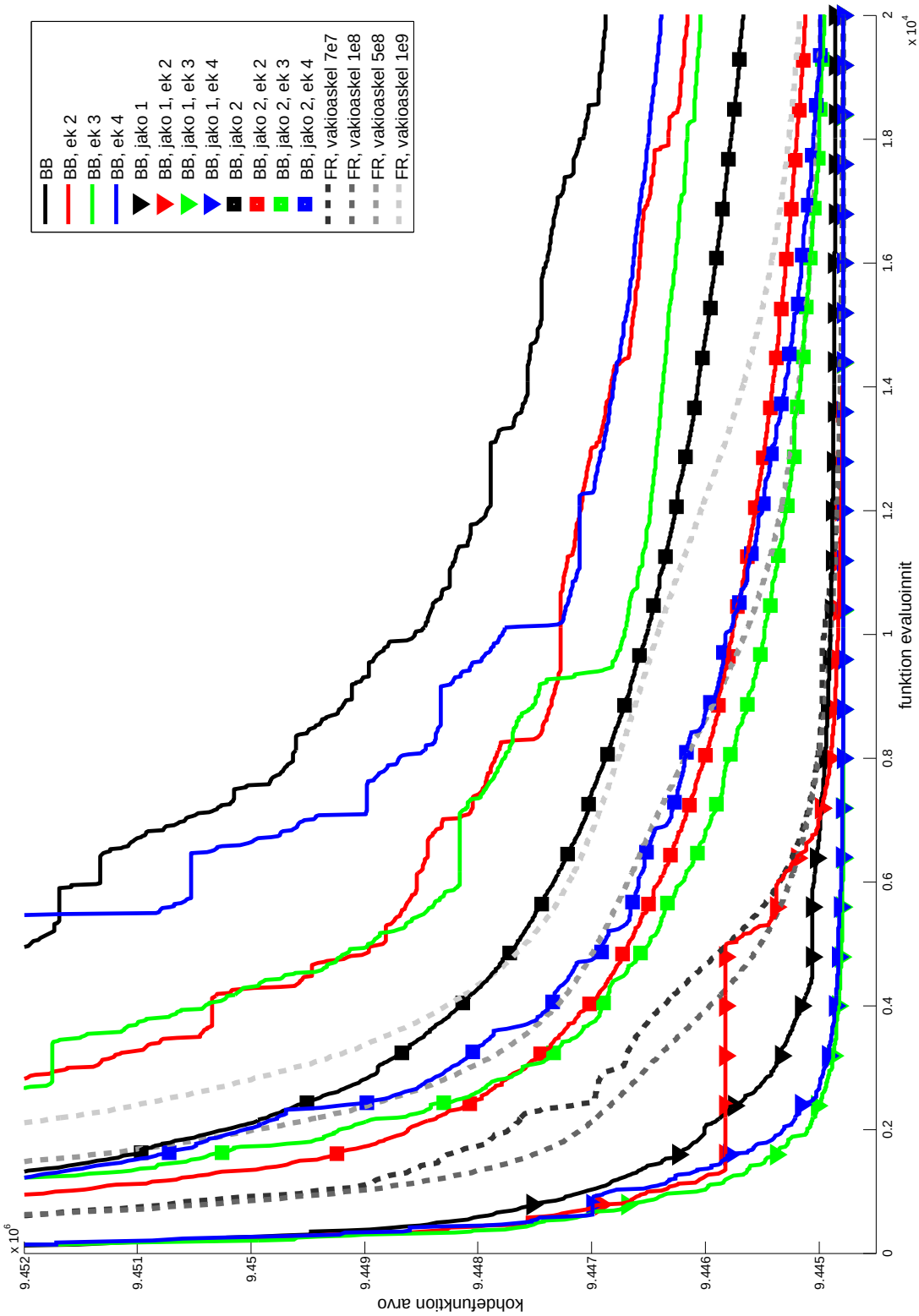
Kuva B.2: *Koe 1: Optimointialgoritmien erillisajoista lasketun keskimääräisen kohdefunktion arvon kehittyminen 2K-mallia käyttäen ja resoluutiolla* 16×16 *lähemmin tarkasteltuna.*



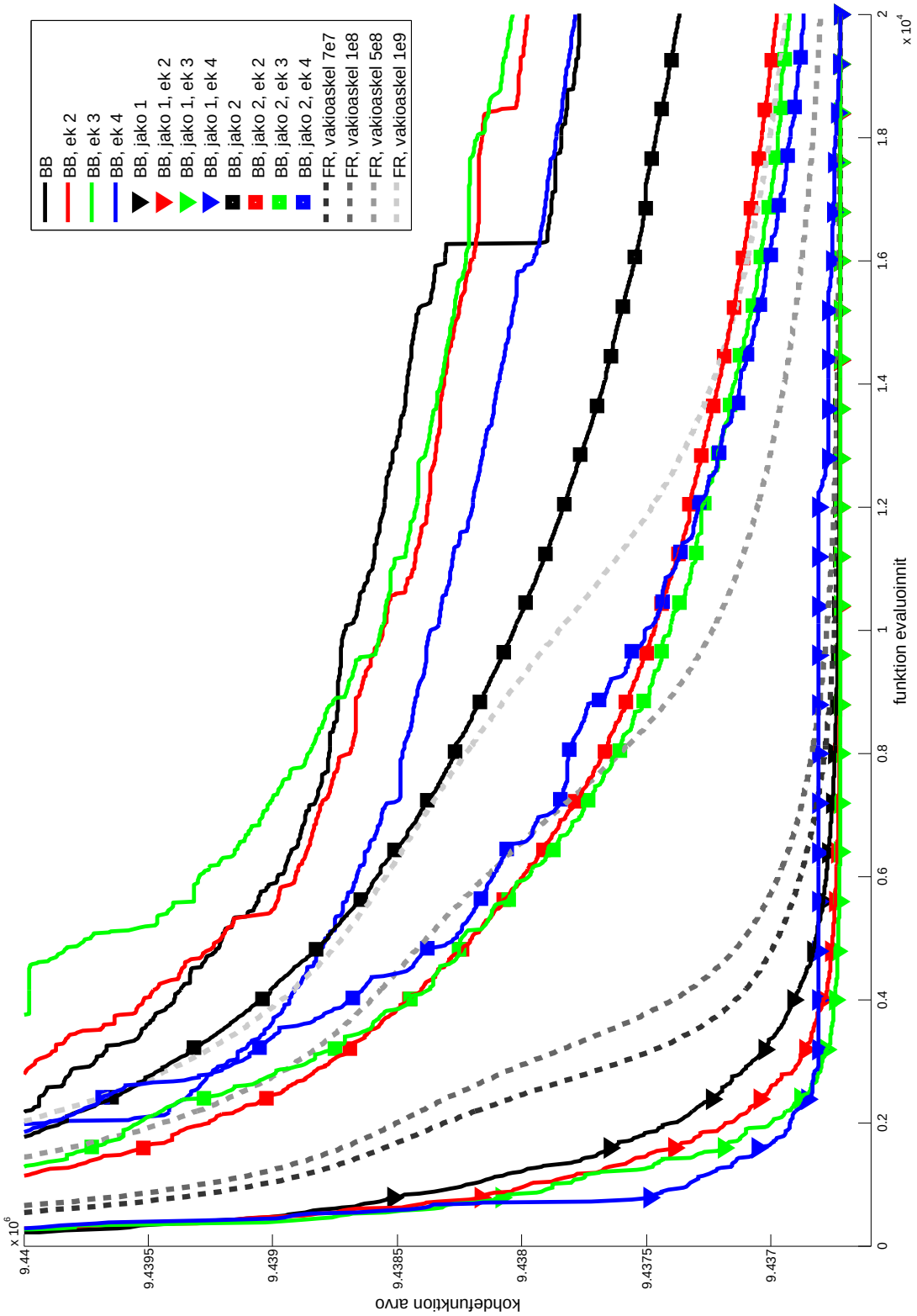
Kuva B.3: *Koe 1: Optimointialgoritmien erillisajoista lasketun keskimääräisen kohdefunktion arvon kehittyminen 3K-mallia käyttäen ja resoluutiolla* 16×16.



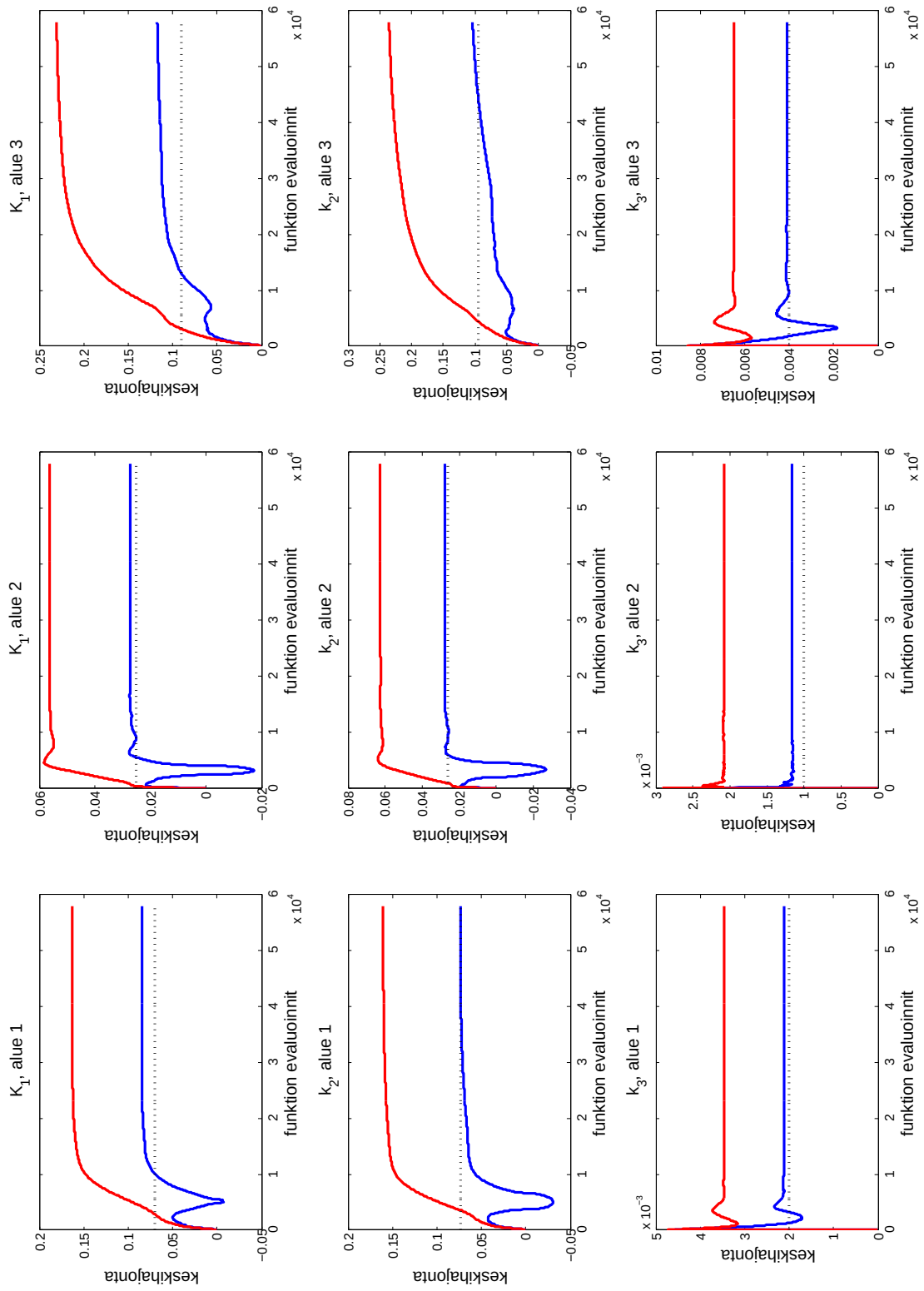
Kuva B.4: *Koe 1: Optimointialgoritmien erillisajoista lasketun keskimääräisen kohdefunktion arvon kehittyminen 3K-mallia käyttäen ja resoluutiolla* 16×16 *lähemmin tarkasteltuna.*



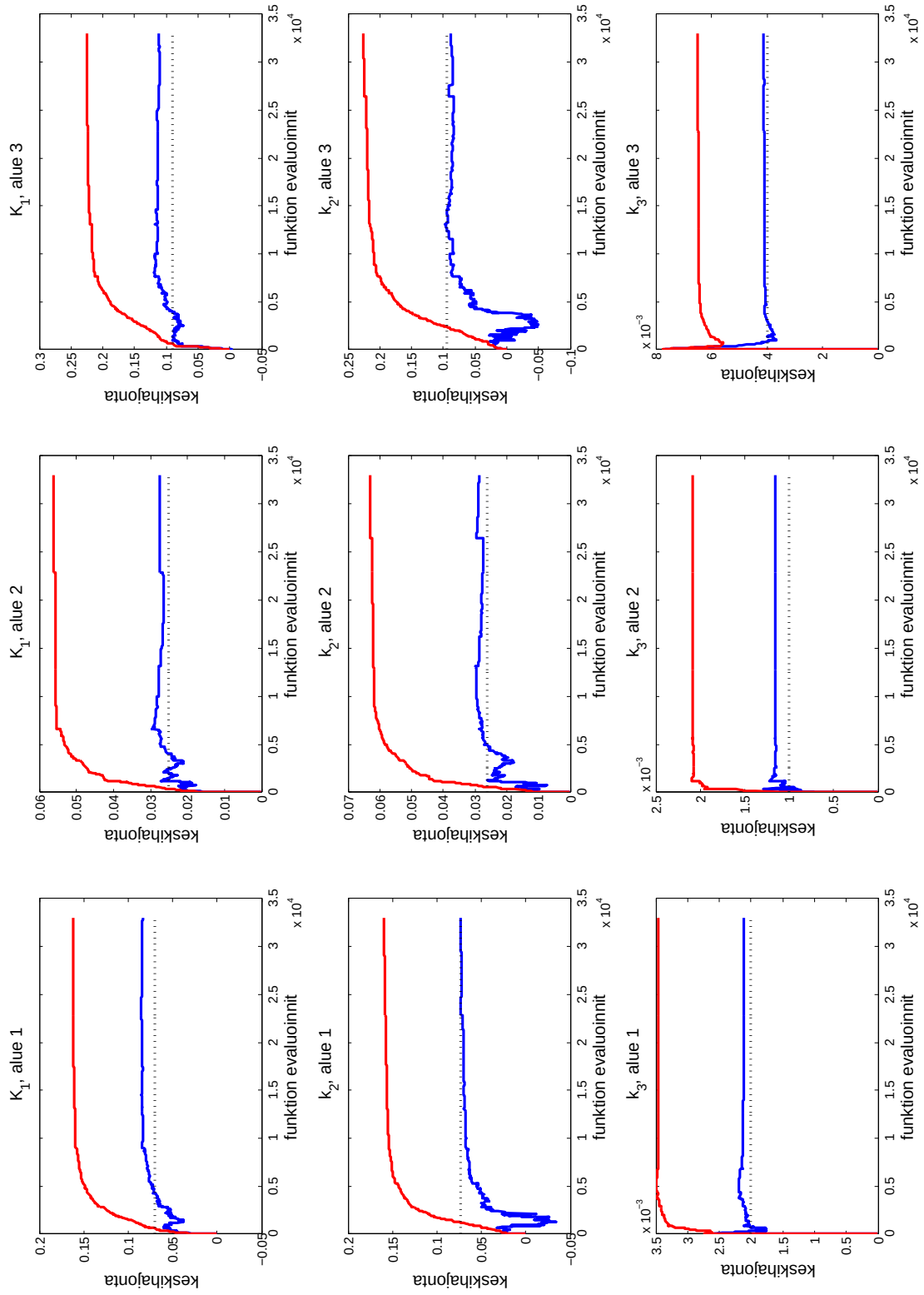
Kuva B.5: *Koe 2: Optimointialgoritmien vertailu heterogeenisen datan yhteydessä, kun käytetty lokeromalli on 3K ja resoluutio* 128×128.



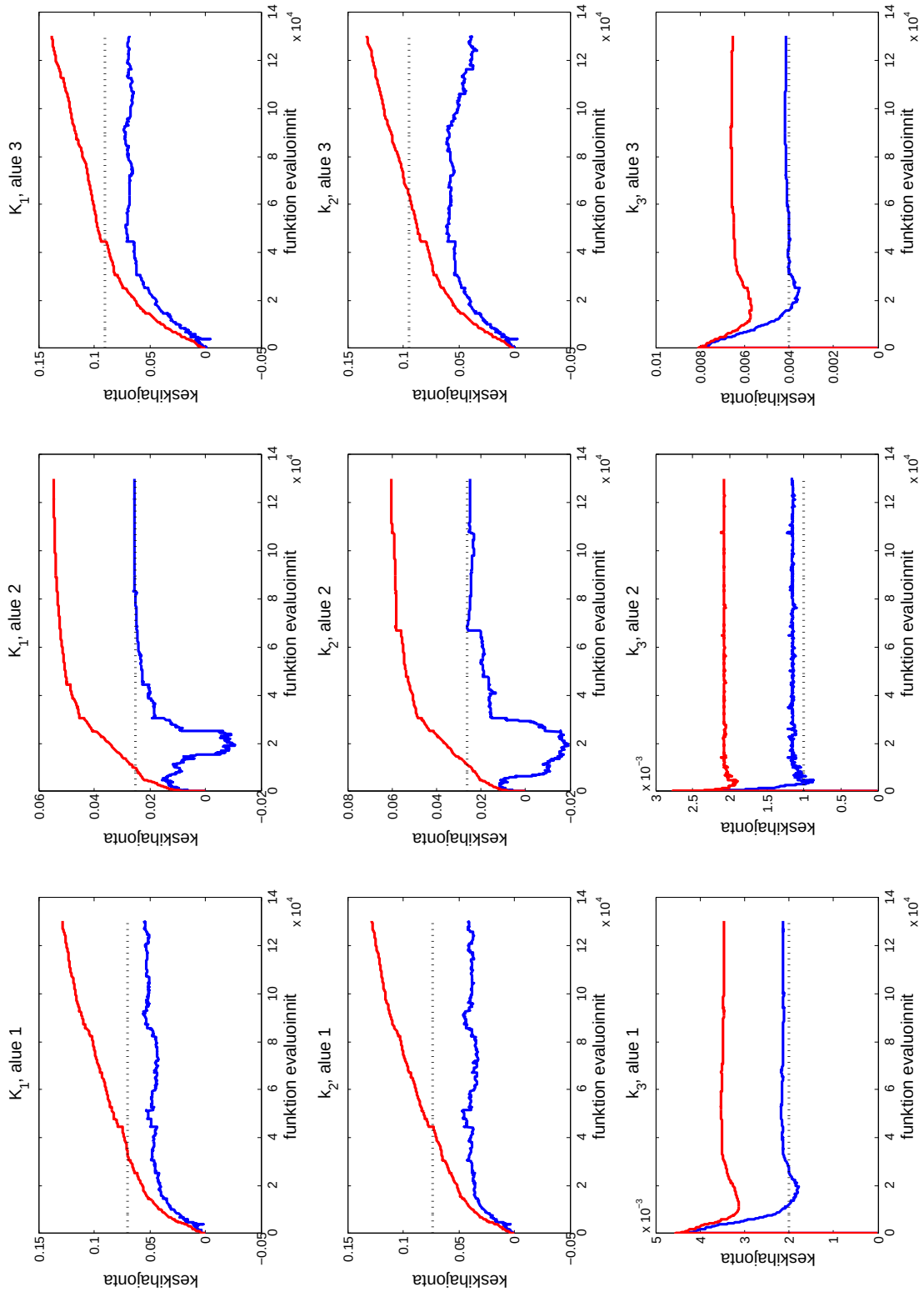
Kuva B.6: *Koe 2: Optimointialgoritmien vertailu homogeenisen datan yhteydessä, kun käytetty lokeromalli on 3K ja resoluutio* 128×128.



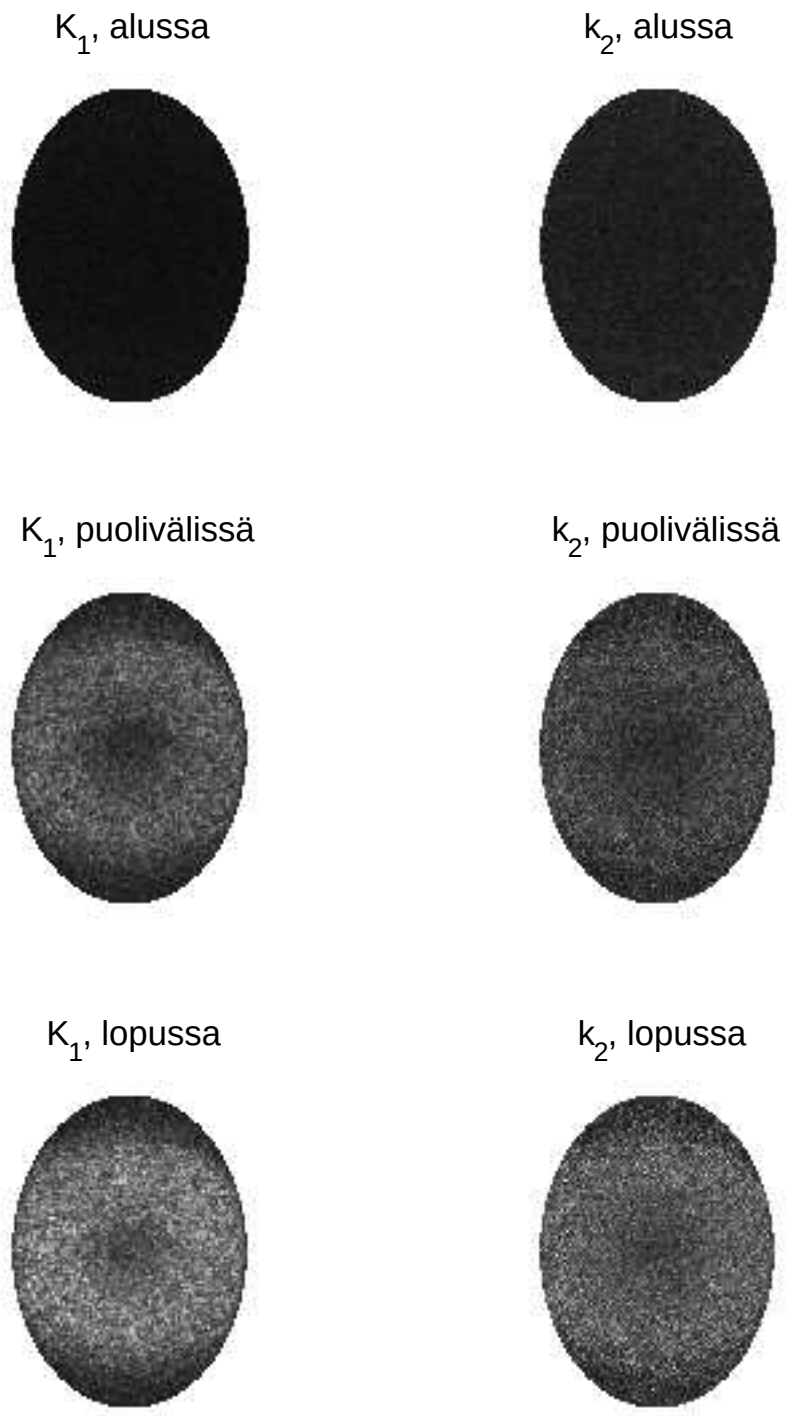
Kuva B.7: *Koe 3: Heterogeenisuusestimaatin (punainen) ja virhekorjatun heterogeenisuusestimaatin (sininen) kehittyminen optimointi-iteraatioiden suhteen, kun algoritmina on Fletcher-Reeves. Mustalla katkoviivalla on ilmaistuna simuloituun dataan asetettu todellinen heterogeenisuus. Käytetty lokeromalli on 3K ja resoluutio* 128×128.



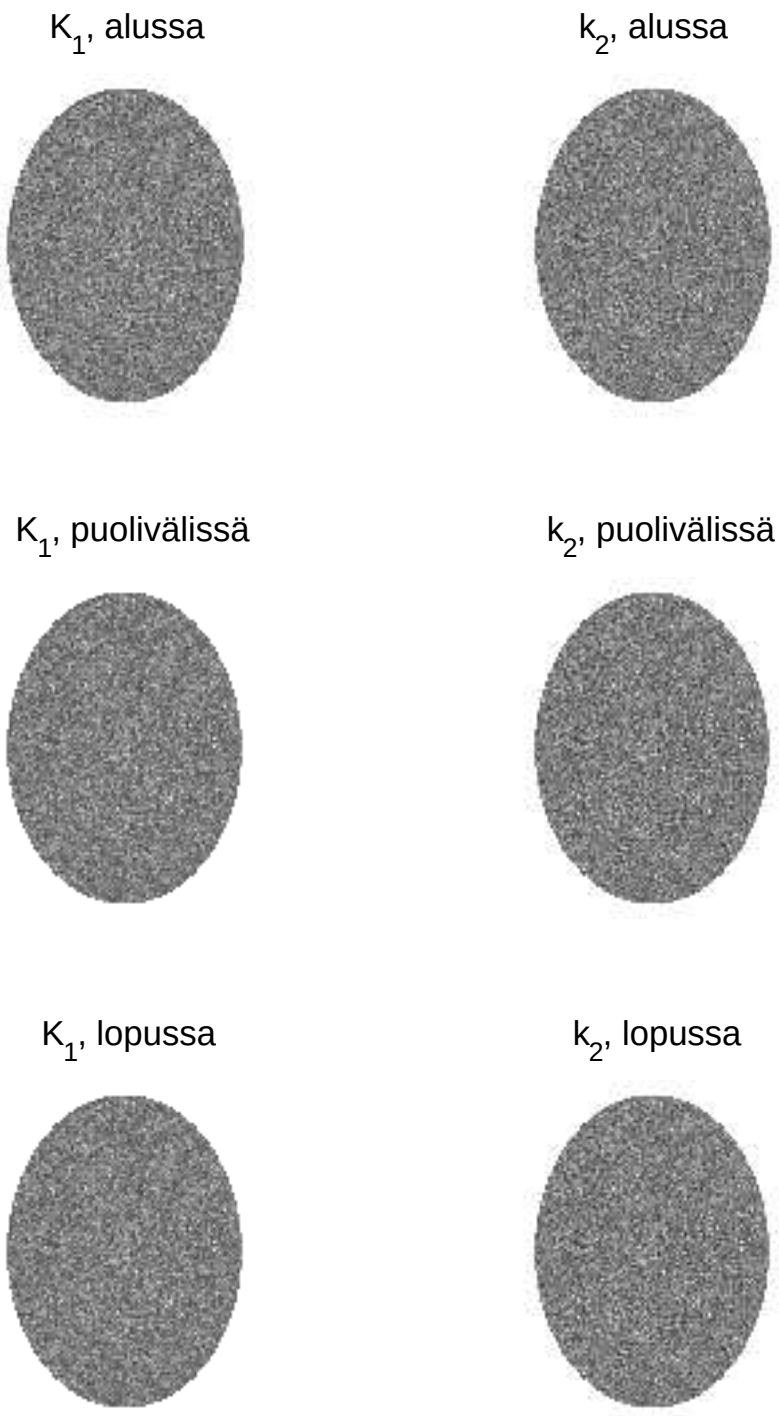
Kuva B.8: *Koe 3: Heterogeenisuusestimaatin (punainen) ja virhekorjatun heterogeenisuusestimaatin (sininen) kehittyminen optimointi-iteraatioiden suhteen, kun algoritmina on Barzilai-Borwein -modifiointi, joka käyttää alue- ja parametrityyppikohtaista osajoukkojakoa ja ensimmäisen asteen polynomilla ennustamista. Mustalla katkoviivalla on ilmaistuna simuloituun dataan asetettu todellinen heterogeenisuus. Käytetty lokeromalli on 3K ja resoluutio* 128×128.



Kuva B.9: *Koe 3: Heterogeenisuusestimaatin (punainen) ja virhekorjatun heterogeenisuusestimaatin (sininen) kehittyminen optimointi-iteraatioiden suhteen, kun algoritmina on Barzilai-Borwein. Mustalla katkoviivalla on ilmaistuna simuloituun dataan asetettu todellinen heterogeenisuus. Käytetty lokeromalli on 3K ja resoluutio* 128×128.



Kuva B.10: *Koe 5: Ylimpänä keskimääräiset absoluuttiset virheet tuhannen optimointi-iteraation jälkeen, keskimmäisenä erillisajojen puolivälistä laskettuna ja alimpana lopusta laskettuna. K_1:n ja k_2:n kaikki kolme tuloskuvaa on skaalattu jakamalla alkiot suurimmalla alkion arvolla, joka on löytynyt näistä kolmesta tuloskuvasta. Käytetty lokeromalli on 2K, resoluutio* 128×128*, binien määrä* 185 *ja hyötysuhde sc* 0.004.



Kuva B.11: *Koe 5: Ylimpänä keskimääräiset absoluuttiset virheet tuhannen optimointi-iteraation jälkeen, keskimmäisenä erillisajojen puolivälistä laskettuna ja alimpana lopusta laskettuna.* K_1*:n ja* k_2*:n kaikki kolme tuloskuvaa on skaalattu jakamalla alkiot suurimmalla alkion arvolla, joka on löytynyt näistä kolmesta tuloskuvasta. Käytetty lokeromalli on 2K, resoluutio* 128×128*, binien määrä* 367 *ja hyötysuhde sc* 0.05.