



**TURUN  
YLIOPISTO**

RADIOLAITTEIDEN PAIKANNUS USKOTTAVUUSESTIMOINNILLA  
LÄHIVERKOSSA

LuK Tarmo Taipale

Pro gradu -tutkielma  
Toukokuu 2023

Tarkastajat:  
Apulaisprofessori Jukka Lempa  
FT Kalle Ranto

MATEMATIIKAN JA TILASTOTIETEEN LAITOS

Turun yliopiston laatu­järjestelmän mukaisesti tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck-järjestelmällä

TURUN YLIOPISTO  
Matematiikan ja tilastotieteen laitos

TARMO TAIPALE: Radiolaitteiden paikannus uskottavuusestimoinnilla lähiverkos-  
sa

Pro gradu -tutkielma, 26 s., 0 liites.

Sovellettu matematiikka

Toukokuu 2023

---

Tämä sovelletun matematiikan työ esittelee tavan paikantaa radiolaitteita lähiverkosta signaalitehohavainnoista lasketun uskottavuusfunktion avulla. Työn alussa on katsaus eri paikannusmenetelmistä, ja katsauksen jälkeen työ keskittyy kokonaan signaalin tehohäviöön perustuvaan paikannukseen. Työssä määritellään tilastomalli signaalitehon käyttäytymiselle etäisyyden funktiona, johdetaan uskottavuusfunktio laitteiden paikoille sekä analysoidaan ympäristön ja satunnaistekijöiden vaikutuksia havaintoihin.

Uskottavuusfunktion johtamisen jälkeen käsitellään sen maksimointia paikkaestimaattien muodostamiseksi. Sitä varten työssä esitellään monen muuttujan funktiota optimoiva konjugaattigradienttialgoritmi ja rakennetaan sen matemaattista taustaa. Konjugaattigradienttialgoritmin avuksi käydään läpi myös yhden muuttujan funktiota optimoiva Brentin menetelmä ja sen lisäksi kultaisen leikkauksen menetelmä. Neljännessä luvussa kirjoitetaan auki täsmällinen algoritmi laitteiden paikantamiseksi, ja lopuksi kerrotaan algoritmin testaamiseksi ajettujen tietokonesimulaatioiden tulokset ja loppupäätelmät.

Asiasanat: radiolaitte, lähiverkko, signaaliteho, paikannus, uskottavuusfunktio, paikkaestimaatti, optimointi, konjugaattigradienttialgoritmi, Brentin menetelmä, kultaisen leikkauksen menetelmä.



# Sisällys

<b>1</b>	<b>Johdanto</b>	<b>1</b>
<b>2</b>	<b>Paikannus, etäisyysestimointi ja sijainnin uskottavuusfunktio</b>	<b>2</b>
2.1	Katsaus paikannusmenetelmistä . . . . .	2
2.1.1	Globaali paikannus . . . . .	2
2.1.2	Havaittuun aikaeroon perustuva paikannus . . . . .	2
2.1.3	Havaittuun suuntaan perustuva paikannus . . . . .	3
2.1.4	Tehohäviöön perustuva paikannus . . . . .	3
2.2	Tehohäviö satunnaismuuttujana . . . . .	4
2.3	Kolmioestimointi . . . . .	5
2.3.1	Bilateraatti . . . . .	5
2.3.2	Trilateraatti . . . . .	6
2.4	Uskottavuusfunktio . . . . .	7
<b>3</b>	<b>Uskottavuusfunktion maksimointi</b>	<b>10</b>
3.1	Konjugaattigradienttialgoritmi . . . . .	10
3.1.1	Konjugaattivektorien ortogonaalisuus ja konjugaattiominaisuu- det . . . . .	12
3.1.2	Parabolisen funktion ääriarvokohdan löytäminen . . . . .	14
3.2	Kohdefunktion maksimointi yhdessä ulottuvuudessa . . . . .	15
3.2.1	Kolmen pisteen kautta kulkevan paraabelin huippukohta . . .	15
3.2.2	Brentin menetelmä . . . . .	16
3.2.3	Kultaisen leikkauksen menetelmä . . . . .	17
3.2.4	Yhden ulottuvuuden maksimoinnin toteutus . . . . .	18
<b>4</b>	<b>Paikannus</b>	<b>20</b>
4.1	Alustus . . . . .	20
4.2	Tarkennus . . . . .	21
<b>5</b>	<b>Lopputulokset</b>	<b>22</b>
5.1	Simulointi . . . . .	22
5.2	Omat päätelmät . . . . .	25



# 1 Johdanto

Tämä työ käsittelee antennien paikantamista lähiverkossa signaalien tehotasoihin (RSSI, Received Signal Strength Index) perustuvia etäisyysestimaatteja käyttäen. Paikannus tehdään tässä työssä käyttäen hyväksi ankkurilaitteita, jotka tietävät oman sijaintinsa vaikkapa GPS:n antaman tiedon perusteella, ja lisäksi käytetään hyväksi sokkolaitteiden eli paikkaansa tietämättömien laitteiden välillä lähetettyjä signaaleja. Tällainen paikannus on hyödyksi esimerkiksi radiosysteemien reititys-algoritmien apuökaluna.

Etäisyysmittaus perustuu tässä paikannusmenetelmässä niin sanottuun Log Distance path loss -malliin eli etäisyystehohäviömalliin, jossa kuvataan signaalin logaritmisen tehon (joka mitataan desibeleissä, dB) laskua signaalin kulkeman matkan myötä. Malli ei kuvaa radiosignaalin käyttäytymistä kaikissa tilanteissa, erityisesti jos kahden laitteen välisen linkin sisällä on kahta erilaista ympäristöä (esimerkiksi toisen laitteen lähellä on vaikka metsää ja toisen laitteen lähellä esteetöntä tasankoa) eli ympäristö on heterogeenistä. Tehohäviömallia käytetään tässä työssä, koska se on yksinkertainen ja toimii riittävän hyvin varsinkin pienillä etäisyyksillä ja koska useista laitteista kasaantuvaa informaatiota pystytään tasoittamaan uskottavuusfunktiota laskettaessa informaation epätarkkuudesta huolimatta. Oletuksia ympäristöstä lisätään tämän työn kannalta niin, että se on koko verkossa samanlaista eli homogeenistä ja lisäksi verkon oletetaan tuntevan ympäristön niin hyvin, että laitteet tietävän ympäristön keskeiset parametrit. Mallissa on etäisyydestä johtuva tehohäviö sekä ympäristöstä johtuva joko signaalia vahvistava tai heikentävä tekijä, jota mallinnetaan normaalijakautuneella satunnaismuuttujalla. Etäisyyden aiheuttamaan häviöön liittyy lisäksi parametreja, kuten referenssietäisyyden tehohäviö ja ympäristöstä riippuva tehohäviöekspONENTTI, mutta ne jäävät tässä matemaattisessa tutkielmassa vähemmälle huomiolle ja ne oletetaan vakioiksi.

Näiden etäisyysmittausten perusteella pyritään sitten selvittämään verkon radiolaitteiden sijainnit toisiinsa nähden, eli sijoittamaan ne koordinaatistolle. Tämä tehdään suurimman uskottavuuden estimaatin avulla: kullekin laitteelle voidaan laskea kussakin mahdollisessa sijainnissa log-uskottavuusfunktio, joka on muotoa

$$l_i(x, y) = C \sum_{j \neq i} (\hat{L}_{i,j} - L_{i,j}(x, y, x_j, y_j))^2.$$

Tässä  $C$  on vakio,  $(x, y)$  postuloitu sijainti laitteelle  $i$ ,  $(x_i, y_i)$  vierekkäisten laitteiden senhetkiset sijaintiarviot,  $\hat{L}_{i,j}$  havaittu tehohäviö (ja samalla suurimman uskottavuuden estimaatti etäisyyden aiheuttamalle tehohäviölle), sekä  $L_{i,j}(x, y, x_j, y_j)$  etäisyyden osuus tehohäviöstä pisteiden  $(x, y)$  ja  $(x_j, y_j)$  välillä. Uskottavuusfunktion laskemista toistetaan laitteissa uudestaan aikaisempien suurimman uskottavuuden estimaattien pohjalta eli laitteet iteroivat itselleen uusia suurimman uskottavuuden estimaatteja jatkuvasti. Näin pyritään lähestymään globaalia uskottavuusmaksimia laitteiden sijainneille verkossa.

Paikannuksessa algoritmin suorituskykyä mitataan laskemalla keskineliövirhe algoritmin antamien arvioiden ja oikeiden sijaintien välillä. Onnistunut paikannus-algoritmi antaa verkon, jonka topologia vastaa mahdollisimman hyvin todellisuutta ja jossa metrimääräinen paikannusvirhe on mahdollisimman pieni.

## 2 Paikannus, etäisysestimointi ja sijainnin uskottavuusfunktio

### 2.1 Katsaus paikannusmenetelmistä

Elektronisten laitteiden paikannukseen on erilaisia fysikaalisia menetelmiä. Useimmissa niistä mitataan jotain laitteiden välillä lähetetystä signaalista etäisyyden estimoimiseksi, mutta jotkut paikannusmenetelmät käyttävät esimerkiksi yhteyden laatu paikan arvioimiseen. Tässä työssä keskitytään signaalitehohäviöön etäisysestimoinnin metodina, mutta tässä kappaleessa otetaan lyhyt katsaus muihinkin paikannusmenetelmiin. Paikannusmenetelmien katsaus perustuu pääosin lähteeseen [3].

#### 2.1.1 Globaali paikannus

Globaalit paikannusjärjestelmät, kuten GPS, käyttävät paikantamiseen saapumisajkoja. Siinä laitteet paikantavat itsensä vastaanottamalla signaaleja satelliiteilta, joita on kulloinkin vähintään 24. Satelliitit tietävät sijaintinsa Maan kiertoradalla sekä ajan suurella tarkkuudella, sillä niiden atomikellot on synkronoitu keskenään täsmällisesti, ja ne korjaavat toistensa välillä syntyviä virheitä säännöllisesti. Peruseriaatteena on, että itseään paikantava laite, jonka kello on epätarkempi, ottaa vastaan signaalin vähintään neljältä satelliitilta ottaen niiltä vastaan tiedot ajasta, jossa satelliitit lähettivät signaalinsa. Satelliittien lähettämät signaalit kulkevat valonnopeudella, jolloin lähetys- ja vastaanottoaikojen perusteella voidaan muodostaa arviot laitteen etäisyydestä kuhunkin satelliittiin. Laite voi myös korjata satelliittien kellojen perusteella omaa epätarkkaa aikaansa.

Käsitellään hieman GPS-paikannuksen matematiikkaa lähteen [4] tietojen perusteella. Oletetaan, että  $N$  satelliittia lähettää signaalin ajanhetkellä  $t$  ja paikannettava laite vastaanottaa satelliitilta numero  $i$  signaalin (oman kellonsa mukaan) ajanhetkellä  $t_i$  kullakin  $i$ . Olkoon satelliitin sijainti  $(x_i, y_i)$  kullakin  $i$ . Oletetaan satelliittien kellojen oletetaan olevan oikeassa ajassa, ja oletetaan, että paikannettavan laitteen kellossa on aikavirhe  $b$ . Jos signaalien nopeus on  $c$ , voidaan paikannettavan laitteen paikan  $(x, y)$  ja oikean ajan ratkaisemiseksi muodostaa yhtälöt

$$t_i = t + b + \frac{d_i}{c},$$

missä  $d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2}$  on paikannettavan laitteen ja satelliitin  $i$  välinen etäisyys. Jos satelliitteja on käytössä neljä näiden yhtälöiden muodostamiseksi, pystytään yhtälöryhmä ratkaisemaan, jolloin saadaan selville paikannettavan laitteen sijainti sekä aikavirhe. Useammalla kuin neljällä satelliitilla yhtälöryhmästä tulee ylimääritelty, jolloin voidaan hakea approksimatiivista ratkaisua vaikkapa pienimmän neliösumman menetelmällä.

#### 2.1.2 Havaittuun aikaeroon perustuva paikannus

Havaittu aikaero (Observed Time Difference Of Arrival, OTDOA), jota käytetään tietoliikenteen Long Time Evolution -standardissa, on aikaeroihin perustuva menetelmä, joka hyödyntää laitteiden paikannuksessa tukiasemia. Siinä tukiasemat lä-



hettävät referenssisignaaleja paikannettavalle laitteelle tiettyinä kellonaikoina, ja paikannettava laite mittaa erot eri tukiasemista lähteneiden signaalien vastaanottoajoissa. Tässä mielessä menetelmä muistuttaa GPS-paikannusta. Suurimmat erot OTDOA:n ja GPS-paikannuksen välillä ovatkin protokollatasolla: OTDOA:ssa erillinen paikannuskeskus lähettää signaaleja aktiivisesti paikannettaville laitteille, joissa niitä kehoitetaan paikantamaan itsensä ja lähettämään paikkatietonsa. Paikannettavat laitteet siis lähettävät signaaleja referensseinä toimiville tukiasemille poiketen GPS:stä, jossa laitteet eivät lähetä mitään satelliiteille. Sitten tukiasemista lähetetään referenssisignaaleja, jotta paikannettavissa laitteissa voitaisiin laskea aikaeroja ja muodostaa niistä etäisyysestimaatteja.

Aikaeroon perustuvan paikannuksen hyvä puoli on se, että etäisyysvirhe ei kasva laitteiden etäisyyden kasvaessa. Huonona puolena aikaeroon perustuva etäisyysestimointi vaatii tarkat ja hyvin synkronoidut kellot, joita useimmissa elektronisissa laitteissa ei ole, mikä tekee aikaerosta huonon etäisyysestimointimenetelmän tämän työn kannalta.

### 2.1.3 Havaittuun suuntaan perustuva paikannus

Havaittuun suuntaan (Angle of Arrival, AoA) perustuvassa paikannuksessa paikannettava laite mittaa, mistä suunnasta jokin toinen laite on lähettänyt signaalin. Tähän on monia menetelmiä, ja yleisimpiä niistä ovat esimerkiksi maksimitehosuunnan etsiminen antennia pyörittäessä, ja laitteen useamman antennin välisten aikatai vaihe-erojen mittaus. Suuntapaikannusta käytetään muunmuassa matkapuhelien sijainnin estimointiin [5].

Suuntaan perustuvassa paikannuksessa heikkoutena on se, että se vaatii erityistä laitteistoa tuekseen, vähintäänkin laitteissa on oltava useampi antenni, jotta niiden avulla voidaan arvioida signaalien tulosuuntia. Lisäksi esteisessä ympäristössä monipolkueteneminen (multipath propagation) voi johtaa signaalien kimpoamisiin, mikä voi haitata paikannusta. Tässä työssä ei oleteta suoraa näköyhteyttä tai esteettömyyttä laitteiden välillä, vaan ainoastaan ympäristön tunnettuus ja homogeenisyys, joten suuntaestimoinnin toimivuudesta ei ole takeita näissä rajoissa.

### 2.1.4 Tehohäviöön perustuva paikannus

Tässä työssä tutkittavassa signaalin tehoon (Received Signal Strength Index, RSSI) perustuvassa paikannuksessa mitataan kahden laitteen välillä lähetetyn signaalin tehohäviötä ja arvioidaan etäisyys siitä. Signaalin lähetys- ja vastaanottotehon erotuksen eli tehohäviön  $P$  (joka ilmaistaan logaritmisella asteikolla desibeleinä) mallinnetaan tässä menetelmässä noudattavan yhtälöä

$$P = P_0 + 10\gamma \lg\left(\frac{d}{d_0}\right) \quad (1)$$

ideaalitapauksessa, eli kun ympäristö tai laitteen ominaisuudet eivät aiheuta signaalitehossa satunnaisvirhettä. Yhtälössä  $d_0$  on mielivaltaisesti valittu referenssiepäisyys, esimerkiksi yksi metri,  $P_0$  tehohäviö referenssiepäisyydellä ja  $\gamma$  ympäristöstä

riippuva ns. tehohäviöeksponentti, joka kuvaa tehohäviön kasvun nopeutta etäisyyden kasvaessa. Tästä voidaan ratkaista laitteiden välinen etäisyys

$$d = d_0 10^{\frac{P-P_0}{10\gamma}}. \quad (2)$$

Tehohäviöön perustuva paikannus toimii hyvin ympäristössä, joissa on vähän esteitä, mutta esteisemmässä ympäristössä se toimii huonommin, sillä ympäristön satunnaistekijät voivat aiheuttaa etäisyysestimoinnissa virheitä, joiden mittaluokka on moninkertainen oikeaan etäisyyteen verrattuna. Sitä käytetään tässä työssä silti, koska se on yksinkertainen tapa, ja koska tehohäviöpaikannuksen virheet ovat usein pienempiä verkoissa, joissa välimatkat ovat lyhyitä. Verrokkina esimerkiksi aikaeroon perustuvassa paikannuksessa estimaatteihin vaikuttavat mm. etäisyydestä riippumattomat kellovirheet. Menetelmä ei vaadi myöskään kallista laitteistoa, ja työssä esiteltävän uskottavuusestimaattorin odotetaan pystyvän tasoittamaan epävarmaa mittaustietoa, kun sitä tulee useista laitteista.

## 2.2 Tehohäviö satunnaismuuttujana

Paikannus lähiverkossa perustuu tässä työssä laitteiden välisten etäisyyksien mittamiseen ja koordinaattien konstruoimiseen niistä. Etäisyysmittauksella ajatellaan olevan jokin todellisesta etäisyydestä ja muista, ympäristöä kuvaavista parametreista riippuva jakauma, ja tässä etäisyysmittauksista lasketaan (log-)uskottavuusfunktiot kunkin laitteen sijainneille. Näistä lasketaan uskottavuusmaksimit, joihin laitteiden estimoidaan sijoittuvan. Uskottavuusmaksimien laskentaa toistetaan eli iteroidaan, ja tällä tavalla pyritään lähestymään todellisia sijainteja. Menetelmässä käytettävä uskottavuusfunktio ja sen pohjana toimiva tehohäviömalli perustuvat lähteeseen [2], laskennan iterointimenetelmä taas on työn kirjoittajan kehittämä.

Etäisyysestimaatti perustuu tässä signaalien tehotasoihin (RSSI, Received Signal Strength Index), ja tehotason oletetaan määräytyvän seuraavan kaavan mukaisesti:

$$P = P_0 + 10\gamma \lg\left(\frac{d}{d_0}\right) + X \quad (3)$$

Muuttuja  $P$  on signaalista mitattu tehohäviö (dB),  $d_0$  mielivaltaisesti valittu referenssietäisyys (esimerkiksi yksi metri) ja  $P_0$  referenssietäisyyden odotustehohäviö, joka määritetään joko teoreettisesti tai kenttämittauksella. Muuttuja  $d$  taas on lähettimen ja vastaanottimen etäisyys toisistaan,  $\gamma$  tehohäviöeksponentti ja  $X$  ympäristöstä johtuva propagaatiovirhe, jota mallinnetaan normaalijakautuneena satunnaismuuttujana, jonka odotusarvo on 0 ja varianssi  $\sigma^2$ .

Tehohäviöeksponentin oletetaan tässä olevan vakio verkossa ja koko verkon tuntevan sen. Samoin parametrin  $P_0$  ja  $\sigma$  oletetaan tunnetuiksi vakioksi koko verkon yli. Nämä oletukset kuvaavat tilannetta jossa ympäristö on homogeeninen eli tilastollisesti samanlainen koko verkon laajuudelta (verkko voi olla esimerkiksi kauttaaltaan kaupunkiympäristön peittämä).

Kaavasta 3 voidaan ratkaista etäisyys, jolloin saadaan

$$d = d_0 10^{\frac{P-P_0-X}{10\gamma}}. \quad (4)$$

Odotusteho (desibeleinä) saadaan poistamalla kaavasta 3 satunnaismuuttuja  $X$ , jonka odotusarvo on siis nolla. Näin saadaan

$$P = P_0 + 10\gamma \lg\left(\frac{d}{d_0}\right). \quad (5)$$

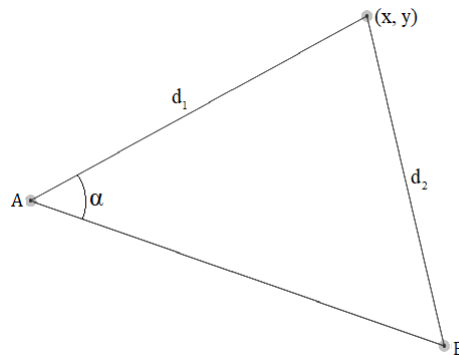
Uskottavuusfunktiota laskettaessa havaittua tehoa voidaan sitten verrata tähän odotustehoon, ja paikkaestimaatteja voidaan laskea yhdistämällä näitä erotuksia useista linkeistä.

## 2.3 Kolmioestimointi

### 2.3.1 Bilateraatti

Trigonometrinen paikannus on klassinen tapa arvioida laitteen paikka, kun sen etäisyys muista laitteista tiedetään. Se toimii hyvin, kun tieto etäisyyksistä on tarkkaa, mutta etäisyysestimaattien ollessa epätarkkoja se usein tuottaa erittäin epätarkkoja tuloksia. Tässä työssä sitä kuitenkin käytetään alkuarvauksien muodostamisessa, koska se antaa jonkin järkevän arvon uskottavuusestimaattorille, joka sitten tarkentaa paikkaestimaattia.

Tutkitaan aluksi tilannetta, jossa laitteita on kolme. Kahden paikat tunnetaan, ja ne ovat  $A = (x_1, y_1)$  ja  $B = (x_2, y_2)$ . Kolmannen laitteen tuntematon paikka on  $C = (x, y)$ , ja laitteen etäisyyden ensimmäisestä laitteesta tiedetään olevan  $d_1$ , ja etäisyys toisesta laitteesta on  $d_2$ . Tilannetta havainnollistaa kuva 1.



Kuva 1: Paikannettava laite ja referenssilaitteet kolmioestimoinnissa

Nyt laitteiden muodostaman kolmion kolmannen sivun pituudeksi voidaan laskea  $d_3 = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ . Sitten kosinilauseen nojalla on voimassa yhtälö

$$d_2^2 = d_1^2 + d_3^2 - 2d_1d_3 \cos \alpha, \quad (6)$$

missä  $\alpha = \angle BAC$ . Yhtälöstä voidaan ratkaista kulman  $\alpha$  kosiniksi

$$\cos \alpha = \frac{d_1^2 + d_3^2 - d_2^2}{2d_1d_3}. \quad (7)$$

Nyt sijainti  $(x, y)$  voidaan esittää laitteen  $A$  sijainnin sekä laitteiden  $B$  ja  $A$  välisen vektorin sekä sitä vastaan kohtisuoran vektorin avulla. Sitä varten lasketaan ensin

sivun  $AC$  projektio sivulle  $AB$ . Tämä projektio on sivun  $AB$  eli vektorin  $u = (x_2 - x_1, y_2 - y_1)$  suuntainen, ja sivun  $AB$  suuntainen yksikkövektori puolestaan on  $\hat{u} = (\frac{x_2 - x_1}{d_3}, \frac{y_2 - y_1}{d_3})$ . Koska vektorin  $AC$  pituus on  $d_1$ , on sen projektion pituus  $\cos \alpha d_1 = \frac{d_1^2 + d_3^2 - d_2^2}{2d_3}$ . Näin saadaan laskettua projektiovektori

$$p = \frac{d_1^2 + d_3^2 - d_2^2}{2d_3^2}(x_2 - x_1, y_2 - y_1).$$

Sivua  $AB$  vastaan kohtisuorassa oleva komponentti puolestaan on vektorin  $(y_2 - y_1, x_1 - x_2)$  suuntainen ja sen pituus on  $d_1 \sin \alpha$ . Trigonometrian peruskaavan mukaan  $\sin^2 \alpha + \cos^2 \alpha = 1$ , jolloin saadaan  $\sin \alpha = \pm \sqrt{1 - \cos^2 \alpha}$ . Näin kohtisuoralle komponentille saadaan kaksi vaihtoehtoa, joista ei voida erottaa, kumpi on oikea. Positiiviseksi ortogonaalikomponentiksi saadaan nyt

$$q = \frac{\sqrt{1 - \cos^2 \alpha} d_1}{d_3}(y_2 - y_1, x_1 - x_2).$$

Lopulta paikan  $(x, y)$  estimaattipariksi saadaan

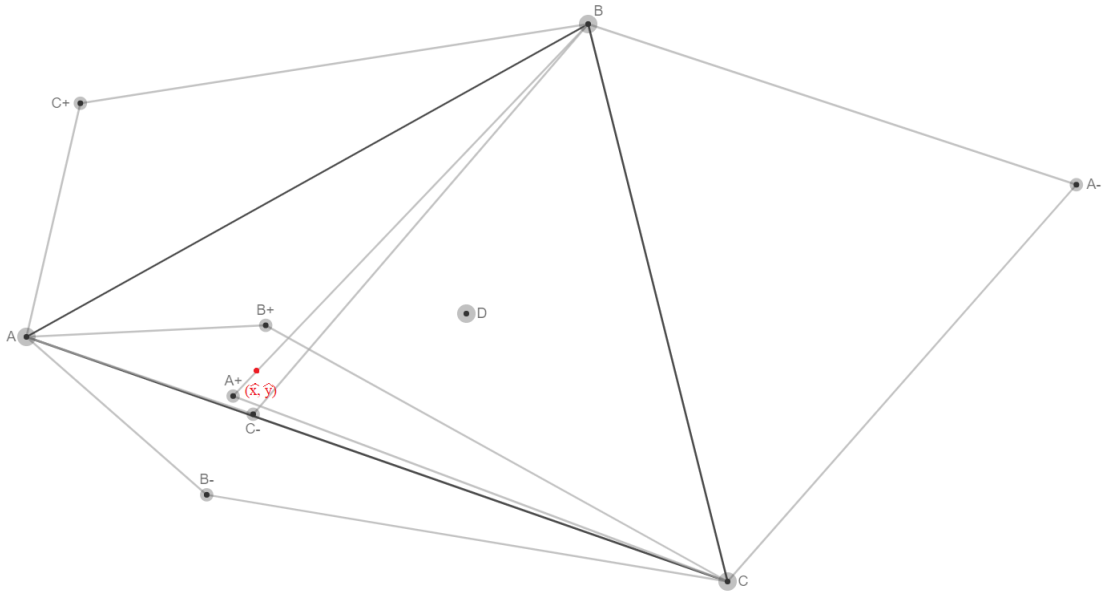
$$(x, y) = (x_1, y_1) + p \pm q.$$

Kyseistä paikkaestimaattiparia kutsutaan jatkossa laitteen  $C$  *bilateraattipariksi* ja parin kumpaakin paikkaa *bilateraatiksi*.

### 2.3.2 Trilateraatti

Tutkitaan seuraavaksi kolmioestimointia tilanteessa, jossa kolmen laitteen paikat tunnetaan, olkoot ne  $A = (x_1, y_1)$ ,  $B = (x_2, y_2)$  ja  $C = (x_3, y_3)$ , ja laitteen  $D = (x, y)$  tuntematonta paikkaa estimoidaan. Nyt, kun laitteita on kolme, saadaan riittävästi informaatiota yksikäsitteisen paikkaestimaatin muodostamiseen, jos etäisyydet niistä tunnetaan. Toisaalta etäisyydestimaatit tuottavat virheellisyydestään johtuen keskenään ristiriitaisia paikkaestimaatteja, joten niistä täytyy yhdistellä kokonaisestimaatti jollain tavalla. Bilateraattien yhdistäminen yhdeksi estimaatiksi kuvaillaan tarkemmin alla. Trilateraattia käytetään tässä työssä laitteiden paikkojen alkuarvauksien muodostamiseen ennen uskottavuusestimointia, ja se muodostetaan yhdistämällä kolmen tunnetun laitteen ja tuntemattoman laitteen välisiä bilateraatteja.

Olkoot  $d_1, d_2$  ja  $d_3$  laitteen  $D$  estimoidut etäisyydet laitteista  $A, B$  ja  $C$ , vastaavasti. Olkoon sitten  $\{C+, C-\} = \{(\hat{x}_{C+}, \hat{y}_{C+}), (\hat{x}_{C-}, \hat{y}_{C-})\}$  laitteiden  $A$  ja  $B$  sijaintien ja etäisyyksien  $d_1$  ja  $d_2$  avulla laitteesta  $D$  muodostettu bilateraattipari. Olkoot sitten vastaavasti  $\{B+, B-\} = \{(\hat{x}_{B+}, \hat{y}_{B+}), (\hat{x}_{B-}, \hat{y}_{B-})\}$  laitteiden  $A$  ja  $C$  sijaintien ja etäisyyksien  $d_1$  ja  $d_3$  avulla laitteesta  $D$  muodostettu bilateraattipari, sekä  $\{A+, A-\} = \{(\hat{x}_{A+}, \hat{y}_{A+}), (\hat{x}_{A-}, \hat{y}_{A-})\}$  laitteiden  $B$  ja  $C$  sijaintien ja etäisyyksien  $d_2$  ja  $d_3$  avulla laitteesta  $D$  muodostettu bilateraattipari. Nyt ideaalitapauksessa eli etäisyyksien ollessa virheettömiä kustakin bilateraattiparista löytyy estimaatti, joka kertoo paikan  $(x, y)$  täsmälleen oikein. Näin voitaisiin valita kolme samaa estimaattia ja määrittää se laitteen paikaksi. Koska etäisyyksissä kuitenkin on virhettä, täytyy luoda jokin metriikka, jonka avulla päättää, mitä estimaattia tai estimaatteja paikan arviointiin tulisi käyttää. Tilannetta havainnollistaa kuva 2.



Kuva 2: Paikannettava laite ja referenssilaitteet sekä ja paikkaestimaatit trilateraatiassa

Tässä työssä valinta toteutetaan seuraavasti. Estimaatiksi  $C' = (x_{C'}, y_{C'})$  valitaan ensin jompikumpi paikoista  $C+$  ja  $C-$ , sitten estimaatiksi  $B' = (x_{B'}, y_{B'})$  valitaan jompikumpi paikoista  $B+$  ja  $B-$  ja vastaavasti estimaatiksi  $A' = (x_{A'}, y_{A'})$  valitaan jompikumpi paikoista  $A+$  ja  $A-$ . Seuraavaksi lasketaan paikkojen keskiarvo  $(\bar{x}, \bar{y}) = (\frac{x_{C'} + x_{B'} + x_{A'}}{3}, \frac{y_{C'} + y_{B'} + y_{A'}}{3})$  ja lopuksi niiden neliövirhesumma kaavalla

$$\hat{\sigma}^2 = (x_{A'} - \bar{x})^2 + (x_{B'} - \bar{x})^2 + (x_{C'} - \bar{x})^2 + (y_{A'} - \bar{y})^2 + (y_{B'} - \bar{y})^2 + (y_{C'} - \bar{y})^2. \quad (8)$$

Nämä keskiarvot ja neliövirhesummat lasketaan jokaisesta mahdollisesta paikkojen  $A'$ ,  $B'$  ja  $C'$  kombinaatiosta erikseen, ja kombinaatioista parhaaksi valitaan se, jonka neliövirhesumma on pienin. Paikan estimaattina käytetään sitten valitusta kombinaatiosta valittua keskiarvoa.

## 2.4 Uskottavuusfunktio

Määritetään ensin tilastotieteellinen malli signaalitehojen käyttäytymiselle verkossa. Oletetaan, että verkossa on  $n$  laitetta joiden sijainnit kaksiulotteisella tasolla ovat  $(x_1, y_1), \dots, (x_n, y_n)$ . Nämä ovat mallin tilastoparametrit.

Olkoon kahden laitteen  $i$  ja  $j$  sijainnit  $(x_i, y_i)$  ja  $(x_j, y_j)$ . Tällöin niiden välinen etäisyys on  $d_{i,j} = d(x_i, y_i, x_j, y_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$  ja niiden välinen tehohäviö noudattaa jakaumaa  $N(P_0 + 10\gamma \lg(\frac{d_{i,j}}{d_0}), \sigma^2)$ . Oletetaan sitten, että laite  $i$  havaitsee laitteen  $j$ , jos laitteesta  $j$  laitteeseen  $i$  lähetetyn signaalin teho on vähintään tietty kynnysarvo, tai toisin ilmaistuna tehohäviö on pienempi tai yhtäsuuri kuin tietty kynnysarvo  $P_{max}$ .

Tehohäviö  $P$  on matemaattisesti tarkasteltuna normaalijakautunut satunnaisuuttuja, jolloin uskottavuusfunktion muodostamiseksi käytetään normaalijakautuman tiheysfunktioita. Jos jokin laite  $i$  havaitsee toisen laitteen  $j$  ja mittaa niiden

välisestä signaalista tehohäviön  $p_{i,j}$ , voidaan laskea laitteen  $i$  sijainnin uskottavuudeksi

$$L(x_i, y_i | p_{i,j}, x_j, y_j) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(p_{i,j} - P(x_i, y_i, x_j, y_j))^2}{2\sigma^2}}. \quad (9)$$

kullakin laitteen  $j$  sijainnilla  $(x_j, y_j)$ . Tässä siis

$$\begin{aligned} P(x_i, y_i, x_j, y_j) &= P_0 + 10\gamma \lg \left( \frac{d(x_i, y_i, x_j, y_j)}{d_0} \right) \\ &= P_0 + 10\gamma \lg \left( \frac{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{d_0} \right) \\ &= P_0 + 5\gamma \lg((x_i - x_j)^2 + (y_i - y_j)^2) - 10\gamma \lg(d_0) \end{aligned}$$

on tehon  $p_{i,j}$  odotusarvo annetuilla sijainneilla  $(x_i, y_i)$  ja  $(x_j, y_j)$ .

Lopullista uskottavuusfunktiota muodostettaessa oletetaan linkkien tehohäviöiden satunnaisvirheiden  $X_{ij}$  olevan riippumattomia toisistaan, jolloin sijainnin  $(x_i, y_i)$  kokonaisuskottavuus voidaan laskea kertomalla keskenään kaikki komponentit  $L(x_i, y_i | p_{i,j}, x_j, y_j)$  yli havaittujen laitteiden  $(x_j, y_j)$ , jolloin saadaan

$$L(x_i, y_i) = \prod_{j \in \mathcal{D}(i)} L(x_i, y_i | p_{i,j}, x_j, y_j). \quad (10)$$

Yllä mainittu joukko  $\mathcal{D}(i)$  sisältää täsmälleen ne laitteet, jotka havaitaan laitteessa  $i$  eli toisin sanoen laitteista lähetetyn signaalin tehohäviö ei ylitä kynnsarvoa  $P_{max}$  laitteen  $i$  ja  $j$  välisessä linkissä. Uskottavuusfunktioon voitaisiin sisällyttää myös todennäköisyydet, joilla havaitsematta jääneet laitteet jäävät havaitsematta tietyllä etäisyydellä, jotta havaitsemattomistakin laitteista saataisiin informaatiota irti. Tämän havaitsemattajäämistodennäköisyyden vaikutusta pidetään kuitenkin kirjoittajan simulaatiokokeilujen perusteella pienenä, ja käytännössä sellainen vaatisi laitteiden tietoisuutta muistakin laitteista kuin lähelläolevista, joten havaitsemattomien laitteiden tuottamia komponentteja ei tässä työssä tarkastella.

Laskennan helpottamiseksi käytetään varsinaisen uskottavuusfunktion sijaan log-uskottavuusfunktiota, joka on siis luonnollinen logaritmi uskottavuusfunktiosta. Se saadaan ottamalla logaritmi jokaisesta uskottavuusfunktion komponentista ja laske-  
malla nämä logaritmit yhteen. Näin log-uskottavuusfunktion lausekkeeksi saadaan

$$l(x_i, y_i) = \sum_{j \in \mathcal{D}(i)} -\frac{(p_{i,j} - P(x_i, y_i, x_j, y_j))^2}{2\sigma^2}. \quad (11)$$

Yllä jokaisessa termissä toistuvan tekijän  $\frac{1}{\sqrt{2\pi\sigma}}$  logaritmi poistettiin lausekkeesta, sillä log-uskottavuusfunktiota ja varsinkaan sen ääriarvokohtia tarkastellessa tällaisia vakiotermejä ei ole tarpeen laskea. Jatkossa termien uskottavuusfunktio ja log-uskottavuusfunktio välillä ei tehdä eroa.

Uskottavuusmaksimin laskemiseksi lasketaan vielä uskottavuusfunktion gradientti ja sitä varten uskottavuusfunktion osittaisderivaatat muuttujien  $x_i$  ja  $y_i$  suhteen. Tässä  $P(x_1, y_1, x_2, y_2) = P_0 + 5\gamma \lg\left(\frac{(x_1 - x_2)^2 + (y_1 - y_2)^2}{d_0}\right) - 10 \lg(d_0)$ . Tällöin funktion osittaisderivaatta muuttujan  $x_i$  suhteen on

$$\begin{aligned}\frac{\partial}{\partial x_i} l(x_i, y_i) &= \sum_{j \in \mathcal{D}(i)} \left( -\frac{\partial}{\partial x_i} \left( \frac{(p_{i,j} - P(x_i, y_i, x_j, y_j))^2}{2\sigma^2} \right) \right) \\ &= \sum_{j \in \mathcal{D}(i)} \left( \frac{(p_{i,j} - P(x_i, y_i, x_j, y_j))}{\sigma^2} \frac{\partial}{\partial x_i} P(x_i, y_i, x_j, y_j) \right).\end{aligned}$$

Tehohäviön osittaisderivaatta taas on

$$\begin{aligned}\frac{\partial}{\partial x_i} P(x_i, y_i, x_j, y_j) &= \frac{5\gamma d_0}{\ln(10)((x_i - x_j)^2 + (y_i - y_j)^2)} \frac{\partial}{\partial x_i} \left( \frac{(x_i - x_j)^2 + (y_i - y_j)^2}{d_0} \right) \\ &= \frac{10\gamma(x_i - x_j)}{\ln(10)((x_i - x_j)^2 + (y_i - y_j)^2)}.\end{aligned}$$

Näin log-uskottavuusfunktion osittaisderivaataksi muuttujan  $x_i$  suhteen saadaan

$$\frac{\partial}{\partial x_i} l(x_i, y_i) = \sum_{j \in \mathcal{D}(i)} \left( \frac{10\gamma(x_i - x_j)(p_{i,j} - P(x_i, y_i, x_j, y_j))}{\ln(10)\sigma^2((x_i - x_j)^2 + (y_i - y_j)^2)} \right).$$

Symmetrian nojalla osittaisderivaatta muuttujan  $y_i$  suhteen on vastaavasti

$$\frac{\partial}{\partial y_i} l(x_i, y_i) = \sum_{j \in \mathcal{D}(i)} \left( \frac{10\gamma(y_i - y_j)(p_{i,j} - P(x_i, y_i, x_j, y_j))}{\ln(10)\sigma^2((x_i - x_j)^2 + (y_i - y_j)^2)} \right).$$

## 3 Uskottavuusfunktion maksimointi

### 3.1 Konjugaattigradienttialgoritmi

Laitteiden paikannus tässä työssä perustuu suurimman uskottavuuden estimaatteihin, toisin sanoen niiden parametrien arvojen löytämiseen, joilla uskottavuusfunktio saa suurimman arvonsa. Sijainnin ehdollinen uskottavuusfunktio on määritelty edellisessä osiossa ja tässä osiossa kuvaillaan laitteen sijainnin ehdollisen uskottavuusfunktion maksimin löytämistä konjugaattigradienttialgoritilla ([1], s. 420 - 425). Juuri tämä algoritmi otetaan käyttöön, koska funktiolla on kohtuullisen helposti laskettavat gradientit ja se on usein "enimmiltä osin siisti".

Konjugaattigradienttialgoritilla voi maksimoida useamman muuttujan jatkuvasti differentioituvia funktioita, erityisesti sellaisia, jotka muistuttavat läheisesti paraboloideja maksimi- tai minimikohdan ympäristössä. Jos optimoitava funktio on tarkalleen paraboloidi, ääriarvokohta löytyy iteroimalla kukin konjugaattivektori korkeintaan kerran.

Algoritmin esittelyä varten oletetaan usean muuttujan funktio  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  ja oletetaan aluksi, että se on parabolinen, ts.

$$f(\mathbf{x}) = c - \mathbf{b} \cdot \mathbf{x} + \frac{1}{2} \mathbf{x} \mathbf{A} \mathbf{x}^T,$$

missä  $\mathbf{x} \in \mathcal{M}_{1,n}$  on vaakavektorimuuttuja, joka sisältää lukumuuttujat  $x_1, \dots, x_n$ ,  $\mathbf{b}$  on samandimensioinen vakiovaakavektori,  $\mathbf{A}$  on symmetrinen  $n \times n$ -matriisi ja  $c$  vakio. Tällöin funktion  $f$  gradientti on

$$\nabla f(\mathbf{x}) = -\mathbf{b} + \mathbf{A}\mathbf{x}. \quad (12)$$

Jatkuvasti differentioituvan funktion ääriarvokohdat sijaitsevat tunnetusti gradientin nollakohdissa, mistä saadaan lineaarinen yhtälö (tai yhtälöryhmä)

$$\mathbf{A}\mathbf{x} + \mathbf{b} = \mathbf{0}.$$

Tällaisten yhtälöryhmien ratkaisu onkin konjugaattigradienttialgoritmin alkuperäinen tarkoitus, mutta nyt sitä sovelletaan funktion maksimointiin.

Algoritmissa määritetään ensin alkupiste  $\mathbf{P}_0$ , josta liikutaan myöhemmin määriteltävien konjugaattivektorien  $\mathbf{h}_i$  suuntaisesti kohti funktion maksimia. Kutakin konjugaattivektoria käsiteltäessä muodostetaan vektorin suuntainen iteraatiopisteen  $\mathbf{P}_i$  kautta kulkeva suora, josta taas etsitään maksimi jollain toisella yhden muuttujan funktion optimointiin tarkoitetulla algoritmilla. Tämä maksimikohta otetaan uudeksi iteraatiopisteeksi  $\mathbf{P}_{i+1}$  ja samalla määritetään uusi konjugaattivektori  $\mathbf{h}_{i+1}$  myöhemmin esiteltävän kaavan mukaan. Konjugaattigradienttialgoritmi hyödyntää konjugaattivektorien ortogonaalisuutta tietyssä matriisin  $\mathbf{A}$  (likimain) määrittämässä sisätuloavaruudessa, mikä johtaa algoritmin nopeaan suppenemiseen verrattuna esimerkiksi yksinkertaisempaan jyrkimmän laskun algoritmiin, jossa vastaavat iteraatiot suoritetaan suoraan funktion gradientin suuntaisesti.

Olkoon siis  $\mathbf{P}_0$  jokin  $n$ -ulotteinen aloituspiste. Olkoon sitten aloitusvektori  $\mathbf{g}_0 = \nabla f(\mathbf{P}_0)$  ja  $\mathbf{h}_0 = \mathbf{g}_0$ . Tällöin kullekin  $i$  saadaan  $\mathbf{P}_{i+1} = \mathbf{P}_i + x \cdot \mathbf{h}_i$ , missä  $x$  on sellainen reaalityyppinen luku, jolla  $f(\mathbf{P}_{i+1})$  saa pienimmän arvonsa. Määritellään sitten  $\mathbf{g}_{i+1} =$



$\nabla f(\mathbf{P}_{i+1})$  ja  $\mathbf{h}_{i+1} = \mathbf{g}_{i+1} + \gamma_i \mathbf{h}_i$ , missä  $\gamma_i = \frac{\mathbf{g}_{i+1} \cdot \mathbf{g}_{i+1}}{\mathbf{g}_i \cdot \mathbf{g}_i}$ . Näillä vektoreilla on hyödyllisiä ortogonaalisuus- ja konjugaattiominaisuuksia, jotka osoitetaan seuraavassa aliluvussa ja joiden nojalla ensimmäiset  $n$  kappaletta (avaruuden dimension ollessa  $n$ ) vektoreita  $\mathbf{h}_i$  johtavat minimointitehtävän täsmälliseen ratkaisuun, jos funktio  $f$  on parabolinen.

Tutkitaan nyt gradientin  $\mathbf{g}_{i+1} = \nabla f(\mathbf{P}_{i+1})$  arvoa parabolisella funktiolla  $f$ , jolle on voimassa lauseke

$$\nabla f(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b}.$$

Lausekkeesta saadaan

$$\mathbf{g}_{i+1} = \nabla f(\mathbf{P}_{i+1}) = \nabla f(\mathbf{P}_i + x\mathbf{h}_i) = \mathbf{A}(\mathbf{P}_i + x\mathbf{h}_i) - \mathbf{b}$$

missä  $x$  on reaaliluku, joka minimoi funktion  $f$  arvon vektorin  $\mathbf{h}_i$  suuntaisella suoralla, joka kulkee pisteen  $\mathbf{P}_i$  kautta. Koska kyseessä on funktion minimikohta tällä suoralla, funktion  $f$  suunnattu derivaatta vektorin  $\mathbf{h}_i$  suuntaan on nolla, siis

$$(\mathbf{A}(\mathbf{P}_i + x\mathbf{h}_i) - \mathbf{b}) \cdot \mathbf{h}_i = 0.$$

Ratkaistaan tästä luku  $x$ :

$$\begin{aligned} \mathbf{h}_i^T \mathbf{A}(\mathbf{P}_i + x\mathbf{h}_i) - \mathbf{b} \cdot \mathbf{h}_i &= 0 \\ \mathbf{h}_i^T \mathbf{A}(\mathbf{P}_i + x\mathbf{h}_i) &= \mathbf{b} \cdot \mathbf{h}_i \\ x \cdot \mathbf{h}_i^T \mathbf{A}\mathbf{h}_i + \mathbf{h}_i^T \mathbf{A}\mathbf{P}_i &= \mathbf{b} \cdot \mathbf{h}_i \\ x \cdot \mathbf{h}_i^T \mathbf{A}\mathbf{h}_i &= \mathbf{b} \cdot \mathbf{h}_i - \mathbf{h}_i^T \mathbf{A}\mathbf{P}_i \\ x &= \frac{\mathbf{b} \cdot \mathbf{h}_i - \mathbf{h}_i^T \mathbf{A}\mathbf{P}_i}{\mathbf{h}_i^T \mathbf{A}\mathbf{h}_i} \end{aligned}$$

Näin pisteelle  $\mathbf{P}_{i+1}$  saadaan esitys

$$\mathbf{P}_{i+1} = \mathbf{P}_i + \left( \frac{\mathbf{b} \cdot \mathbf{h}_i - \mathbf{h}_i^T \mathbf{A}\mathbf{P}_i}{\mathbf{h}_i^T \mathbf{A}\mathbf{h}_i} \right) \mathbf{h}_i$$

ja lopulta vektorille  $\mathbf{g}_{i+1}$  saadaan muoto

$$\begin{aligned} \mathbf{g}_{i+1} &= \nabla f(\mathbf{P}_{i+1}) = \mathbf{A}\mathbf{P}_{i+1} - \mathbf{b} \\ &= \mathbf{A}\mathbf{P}_i - \mathbf{b} + \left( \frac{\mathbf{b} \cdot \mathbf{h}_i - \mathbf{h}_i^T \mathbf{A}\mathbf{P}_i}{\mathbf{h}_i^T \mathbf{A}\mathbf{h}_i} \right) \mathbf{A}\mathbf{h}_i \\ &= \mathbf{g}_i - \left( \frac{\mathbf{h}_i \cdot \mathbf{g}_i}{\mathbf{h}_i^T \mathbf{A}\mathbf{h}_i} \right) \mathbf{A}\mathbf{h}_i \\ &= \mathbf{g}_i - \left( \frac{\mathbf{g}_i \cdot \mathbf{h}_i}{\mathbf{h}_i^T \mathbf{A}\mathbf{h}_i} \right) \mathbf{A}\mathbf{h}_i. \end{aligned}$$

### 3.1.1 Konjugaattivektorien ortogonaalisuus ja konjugaattiominaisuudet

Osoitetaan, että vektorit  $\mathbf{g}_i$  ja  $\mathbf{h}_i$  täyttävät seuraavat ortogonaalisuus- ja konjugaattiehtdot:

$$(C1) \quad \mathbf{g}_i \cdot \mathbf{g}_j = 0$$

$$(C2) \quad \mathbf{g}_i \cdot \mathbf{h}_j = 0$$

$$(C3) \quad \mathbf{h}_i^T \mathbf{A} \mathbf{h}_j = 0$$

aina, kun  $j < i < n$ .

*Todistus.* Osoitetaan väitteet induktiolla luvun  $i$  suhteen, siis induktioaskeleessa oletetaan, että kukin edelläolevista väitteistä on tosi arvolla  $i$ , kaikilla  $j < i$ . Lisäksi määritellään todistuksen tueksi kaksi uutta ehtoa:

$$(C4) \quad \mathbf{g}_i \cdot \mathbf{g}_i = \mathbf{g}_i \cdot \mathbf{h}_i$$

$$(C5) \quad \mathbf{g}_j^T \mathbf{A} \mathbf{h}_i = \mathbf{h}_j^T \mathbf{A} \mathbf{h}_i \quad \text{kaikilla } j \leq i + 1,$$

eli oletetaan myös nämä todeksi kullakin induktioaskeleen arvolla  $i$  ja osoitetaan ne todeksi arvolla  $i + 1$ .

Alkuaskelta varten todetaan, että ehdot C1, C2 ja C3 ovat triviaalisti tosia, kun  $i = 0$ , sillä tutkittavia indeksejä  $j < 0$  ei ole. Ehdot C4 ja C5 taas ovat tosia arvolla  $i = 0$ , sillä  $\mathbf{g}_0 = \mathbf{h}_0$ .

Sitten siirrytään induktioaskeleeseen ja oletetaan, että ehdot C1-C5 ovat voimassa arvolla  $i$ , ja osoitetaan ne todeksi arvolla  $i + 1$ . Vektorille  $\mathbf{g}_{i+1}$  pätee  $\mathbf{g}_{i+1} = \nabla f(\mathbf{P}_{i+1})$ . Funktion ollessa parabolista muotoa saadaan

$$\nabla f(\mathbf{P}_{i+1}) = \nabla f(\mathbf{P}_i + x\mathbf{h}_i) = \mathbf{A}(\mathbf{P}_i + x\mathbf{h}_i) + \mathbf{b} = \mathbf{A}\mathbf{P}_i + \mathbf{b} + x\mathbf{A}\mathbf{h}_i = \mathbf{g}_i + x\mathbf{A}\mathbf{h}_i$$

jollain reaaliluvulla  $x$ .

Osoitetaan nyt ehto C2 arvolle  $i+1$ . Vektori  $\mathbf{g}_{i+1}$  on funktion  $f$  gradientti pisteessä  $\mathbf{P}_{i+1}$ , johon päädyttiin minimoimalla funktiota vektorin  $\mathbf{h}_i$  suuntaisella suoralla, jolloin vektorin  $\mathbf{g}_{i+1}$  täytyy olla ortogonaalinen vektorin  $\mathbf{h}_i$  kanssa, siis

$$\mathbf{g}_{i+1} \cdot \mathbf{h}_i = 0.$$

Jos taas  $j < i$ , niin edellä todetun perusteella sekä käyttämällä induktio-oletuksen ehtoja C2 ja C3 saadaan

$$\mathbf{g}_{i+1} \cdot \mathbf{h}_j = \mathbf{g}_i \cdot \mathbf{h}_j + x\mathbf{A}\mathbf{h}_i \cdot \mathbf{h}_j = \mathbf{g}_i \cdot \mathbf{h}_j + x\mathbf{h}_i^T \mathbf{A} \mathbf{h}_j = 0 + 0 = 0.$$

Siis ehto C2 on voimassa arvolle  $i + 1$ . Osoitetaan sitten ehto C1 arvolle  $i + 1$ . Kullakin  $j < i + 1$  saadaan:

$$\mathbf{g}_{i+1} \cdot \mathbf{g}_j = \mathbf{g}_i \cdot \mathbf{g}_j + x\mathbf{A}\mathbf{h}_i \cdot \mathbf{g}_j = \mathbf{g}_i \cdot \mathbf{g}_j + x\mathbf{g}_j^T \mathbf{A} \mathbf{h}_i$$

Jos  $j < i$ , niin induktio-oletuksen mukaan ehto C1 antaa  $\mathbf{g}_i \cdot \mathbf{g}_j = 0$  kun taas ehdot C3 ja C5 antavat

$$\mathbf{g}_j^T \mathbf{A} \mathbf{h}_i = \mathbf{h}_i^T \mathbf{A} \mathbf{h}_j = 0,$$

ja siten  $\mathbf{g}_{i+1} \cdot \mathbf{g}_j = 0$ .

Jos taas  $j = i$ , saadaan ehtojen C4 ja C5 avulla

$$\mathbf{g}_{i+1} \cdot \mathbf{g}_i = \mathbf{g}_i \cdot \mathbf{g}_i + x \mathbf{g}_i^T \mathbf{A} \mathbf{h}_i = \mathbf{g}_i \cdot \mathbf{h}_i + x \mathbf{h}_i^T \mathbf{A} \mathbf{h}_i = \mathbf{g}_{i+1} \cdot \mathbf{h}_i,$$

joka on nolla arvolle  $i+1$  edellä osoitetun ehdon C2 nojalla. Siis ehto C1 on voimassa arvolle  $i+1$ .

C3 saadaan osoittamalla, että vektorit  $\mathbf{h}_i$  tulevat itseasiassa Gramin-Schmidtin ortogonalisointimenetelmästä vektoreilla  $\mathbf{g}_i$  yli sisätulon  $\langle \mathbf{g}, \mathbf{h} \rangle_{\mathbf{A}} = \mathbf{g}^T \mathbf{A} \mathbf{h}$ . Tätä varten todetaan, että indeksistä 0 indeksiin  $i+1$  asti vektorit  $\mathbf{g}_i$  ovat ortogonaaliset tavanomaisen sisätulon suhteen ja muodostavat siten avaruuden  $\mathbb{R}^n$  jonkin  $i+1$ -ulotteisen aliavaruuden kannan. Toisin sanoen osoitetaan, että

$$\mathbf{h}_{i+1} = \mathbf{g}_{i+1} - \left( \frac{\mathbf{h}_i^T \mathbf{A} \mathbf{g}_{i+1}}{\mathbf{h}_i^T \mathbf{A} \mathbf{h}_i} \right) \mathbf{h}_i.$$

Toisaalta  $\mathbf{h}_i$ :n määritelmän mukaan

$$\mathbf{h}_{i+1} = \mathbf{g}_{i+1} + \left( \frac{\mathbf{g}_{i+1} \cdot \mathbf{g}_i}{\mathbf{g}_i \cdot \mathbf{g}_i} \right) \mathbf{h}_i.$$

Pitää siis osoittaa, että

$$\frac{\mathbf{h}_i^T \mathbf{A} \mathbf{g}_{i+1}}{\mathbf{h}_i^T \mathbf{A} \mathbf{h}_i} = - \frac{\mathbf{g}_{i+1} \cdot \mathbf{g}_i}{\mathbf{g}_i \cdot \mathbf{g}_i}$$

Vasenta puolta saadaan avattua käyttämällä yhtälöä  $\mathbf{g}_{i+1} = \mathbf{g}_i - \left( \frac{\mathbf{g}_i \cdot \mathbf{h}_i}{\mathbf{h}_i^T \mathbf{A} \mathbf{h}_i} \right) \mathbf{A} \mathbf{h}_i$  sekä ehtoa C5:

$$\frac{\mathbf{h}_i^T \mathbf{A} \mathbf{g}_{i+1}}{\mathbf{h}_i^T \mathbf{A} \mathbf{h}_i} = \frac{\mathbf{h}_i^T \mathbf{A} \mathbf{h}_i}{\mathbf{h}_i^T \mathbf{A} \mathbf{h}_i} - \left( \frac{\mathbf{g}_i \cdot \mathbf{h}_i}{\mathbf{h}_i^T \mathbf{A} \mathbf{h}_i} \right) \frac{\mathbf{h}_i^T \mathbf{A}^2 \mathbf{h}_i}{\mathbf{h}_i^T \mathbf{A} \mathbf{h}_i} = 1 - \left( \frac{\mathbf{g}_i \cdot \mathbf{h}_i}{(\mathbf{h}_i^T \mathbf{A} \mathbf{h}_i)^2} \right) \mathbf{h}_i^T \mathbf{A}^2 \mathbf{h}_i$$

Oikeasta puolesta taas tulee

$$\begin{aligned} \frac{\mathbf{g}_{i+1} \cdot \mathbf{g}_{i+1}}{\mathbf{g}_i \cdot \mathbf{g}_i} &= \frac{\left( \mathbf{g}_i - \left( \frac{\mathbf{g}_i \cdot \mathbf{h}_i}{\mathbf{h}_i^T \mathbf{A} \mathbf{h}_i} \right) \mathbf{A} \mathbf{h}_i \right) \cdot \left( \mathbf{g}_i - \left( \frac{\mathbf{g}_i \cdot \mathbf{h}_i}{\mathbf{h}_i^T \mathbf{A} \mathbf{h}_i} \right) \mathbf{A} \mathbf{h}_i \right)}{\mathbf{g}_i \cdot \mathbf{g}_i} \\ &= \frac{\mathbf{g}_i \cdot \mathbf{g}_i}{\mathbf{g}_i \cdot \mathbf{g}_i} - 2 \cdot \frac{\left( \frac{\mathbf{g}_i \cdot \mathbf{h}_i}{\mathbf{h}_i^T \mathbf{A} \mathbf{h}_i} \right) \mathbf{g}_i^T \mathbf{A} \mathbf{h}_i}{\mathbf{g}_i \cdot \mathbf{g}_i} + \frac{\left( \frac{\mathbf{g}_i \cdot \mathbf{h}_i}{\mathbf{h}_i^T \mathbf{A} \mathbf{h}_i} \right)^2 \mathbf{h}_i^T \mathbf{A}^T \mathbf{A} \mathbf{h}_i}{\mathbf{g}_i \cdot \mathbf{g}_i} \end{aligned}$$

Käyttämällä ehtoja C4 ja C5 arvolla  $i$  sekä matriisin  $\mathbf{A}$  symmetrisyyttä saadaan edelleen

$$\begin{aligned} \frac{\mathbf{g}_{i+1} \cdot \mathbf{g}_{i+1}}{\mathbf{g}_i \cdot \mathbf{g}_i} &= 1 - 2 \cdot \left( \frac{\mathbf{g}_i \cdot \mathbf{h}_i}{\mathbf{g}_i \cdot \mathbf{h}_i} \right) \left( \frac{\mathbf{h}_i^T \mathbf{A} \mathbf{h}_i}{\mathbf{h}_i^T \mathbf{A} \mathbf{h}_i} \right) + \left( \frac{\mathbf{g}_i \cdot \mathbf{h}_i}{(\mathbf{h}_i^T \mathbf{A} \mathbf{h}_i)^2} \right) \mathbf{h}_i^T \mathbf{A}^2 \mathbf{h}_i \\ &= -1 + \left( \frac{\mathbf{g}_i \cdot \mathbf{h}_i}{(\mathbf{h}_i^T \mathbf{A} \mathbf{h}_i)^2} \right) \mathbf{h}_i^T \mathbf{A}^2 \mathbf{h}_i \end{aligned}$$

siis  $\frac{\mathbf{h}_{i+1}^T \mathbf{A} \mathbf{g}_{i+1}}{\mathbf{h}_{i+1}^T \mathbf{A} \mathbf{h}_{i+1}} = -\frac{\mathbf{g}_{i+1} \cdot \mathbf{g}_{i+1}}{\mathbf{g}_i \cdot \mathbf{g}_i}$ , jolloin vektorit  $\mathbf{h}_i$  ovat ortogonaaliset yli operaation  $\langle \mathbf{g}, \mathbf{h} \rangle_{\mathbf{A}} = \mathbf{g}^T \mathbf{A} \mathbf{h}$  muodostaman sisätuloavaruuden ja ehto C3 näin voimassa arvolle  $i + 1$ .

Ehto C4 saadaan käyttämällä edellä todistettua induktioväitteen ehtoa C2. Näin vektorin  $\mathbf{h}_i$  lausekkeesta saadaan

$$\mathbf{g}_{i+1} \cdot \mathbf{h}_{i+1} = \mathbf{g}_{i+1} \cdot \mathbf{g}_{i+1} + \left( \frac{\mathbf{g}_{i+1} \cdot \mathbf{g}_{i+1}}{\mathbf{g}_i \cdot \mathbf{g}_i} \right) \mathbf{g}_{i+1} \cdot \mathbf{h}_i = \mathbf{g}_{i+1} \cdot \mathbf{g}_{i+1}.$$

Käyttämällä edellä todistettua ehtoa C3 arvolla  $i + 1$  saadaan todistettua C5 arvolle  $i + 1$ . Kaikilla  $j < i + 1$  pätee

$$\begin{aligned} \mathbf{h}_{j+1}^T \mathbf{A} \mathbf{h}_{i+1} &= \mathbf{g}_{j+1} \mathbf{A} \mathbf{h}_{i+1} + \left( \frac{\mathbf{g}_{i+1} \cdot \mathbf{g}_{i+1}}{\mathbf{g}_i \cdot \mathbf{g}_i} \right) \mathbf{h}_j^T \mathbf{A} \mathbf{h}_{i+1} \\ &= \mathbf{g}_{j+1} \mathbf{A} \mathbf{h}_{i+1} + \left( \frac{\mathbf{g}_{i+1} \cdot \mathbf{g}_{i+1}}{\mathbf{g}_i \cdot \mathbf{g}_i} \right) \mathbf{h}_{i+1}^T \mathbf{A} \mathbf{h}_j = \mathbf{g}_{j+1} \mathbf{A} \mathbf{h}_{i+1}. \end{aligned}$$

Lisäksi koska  $\mathbf{g}_0 = \mathbf{h}_0$ , niin  $\mathbf{h}_0^T \mathbf{A} \mathbf{h}_{i+1} = \mathbf{g}_0^T \mathbf{A} \mathbf{h}_{i+1}$ . Siis C5 on voimassa luvuille  $i + 1$  ja kaikille  $j < i + 1$ .

Siis ehtojen C1-C5 ollessa voimassa arvolle  $i$ , ne ovat voimassa arvolle  $i + 1$  ja väite on todistettu induktiolla.  $\square$

### 3.1.2 Parabolisen funktion ääriarvokohdan löytäminen

Koska vektorit  $\mathbf{h}_0, \dots, \mathbf{h}_{n-1}$  ovat edellä osoitettujen ehtojen mukaan ortogonaaliset sisätulon  $\langle \mathbf{g}, \mathbf{h} \rangle_{\mathbf{A}}$  suhteen, ne muodostavat avaruuden  $\mathbb{R}^n$  kannan. Osoitetaan, että piste  $\mathbf{P}_n$  on parabolisen funktion  $f$  minimikohta osoittamalla, että funktion gradientti pisteessä  $\mathbf{P}_n$  nollavektori, siis  $\mathbf{A} \mathbf{P}_i - \mathbf{b} = 0$ .

Kullakin  $0 \leq i < n$  piste  $\mathbf{P}_{i+1}$  on saatu minimoimalla funktiota vektorin  $\mathbf{h}_i$  suuntaisesti, siis vektorin  $\mathbf{h}_i$  suuntainen komponentti gradientista on nolla. Jos lisäksi oletetaan induktiivisesti, että pisteessä  $\mathbf{P}_i$  vektorin  $\mathbf{h}_j$  suuntainen komponentti gradientista on nolla kaikilla  $j < i$ , eli  $\nabla f(\mathbf{P}_i) \cdot \mathbf{h}_j = 0$ , niin vastaavaksi komponentiksi pisteessä  $\mathbf{P}_{i+1}$  saadaan

$$\begin{aligned} \nabla f(\mathbf{P}_{i+1}) \cdot \mathbf{h}_j &= \nabla f(\mathbf{P}_i + x_i \mathbf{h}_i) \cdot \mathbf{h}_j = \mathbf{A}(\mathbf{P}_i + x_i \mathbf{h}_i) \cdot \mathbf{h}_j - \mathbf{b} \cdot \mathbf{h}_j \\ &= (\mathbf{A} \mathbf{P}_i - \mathbf{b}) \cdot \mathbf{h}_j + x_i \cdot \mathbf{h}_j^T \mathbf{A} \mathbf{h}_i \\ &= \nabla f(\mathbf{P}_i) \cdot \mathbf{h}_j + x_i \langle \mathbf{h}_j, \mathbf{h}_i \rangle_{\mathbf{A}} = 0 + 0 = 0. \end{aligned}$$

Siis induktion perusteella kullakin  $i$  on voimassa  $\nabla f(\mathbf{P}_i) \cdot \mathbf{h}_j = 0$  kaikilla  $j < i$ , erityisesti  $\nabla f(\mathbf{P}_n) \cdot \mathbf{h}_j = 0$  kaikilla  $j < n$ . Siispä funktion gradientti on nolla pisteessä  $\mathbf{P}_n$  ja piste  $\mathbf{P}_n$  on siis parabolisen funktion  $f$  globaali minimi tai maksimi.

Näin konjugaattigradienttialgoritmi antaa hyvän numeerisen menetelmän (lokaalisti) likimain parabolisen funktion maksimointiin. On myös syytä huomata, että edellä kuvattu algoritmi ei evaluoi matriisia  $A$  missään vaiheessa, vaan ainoastaan funktiota  $f$  sekä sen gradienttia.

## 3.2 Kohdefunktion maksimointi yhdessä ulottuvuudessa

### 3.2.1 Kolmen pisteen kautta kulkevan paraabelin huippukohta

Johdetaan aluksi kaava pisteiden  $(a, f(a))$ ,  $(b, f(b))$  ja  $(c, f(c))$  kulkevan paraabelin huippukohdalle. Olkoon kyseisen paraabelin yhtälö

$$f(x) = k_2x^2 + k_1x + k_0. \quad (13)$$

Paraabelin nollakohtien summa on  $-\frac{k_1}{k_2}$ , ja paraabelin äärikohta  $x_h$  puolestaan on nollakohtien keskiarvo eli puolet nollakohtien summasta. Siis

$$x_h = -\frac{k_1}{2k_2}$$

Sijoittamalla edellä mainitut pisteet yhtälöön 13 saadaan yhtälöryhmä

$$\begin{cases} f(a) = k_2a^2 + k_1a + k_0 \\ f(b) = k_2b^2 + k_1b + k_0 \\ f(c) = k_2c^2 + k_1c + k_0 \end{cases}.$$

Vähentämällä ensimmäinen yhtälö toisesta ja kolmannesta voidaan eliminoida parametri  $k_0$  näistä yhtälöistä, jolloin saadaan

$$\begin{cases} f(a) & = k_2a^2 + k_1a + k_0 \\ f(b) - f(a) & = k_2(b^2 - a^2) + k_1(b - a) \\ f(c) - f(a) & = k_2(c^2 - a^2) + k_1(c - a) \end{cases}.$$

Jakamalla toinen ja kolmas yhtälö parametrilla  $k_2$  (joka oletetaan nollassa eroavaksi, muuten kyseessä on suora eikä sillä ole ääriarvokohtaa) saadaan näistä

$$\begin{cases} \frac{f(b)-f(a)}{k_2} & = (b^2 - a^2) + \frac{k_1}{k_2}(b - a) \\ \frac{f(c)-f(a)}{k_2} & = (c^2 - a^2) + \frac{k_1}{k_2}(c - a) \end{cases}.$$

Vähennetään sitten toinen yhtälö kolmannesta kerrottuna vakiolla  $\frac{f(c)-f(a)}{f(b)-f(a)}$ , ja kerrotaan sitten koko yhtälö vakiolla  $f(b) - f(a)$ , jolloin kolmannesta yhtälöstä voidaan ratkaista

$$\begin{aligned} 0 &= (f(b) - f(a))(c^2 - a^2) - (f(c) - f(a))(b^2 - a^2) \\ &\quad + \frac{k_1}{k_2}((f(b) - f(a))(c - a) - (f(c) - f(a))(b - a)) \\ \frac{k_1}{k_2} &= \frac{(f(a) - f(b))(c^2 - a^2) - (f(a) - f(c))(b^2 - a^2)}{((f(b) - f(a))(c - a) - (f(c) - f(a))(b - a))}. \end{aligned}$$

Sijoittamalla tämä muuttujan  $x_h$  kaavaan saadaan

$$x_h = \frac{(f(b) - f(a))(c^2 - a^2) - (f(c) - f(a))(b^2 - a^2)}{2((f(b) - f(a))(c - a) - (f(c) - f(a))(b - a))}. \quad (14)$$

Muutetaan tämä kaava vielä muotoon, josta näkee suoraan poikkeaman kohdan  $a$  ja paraabelin huippukohdan välillä:

$$\begin{aligned} x_h &= \frac{(f(b) - f(a))(c^2 - a^2) - (f(c) - f(a))(b^2 - a^2)}{2((f(b) - f(a))(c - a) - (f(c) - f(a))(b - a))} - a + a \\ &= \frac{(f(b) - f(a))[c^2 - a^2 + 2a(a - c)] - (f(c) - f(a))[b^2 - a^2 + 2a(a - b)]}{2((f(b) - f(a))(c - a) - (f(c) - f(a))(b - a))} + a \\ &= a + \frac{1}{2} \frac{(a - c)^2[f(a) - f(b)] - (a - b)^2[f(a) - f(c)]}{(a - c)[f(a) - f(b)] - (a - b)[f(a) - f(c)]}. \end{aligned}$$

### 3.2.2 Brentin menetelmä

Brentin menetelmä ([1], s. 402-405) on paraabeliapproksimaatiota hyödyntävä numeerinen yksiulotteisen funktion optimointimenetelmä. Sitä käytetään tässä työssä yhden ulottuvuuden maksimointimenetelmänä osana edellä esiteltyä konjugaattigradienntimenetelmää. Brentin menetelmää käytettäessä kullakin iteraatiolla on muututtajat  $x_n, w_n, v_n$ . Tässä  $x_n$  on  $n$ :s arvaus funktion maksimikohdasta,  $w_n$  toiseksi suurimman lasketun arvon sijainti ja  $v_n$  muuttujan  $w_i$  edellinen luvusta  $w_n$  eroava arvo.

Kussakin iteraatiossa haetaan pisteiden  $(x_n, f(x_n)), (v_n, f(v_n))$  ja  $w_n, f(w_n)$  kautta kulkevan paraabelin huippukohta ja testataan, sopiiko se funktion uudeksi minimikohdaksi. Tällöin uusi kandidaatti  $u_{n+1}$  funktion maksimikohdaksi saadaan edellä johdetusta kaavasta

$$u_{n+1} = x_n + \frac{1}{2} \frac{(x_n - v_n)^2[f(x_n) - f(w_n)] - (x_n - w_n)^2[f(x_n) - f(v_n)]}{(x_n - v_n)[f(x_n) - f(w_n)] - (x_n - w_n)[f(x_n) - f(v_n)]}. \quad (15)$$

Funktion arvo lasketaan seuraavan arvauksen  $u_{n+1}$  kohdalla, minkä jälkeen muiden muuttujien arvot päivitetään. Ensiksi tarkistetaan, päteekö  $f(u_{n+1}) < f(x_n)$ . Jos epäyhtälö pätee, niin asetetaan

$$\begin{aligned} x_{n+1} &= u_{n+1} \\ w_{n+1} &= x_n \\ v_{n+1} &= w_n, \end{aligned}$$

minkä lisäksi päivitetään hakualue sen mukaan, onko  $u_{n+1}$  arvon  $x_n$  oikealla vai vasemmalla puolella: Jos  $u_{n+1} \geq x_n$ , niin asetetaan  $a_{n+1} = x_{n+1}, b_{n+1} = b_n$ , muussa tapauksessa  $a_{n+1} = a_n, b_{n+1} = x_n$ .

Jos taas uusin funktion arvo ei ole suurin, pidetään  $x_{n+1} = x_n$  ja päivitetään muuttujien  $a_{n+1}$  ja  $b_{n+1}$  arvoiksi  $a_{n+1} = u_{n+1}, b_{n+1} = b_n$ , jos  $u < x$  ja muussa tapauksessa  $a_{n+1} = a_n, b_{n+1} = u_{n+1}$ . Tämän jälkeen tarkistetaan, onko funktion arvo pisteessä  $u_{n+1}$  toiseksi suurin eli arvoa  $w_n$  suurempi. Jos  $u_{n+1} > w_n$  tai  $w_n = x_n$ , niin asetetaan

$$\begin{aligned} v_{n+1} &= w_n \\ w_{n+1} &= u_{n+1}, \end{aligned}$$

ja muussa tapauksessa pidetään

$$\begin{aligned}v_{n+1} &= v_n \\w_{n+1} &= w_n.\end{aligned}$$

Brentin menetelmä ei aina käyttäydy hyvin. Jos funktio ei ole likimain parabolinen muuttujien ympäristössä, uusi arvaus saattaa mennä hakualueen ulkopuolelle, ja menetelmä saattaa hajaantua. Menetelmä ei myöskään sovi tilanteisiin, joissa kaksi paraabelin sovitepisteistä on samoja, koska tällöin soviteparaabeli ei ole yksikäsitteinen. Alkuarvauspisteiden asettaminen onkin yksi suurimmista ongelmista Brentin menetelmässä. Tämän takia tässä työssä ei tukeuduta pelkästään siihen, vaan sen tukena käytetään kultaisen leikkauksen menetelmää silloin, kun soviteparaabelit eivät kelpaa maksimikohdan iteroimiseen. Tästä kerrotaan tarkemmin seuraavissa luvuissa.

### 3.2.3 Kultaisen leikkauksen menetelmä

Kultaisen leikkauksen menetelmä ([1], s.397-402) on yksinkertainen numeerinen menetelmä funktion optimoimiseksi yhdessä ulottuvuudessa. Se ei ole yhtä nopea kuin Brentin menetelmä ideaalitapauksessa, mutta se on erittäin luotettava. Tässä työssä sitä käytetään Brentin menetelmän tukena yksiulotteisen optimointitehtävän hakualueen pienentämiseksi, kunnes funktio on iteraatiopisteiden ympäristössä riittävän lähellä parabolista funktiota nopeamman Brentin menetelmän käyttämiseen.

Kultaisen leikkauksen menetelmässä pidetään yllä muuttujia  $a_n, b_n$  ja  $x_n$ , missä  $x_n$  on funktion maksimikohdan senhetkinen arvio ja  $a_n$  ja  $b_n$  rajaavat hakualueen. Kullakin  $n$  tutkitaan ensin, onko arvo  $x_n$  lähempänä arvioa  $a_n$  vai arvoa  $b_n$ . Jos  $x_n < x_{mid,n} = \frac{a_n+b_n}{2}$ , niin seuraava funktion arvon laskemiskohta  $u_{n+1}$  otetaan pisteiden  $x_n$  ja  $b_n$  välistä (tapaus 1), muussa tapauksessa pisteiden  $x_n$  ja  $a_n$  välistä (tapaus 2).

Tapauksessa 1 asetetaan  $d_{n+1} = \phi^2(a_n - x_n)$ , missä  $\phi = \frac{\sqrt{5}-1}{2}$  eli pienempi kultainen leikkaus ja  $\phi^2 = 1 - \phi = \frac{3-\sqrt{5}}{2} \approx 0,38197$ . Kultaisen leikkauksen valinta leikkaussuhteeksi optimoi menetelmän suppenemisen kohti minimikohtaa huonoimassa skenaariossa. Tällöin evaluointipisteen siirtymää kuvaava arvo  $d_{n+1}$  on negatiivinen siirtäen pistettä vasemmalle. Asetetaan sitten  $u_{n+1} = x_n + d_{n+1}$  ja lasketaan funktion arvo pisteessä  $u_{n+1}$ . Verrataan uuden arvon maksimaalisuutta: Jos  $f(u_{n+1}) \geq f(x_n)$ , niin päivitetään maksimikohdan ja hakualueen rajojen arvoiksi

$$\begin{aligned}a_{n+1} &= a_n \\x_{n+1} &= u_{n+1} \\b_{n+1} &= x_n\end{aligned}$$

ja muuten

$$\begin{aligned}a_{n+1} &= u_{n+1} \\x_{n+1} &= x_n \\b_{n+1} &= b_n.\end{aligned}$$

Tapauksessa 2 asetetaan  $d_{n+1} = \phi(b_n - x_n)$ . Tällöin funktion arvon laskemiskohdan siirtymää kuvaava arvo  $d_{n+1}$  on positiivinen siirtäen pistettä oikealle. Asetetaan sitten  $u_{n+1} = x_n + d_{n+1}$  ja lasketaan funktion arvo pisteessä  $u_{n+1}$ . Verrataan uuden arvon maksimaalisuutta: Jos  $f(u_{n+1}) \geq f(x_n)$ , niin päivitetään maksimikohdan ja hakualueen rajojen arvoiksi

$$\begin{aligned} a_{n+1} &= x_n \\ x_{n+1} &= u_{n+1} \\ b_{n+1} &= b_n \end{aligned}$$

ja muuten

$$\begin{aligned} a_{n+1} &= a_n \\ x_{n+1} &= x_n \\ b_{n+1} &= u_n. \end{aligned}$$

### 3.2.4 Yhden ulottuvuuden maksimoinnin toteutus

Lopullisessa yhden ulottuvuuden maksimointialgoritmissa sovelletaan sekä kultaisen leikkauksen menetelmää että Brentin menetelmää tarpeen mukaan. Aluksi, kun ollaan kaukana huippukohdasta ja funktion parabolisuus hakualueella on epävarmaa, lähestytään ratkaisua varmemmalla kultaisen leikkauksen menetelmällä. Kun funktio on riittävän lähellä paraabelia, siirrytään Brentin menetelmään, joka suppenee nopeammin kohti funktion maksimikohtaa.

Toteutuksessa valitaan aluksi kolme alkuarvauspistettä  $a_0, b_0$  ja  $x_0$ . Näistä  $x_0$  on alkuarvaus itse funktion maksimikohdalle ja luvut  $a_0$  ja  $b_0$  hakualueen alkurajat. Lisäksi asetetaan  $u_0 = e_0 = d_0 = 0$  ja  $v_0 = w_0 = b_0$ , ja metodin niin sanotut toleranssivakiot  $t_1$  ja  $t_2$ , joiden tarkoituksena on ohjata kelpoisten paraabelien ja arvauksien valitsemista sekä iteroinnin lopettamispäätöksen tekemistä. Nyt kullakin  $a_n, b_n, x_n, u_n, e_n, d_n, v_n, w_n$  muodostetaan paraabeli, jonka avulla funktion minimikohtaa yritetään hakea, ja iteraation loppuksi päivitetään uudet arvot  $a_{n+1}, b_{n+1}, x_{n+1}, u_{n+1}, e_{n+1}, d_{n+1}, v_{n+1}, w_{n+1}$ .

Muuttujien  $a_n$  ja  $b_n$  tarkoitus on rajoittaa maksimikohdan hakualuetta, ja nämä lähestyvät toisiaan sitä mukaa, kun iteraatioissa löytyy pienempiä funktion arvoja. Muuttuja  $x_n$  on senhetkinen arvio funktion maksimikohdasta, muuttuja  $u_n$  funktion arvon laskemiskohta iteraatiolla  $n$ , ja  $w_n$  on toiseksi suurimman lasketun arvon havaitsemiskohta, sekä  $v_n$  on muuttujan  $w_i$  kolmanneksi suurimman lasketun arvon havaitsemiskohta. Kussakin iteraatiossa haetaan pisteiden  $(x_n, f(x_n)), (v_n, f(v_n))$  ja  $w_n, f(w_n)$  kautta kulkevan paraabelin huippukohta ja testataan, sopiiko se funktion uudeksi minimikohdaksi. Sitä varten jokaisessa iteraatiossa lasketaan arvot

$$\begin{aligned} r_{n+1} &= (x_n - w_n)(f(x_n) - f(v_n)) \\ q_{n+1} &= (x_n - v_n)(f(x_n) - f(w_n)) \\ p_{n+1} &= (x_n - w_n)q_{n+1} - (x_n - v_n)r_{n+1} \\ q'_{n+1} &= 2(q_{n+1} - r_{n+1}) \\ e_{n+1} &= d_n. \end{aligned}$$



Lisäksi parametrin  $q'_{n+1}$  merkki käännetään positiiviseksi, jollei se ole jo positiivinen, ja muussa tapauksessa parametrin  $p_{n+1}$  merkki käännetään.

Seuraavaksi testataan paraabelin kelpoisuus. Paraabeli on kelpoinen, jos  $|p_{n+1}| < |\frac{q_{n+1}c_n}{2}|$ ,  $p_{n+1} > q_{n+1}(a_{n+1} - x_n)$  ja  $p_{n+1} < q(b_{n+1} - x_n)$ . Näiden ehtojen tarkoituksena on tarkistaa, onko paraabelisovituksesta laskettu piste muuttujien  $a_n$  ja  $b_n$  asettamisrajoissa. Jos paraabeli on kelpoinen, suoritetaan iteraatio Brentin menetelmällä ja asetetaan

$$d_{n+1} = \begin{cases} t_1 \operatorname{sgn}(x_n - \frac{a_n+b_n}{2}), & \text{jos } \min\{x_n + d_n - a_n, b_n - x_n - d_n\} < t_2 \\ \frac{p_{n+1}}{q_{n+1}}, & \text{muuten} \end{cases}$$

$$u_{n+1} = \begin{cases} x_n + d_{n+1}, & \text{jos } d_{n+1} > t_1 \\ x_n + t_1 \operatorname{sgn}(d_{n+1}), & \text{muuten} \end{cases}.$$

Tässä  $u_{n+1}$  on uusi kandidaatti funktion maksimikohdaksi. Sen määrittämisen jälkeen päivitetään loput muuttujat hakualueen työstämiseksi.

Jos paraabeli ei ole kelpoinen, eli paraabelin huippukohta ei ole lukujen  $a_n$  ja  $b_n$  välissä, muodostetaan seuraava arvaus kultaisen leikkauksen menetelmällä, joka esitellään edellä.

Funktion arvo lasketaan seuraavan arvauksen  $u_{n+1}$  kohdalla, minkä jälkeen muiden muuttujien arvot päivitetään. Ensiksi tarkistetaan, päteekö  $f(u_{n+1}) > f(x_n)$ . Jos epäyhtälö pätee, niin asetetaan

$$x_{n+1} = u_{n+1}$$

$$w_{n+1} = x_n$$

$$v_{n+1} = w_n,$$

minkä lisäksi päivitetään hakualue sen mukaan, onko  $u_{n+1}$  arvon  $x_n$  oikealla vai vasemmalla puolella: Jos  $u_{n+1} \geq x_n$ , niin asetetaan  $a_{n+1} = x_{n+1}$ ,  $b_{n+1} = b_n$ , muussa tapauksessa  $a_{n+1} = a_{n+1}$ ,  $b_{n+1} = x_n$ .

Jos taas uusi funktion arvo ei ole suurin, pidetään  $x_{n+1} = x_n$  ja päivitetään muuttujien  $a_{n+1}$  ja  $b_{n+1}$  arvoiksi  $a_{n+1} = u_{n+1}$ ,  $b_{n+1} = b_n$ , jos  $u < x$  ja muussa tapauksessa  $a_{n+1} = a_n$ ,  $b_{n+1} = u_{n+1}$ . Tämän jälkeen tarkistetaan, onko funktion arvo pisteessä  $u_{n+1}$  toiseksi suurin eli arvoa  $f(w_n)$  suurempi.

Jos  $f(u_{n+1}) \geq f(w_n)$  tai  $w_n = x_n$ , niin asetetaan

$$v_{n+1} = w_n$$

$$w_{n+1} = u_{n+1},$$

muuten pidetään  $w_{n+1} = w_n$  ja tarkistetaan vielä, päteekö jos  $f(u_{n+1}) \geq f(v_n)$ . Jos tämä epäyhtälö on voimassa tai  $v_n = w_n$  tai  $v_n = x_n$ , niin asetetaan

$$v_{n+1} = u_n$$

$$w_{n+1} = w_n.$$

Muussa tapauksessa pidetään

$$v_{n+1} = v_n$$

$$w_{n+1} = w_n.$$

## 4 Paikannus

### 4.1 Alustus

Itse paikannusalgoritmissa on kaksi vaihetta: alustus ja tarkennus. Alustusvaiheessa pyritään muodostamaan ankkureita hyödyntämällä hyvä alkuarvaus laitteiden sijainneille. Alkuarvaus syötetään tarkennusalgoritmille, joka määrittää laitteiden sijainnit edellä kuvattuja uskottavuusfunktioita ja konjugaattigradienttialgoritmia käyttämällä.

Alustusta varten oletetaan, että verkossa on  $N$  laitetta, joista  $n$  on *ankkureita*, jotka tietävät oman sijaintinsa. Muut laitteet ovat *sokkolaitteita*, joiden sijaintia halutaan estimoida. Ankkurien indeksit ovat  $1, \dots, n$  ja sokkolaitteiden  $n + 1, \dots, N$ .

Paikkaestimaatit alustetaan uskottavuusfunktion avulla käyttämällä vain niistä linkeistä mitattuja tehohäviöitä, joissa toinen laite on ankkuri ja toinen sokkolaite. Jos ankkureita on kuitenkin harvassa, eikä niitä ole riittävästi (kolme kappaletta) jokaisen estimoitavan laitteen havainnointietäisyydellä, voidaan joutua käyttämään ankkureita, joita paikannettava laite ei havainnut. Tätä varten laajennetaan tehohäviömittauksia niinsanottua hyppyetäisyyttä käyttämällä, jotta kaikkien laitteiden välille saataisiin jokin path loss -arvo.

Olkoon siis  $L_{i,j}$  laitteiden  $i$  ja  $j$  välillä havaittu tehohäviö, jos laitteet havaitsivat toisensa. Olkoon sitten *hyppyetäisyys* rekursiivisesti määriteltynä

- $h(i, j) = 0$ , jos  $i = j$
- $h(i, j) = n$ , jos jokin laite  $k$ , jolla  $h(i, k) = n - 1$  ja laite  $i$  havaitsivat toisensa, eikä mikään laite  $m$ , jolla  $h(i, k) < n - 1$  havaitselaitetta  $i$ .

Tällöin laitteen hyppyetäisyys itsensä kanssa on nolla ja laitteiden, jotka havaitsivat toisensa, välinen etäisyys on 1. Toistensa kantamattomissa olevien laitteiden etäisyys on vähintään 2. Olkoon  $P_m$  suurin tehohäviö, jolla laitteet voivat vielä havaita toisensa ja  $d_m$  etäisyys, jolla (logaritmisen) tehohäviön odotusarvo on  $P_m$ , siis  $P_m = 10\gamma \lg(\frac{d_m}{d_0}) + P_0$ . Voidaan tehdä yksinkertainen heuristinen päätelmä, että tiheässä verkossa hyppyetäisyyden ollessa  $h > 1$ , etäisyys on suurella todennäköisyydellä välillä  $(h - 1)d_m < d < hd_m$ . Nimittäin tiheässä verkossa laitteiden  $i$  ja  $j$  välisen suoran linjan läheltä löytyy laitteita likimain etäisyyden  $d_m$  välein. Ottamalla näistä keskiarvo voidaan alustusta varten arvata laitteiden  $i, j$  etäisyydeksi

$$d(i, j) = \frac{2h - 1}{2} d_m.$$

Tämän oletuksen pätiessä saadaan tehohäviöksi

$$\begin{aligned} P_{ij} = P_{ji} &= 10\gamma \lg\left(\frac{2h(i, j) - 1}{2} \frac{d_m}{d_0}\right) + P_0 \\ &= 10\gamma \lg\left(\frac{2h(i, j) - 1}{2}\right) + 10\gamma \lg\left(\frac{d_m}{d_0}\right) + P_0 \\ &= 10\gamma \lg\left(\frac{2h(i, j) - 1}{2}\right) + P_m. \end{aligned}$$

Nyt voidaan määritellä laajennetut tehohäviöhavainnot  $L'_{ij}$ . Kullakin  $i, j$  joilla  $i \neq j$ :

$$P'_{ij} = \begin{cases} P_{ij}, & \text{jos } h(i, j) = 1 \\ 10\gamma \lg\left(\frac{2h(i, j)-1}{2}\right) + P_m, & \text{muuten.} \end{cases}$$

Seuraavaksi on huomattava, että arvot  $P_{ij}$  linkeissä, joissa laitteet havaitsivat toisensa, ovat paljon luotettavampia kuin kuin arviot  $P'_{ij}$  linkeissä, joissa mitattua tehohäviötä ei ole. Siksi havainnottomia arvoja  $P'_{ij}$  käytetään vain pakon edessä, ja jos oikeita tehomittauksia on riittävästi, turvaudutaan pelkästään niihin. Vastaavasti havainnottomissa linkeissä, joissa hyppyetäisyys on suurempi, arvio muuttuu epäluotettavammaksi, ja samanvahvuisia arvoja  $P'_{ij}$  halutaan käyttää mahdollisimman paljon, erivahvuisia taas mahdollisimman vähän.

Olkoot sitten kullakin sokkolaiteindeksillä  $i$  indekset  $m_i(j)$  ankkuri-indeksien  $1, \dots, n$  permutaatio, joka on järjestetty seuraavilla kriteereillä:

- Jos  $h(i, j) < h(i, k)$ , niin  $m_i(j) < m_i(k)$ .
- jos  $h(i, j) = h(i, k) = 1$  ja  $P_{ij} < P_{ik}$ , niin  $m_i(j) < m_i(k)$ .
- muutoin jos  $j < k$ , niin  $m_i(j) < m_i(k)$ .

Nyt konjugaattigradienttialgoritmeille syötetään parhaat kolme ankkurilaitetta, siis ankkurilaitteet indekseissä  $m_i(1), m_i(2)$  ja  $m_i(3)$ . Lisäksi ennen varsinaista uskottavuusfunktion maksimointia lasketaan alkuarvaus vielä alustusvaiheessa syöttämällä laitteet ensin luvussa 2.3.2 esiteltyyn trilateraattoriin.

## 4.2 Tarkennus

Olkoon edelleen  $p_{ij}$  mitattu tehohäviö laitteiden  $i$  ja  $j$  välillä ja  $(\mathbf{x}_0, \mathbf{y}_0)$  alustusalgoritmin antama alkuarvausmatriisi laitteiden sijainneille. Tällöin kustakin estimaatista  $(\mathbf{x}_k, \mathbf{y}_k)$  lasketaan rekursiivisesti uusi estimaatti  $(\mathbf{x}_{k+1}, \mathbf{y}_{k+1})$  seuraavasti:

Kutakin sokkolaiteindeksiä  $n+1 \leq i \leq N$  kohti määritellään uskottavuusfunktio

$$L_k(x_i, y_i) = L(x_i, y_i | x_{j,k}, y_{j,k}, p_{ji}; j \in \mathcal{D}(i))$$

laitteen  $i$  sijainnille kuten osiossa 2 on kuvattu. Ankkuri-indeksien  $1, \dots, n$  estimaatit puolestaan pysyvät aina vakiona. Tällöin uudessa estimaatissa asetetaan

$$(x_{i,k+1}, y_{i,k+1}) = (\hat{x}_i, \hat{y}_i) = \arg \max(L_n(x_i, y_i)),$$

missä maksimointi suoritetaan edellä kuvatulla konjugaattigradienttialgoritmeilla. Estimaatteja  $(\mathbf{x}_k, \mathbf{y}_k)$  lasketaan tässä työssä aina tietty lukumäärä  $I$ , jota kutsutaan iteraatiolukumääräksi. Toinen tapa voisi olla toistaa iteraatiota kunnes peräkkäisten iteraatitien välinen ero on riittävän pieni.

## 5 Lopputulokset

### 5.1 Simulointi

Edellä kuvattua algoritmia simuloitiin kirjoittajan itse tekemällä Python-ohjelmalla, joka löytyy julkisesti osoitteesta

<https://gitlab.utu.fi/tajotai/position-estimator>. Python-ohjelma generoi neliön  $[-L, L] \times [-L, L]$  sisään satunnaisesti  $N$  laitetta, joista  $N_a$  on ankkureita. Ohjelma myös laskee kunkin kahden laitteen välisen etäisyyden ja arpoo niiden välisen tehohäviön (desibeleinä) parametrien  $P_0, \gamma$  ja  $\sigma$  mukaisesti normaalijakaumasta. Ohjelmassa oleva estimaattori estimoi sitten arvottujen tehohäviöiden pohjalta sokkolaitteiden paikat edellä kuvatulla tavalla.

Estimaattorin virhe mitataan useiden iteraatioiden yli laskemalla laitteiden paikoista keskineliövirheet. Toisin sanoen, jos kunkin laitteen  $i$  todellinen sijainti on  $(x_i, y_i)$  ja estimoitu sijainti  $(\hat{x}_i, \hat{y}_i)$ , niin estimaattorin keskineliövirheeksi lasketaan

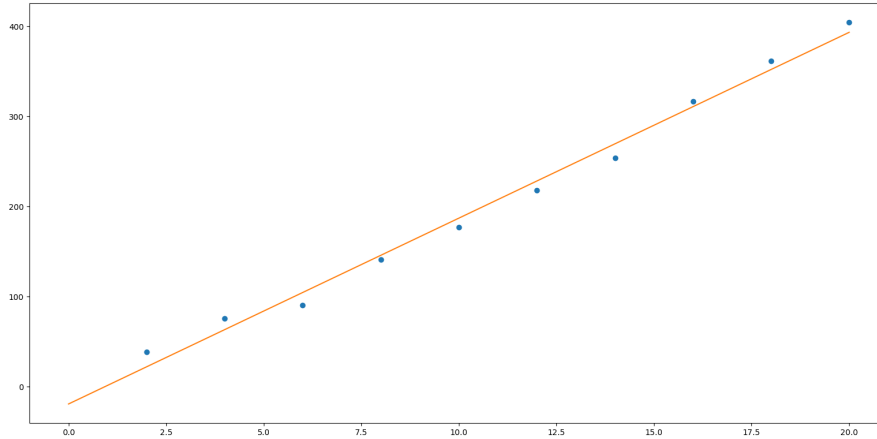
$$E = \frac{1}{N} \sum_{i=1}^N [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2].$$

Haluttaessa arvo laitteiden sijainnin keskimääräiselle virheelle, otetaan keskineliövirheestä neliöjuuri ja määritellään RMS-virhe  $\varepsilon = \sqrt{E}$ .

Työn tuloksia varten ajettiin viisi simulaatiota. Kussakin simulaatiossa verkon ominaisuuksia kuvaa neljä parametria:  $N$  eli laitteiden lukumäärä,  $q$  eli ankkurilaitteiden osuus prosentteina,  $r$  eli pienintä havaittavissa olevaa tehoa vastaava etäisyys metreinä ja ympäristön hajontaparametri  $\sigma$ . Kussakin simulaatiossa yksi parametri käy läpi joukon arvoja muiden parametrien pysyessä kiinnitettynä, ja muuttuvan parametrin kutakin arvoa kohti ajetaan 20 ottoa verkon sijaintiestimaatista. Verkon laajuutta kuvaava parametri  $L$  saa arvon 1000 kaikissa simulaatioissa. TehohäviöekspONENTTI  $\gamma$  saa simulaatioissa arvon 2, sillä se on tehohäviöekspONENTIN arvo "vapaassa avaruudessa" eli kahden radiolaitteen välillä, kun välissä on pelkkää ilmaa eikä esimerkiksi puita tai rakennuksia häiritsemässä.

Ennako-odotuksina estimaattien odotettiin huononevan hajontaparametrin  $\sigma$  kasvaessa ja paranevan minimitehoetäisyyden  $r$ , laitteiden lukumäärän  $N$ , ankkurisuuden  $n$  ja iteraatiolukumäärän  $I$  kasvaessa.

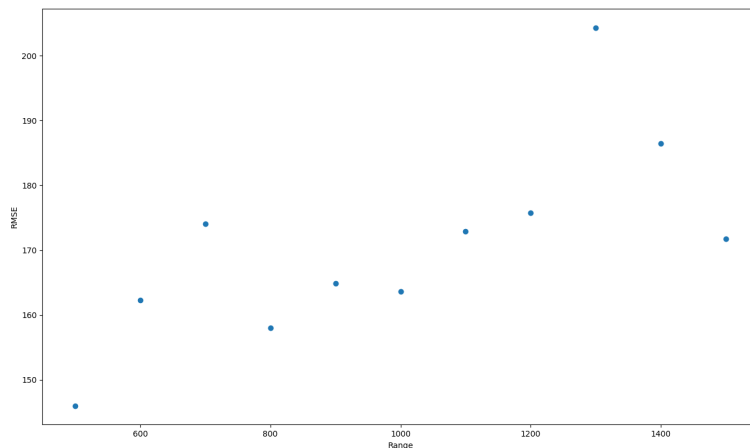
Ensimmäisessä simulaatiossa kiinnitetyt parametrit olivat  $N = 200$ ,  $q = 10\%$ ,  $r = 1000$  ja  $I = 20$ , ja parametri  $\sigma$  sai arvot 2, 4, 6, 8, 10, 12, 14, 16, 18 ja 20. Laitteiden sijaintien RMS-virhe parametrin  $\sigma$  funktiona esitetään kuvassa 3.



Kuva 3: RMS-virhe hajontaparametrin  $\sigma$  funktiona annetuilla parametreilla

Ensimmäisessä simulaatiossa pisteiden havaitaan olevan likimain lineaarisella suoralla, joten havaintopisteisiin on lisäksi sovitettu lineaarinen pienimmän neliösumman estimaatti komennolla `np.linalg.lstsq`. Suoran yhtälö on kahden desimaalin tarkkuudella  $y = 20,62x - 19,08$ .

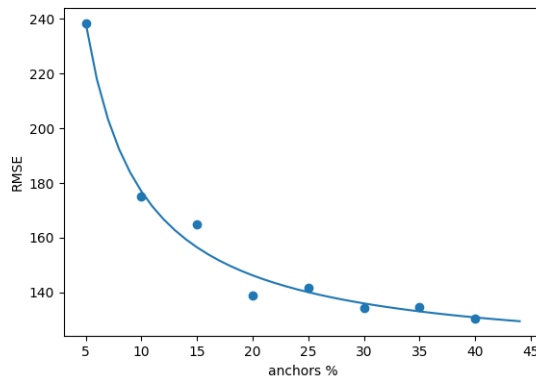
Toisessa simulaatiossa kiinnitetyt parametrit olivat  $N = 200$ ,  $q = 10\%$ ,  $\sigma = 10$  ja  $I = 20$ , ja parametri  $r$  sai arvot 500, 600, 700, 800, 900, 1000, 1100, 1200, 1300, 1400, 1500 sekä  $\infty$ . Laitteiden sijaintien RMS-virhe parametrin  $r$  funktiona esitetään kuvassa 4. Äärettömän etäisyyden ajon tulos ei kuitenkaan näy kuvassa, mutta sen rms-virhe oli 242,635.



Kuva 4: RMS-virhe maksimietäisyyden  $r$  funktiona annetuilla parametreilla

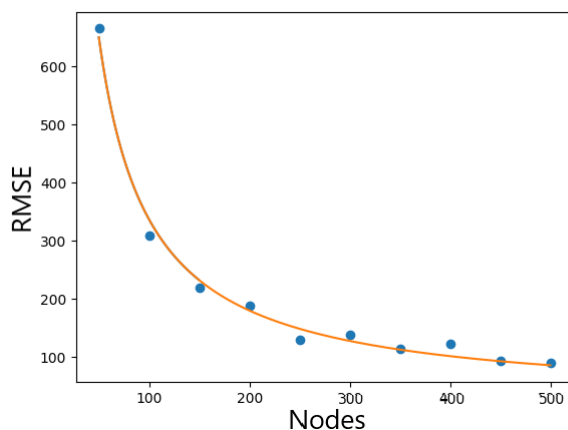
Kolmannessa simulaatiossa kiinnitetyt parametrit olivat  $N = 200$ ,  $r = 1000$ ,  $\sigma = 10$  ja  $I = 20$ , ja parametri  $q$  sai arvot 5%, 10%, 15%, 20%, 25%, 30%, 35% ja 40%. Laitteiden sijaintien RMS-virhe sekä pienimmän neliösumman muotoa  $y = \frac{a}{x} + b$

oleva hyperbolinen estimaatti parametrin  $q$  funktiona esitetään kuvassa 5. Hyperbolinen sovitefunktio muodostettiin pienimmän neliösumman estimaatin avulla. Sovitefunktion lauseke on  $y = 614,9\frac{1}{x} + 115,5$ .



Kuva 5: RMS-virhe ankkuriosuuden  $q$  funktiona annetuilla parametreilla

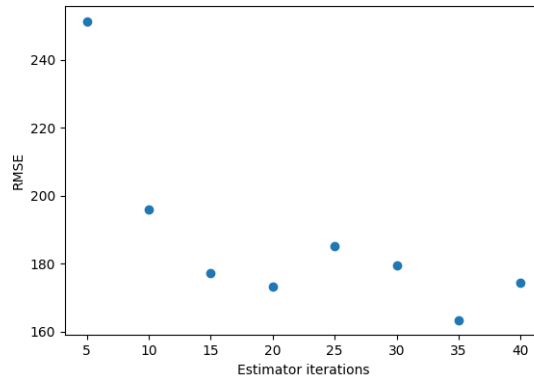
Neljännessä simulaatiossa kiinnitetyt parametrit olivat  $q = 10\%$ ,  $r = 1000$ ,  $\sigma = 10$  ja  $I = 20$ , ja parametri  $N$  sai arvot 50, 100, 150, 200, 250, 300, 350, 400, 450 ja 500. Koska laitelukumäärän lisääminen johtaa simulaation keston pidentymiseen, tässä simulaatiossa iteraatioita ajettiin poikkeuksellisesti seuraavalla tavalla: arvolla  $N = 50$  iteraatioita oli 100, arvolla  $N = 100$  iteraatioita oli 50, arvolla  $N = 150$  iteraatioita oli 30, arvoilla  $N = 200$  ja  $N = 250$  iteraatioita oli 20 ja muilla parametrin  $N$  arvoilla iteraatioita oli 10. Laitteiden sijaintien RMS-virhe parametrin  $q$  funktiona esitetään kuvassa 6. Kuvassa on myös hyperbolinen sovite, joka on muodostettu samoin kuin kolmannessa simulaatiossa. Funktion lauseke on  $y = 31395\frac{1}{x} + 22,6$ .



Kuva 6: RMS-virhe laitelukumäärän  $N$  funktiona annetuilla parametreilla

Viidennessä simulaatiossa kiinnitetyt parametrit olivat  $N = 200$ ,  $q = 10\%$ ,  $r = 1000$ ,  $\sigma = 10$ , ja iteraatioiden lukumäärä  $I$  sai arvot 5, 10, 15, 20, 25, 30, 35 ja 40.

Laitteiden sijaintien havaitut RMS-virheet parametrin  $I$  funktiona esitetään kuvassa 7. Sovitetta ei tehty, sillä havainnot olivat sen verran hajaantuneet.



Kuva 7: RMS-virhe iteraatiolukumäärän  $I$  funktiona annetuilla parametreilla

## 5.2 Omat päätelmät

Simulaatioiden perusteella estimaattori toimii, joskin paikannus on melko epätarkkaa. Parametrin  $\sigma$  kasvattaminen kasvatti paikannuksen virhettä lineaarisesti, ja näin tulos on muutoksen suunnan kannalta odotetunlainen. Lineaarisuudessa on myös järkeä, sillä riittävän suuresta summasta muodostuva log-uskottavuusfunktion maksimi noudattaa tilastollisen päättelyn kurssin luentomonisteen ([6], s. 32) mukaan likimain normaalijakaumaa, jonka keskihajonta on suoraan verrannollinen tiheysfunktion keskihajontaan, joka puolestaan on  $\sigma$ .

Odotetusti ankkurien osuuden kasvattaminen parantaa paikannuksen tarkkuutta, mikä on luontevaa, sillä ankkurien paikasta on varma tieto. Ankkurien määrän kasvu tekee näin paikannuksesta luotettavampaa. Ankkurien määrän osalta tämä tarkkuuden kasvu kuitenkin hidastui laitelukumäärän kasvaessa suureksi ja näytti jumittuvan tiettyyn raja-arvoon, noudattaen käännteistä verrantoa vakiotermejä vaille (tuo vakiotermi on kyseinen raja-arvo, johon tarkkuus jumittuu). On järkevä seuraus, että tarkkuus ei suppene nollaan ankkuriosuuden kasvaessa vaan juuttuu tiettyyn arvoon, sillä yksittäistä laitetta ei pystyttäisi paikantamaan täysin tarkasti ympäristön aiheuttamien satunnaisvirheiden takia, vaikka kaikki muut laitteet verkossa olisivat ankkureita.

Samoin laitteiden määrän lisääntyminen kasvatti odotetusti tarkkuutta, eli estimaattori pystyy hyödyntämään näin laitteiden lisääntymisen johdosta lisääntyneen informaation. Laitteiden määrän kasvaessa voisi uskoa virheen suppenevan kohti nolaa, sillä useampi laite tarkoittaa myös suurempaa määrää informaatiota. Virheelle laitelukumäärän funktiona annettu hyperbolinen sovite näyttäisi kertovan, että tässäkin tilanteessa tarkkuus jumittuu tiettyyn raja-arvoon, mutta toisaalta tämä raja-arvo on pienempi kuin ankkuriosuuden tapauksessa, ja hyperbolinen funktio ei välttämättä ole tarkkaan oikeanlainen funktio kuvaamaan tarkkuuden parantumista, vaikka se sopiikin havaintoihin hyvin.

Neljännessä simulaatiossa kokeiltiin myös sovitetta, jossa RMS-virhe on kääntäen verrannollinen laitelukumäärän neliöjuureen, sillä yksittäisen laitteen havaitse-

mien laitteiden lukumäärä on suoraan verrannollinen laitteiden lukumäärään verkossa, ja jokainen havaittu laite taas pidentää uskottavuusfunktion summalausekettä. Riippumattomien satunnaismuuttujien summan pituus puolestaan on kääntäen verrannollinen sen keskihajonnan neliöön. Näin neliöjuurilausekkeelle olisi teoreettista pohjaa, mutta käytännössä tämä sovite ei kuitenkaan tuottanut parhaiten sopivaa käyrää.

Yllättäen havaintoetäisyyksien kasvaessa tarkkuus ei kuitenkaan parantunut. Kuvaajien perusteella se näyttäisi jopa huonontuneen, mutta toisaalta kuvassa on niin paljon hajontaa, että asiaa ei voida ihan varmasti todeta. Intuitiivisesti informaation pitäisi lisääntyä, kun yhä enemmän muita laitteita pystytään havaitsemaan, mutta ehkä huonolaatuiset havainnot eivät juurikaan auta estimaatin tarkentamisessa. Voi olla, että paikannuksessa kannattaakin hyödyntää etupäässä niitä laitteita, joiden havaittiin olleen lähellä.

Tutkimuksen oletuksista on syytä huomata, että ne olivat erittäin yksinkertaiset ja rajatut. Parametrien  $\gamma$ ,  $P_0$  ja  $\sigma$  ei voi tositilanteessa aina olettaa olevan vakioita koko verkossa, eikä laitteet tunne niitä ellei niitä käydä manuaalisesti mittaamassa. Mahdolliset jatkotutkimukset voisivat suuntautua paikannukseen tilanteessa, jossa nämä parametrit ovat tuntemattomat, tai muuttuvat verkon sisällä.

## Viitteet

- [1] W.H.Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery: Numerical Recipes in C. Cambridge University Press, toinen painos, 1992
- [2] N. Patwari, R.J. O’Dea, Y. Wang: Relative Location in Wireless Networks. IEEE VTS 53rd Vehicular Technology Conference, Proceedings (Cat. No.01CH37202), 2001, pp. 1149-1153 vol.2
- [3] Hughes Systique: Positioning techniques for mobile devices in LTE. Blogiteksti. <https://www.hsc.com/resources/blog/positioning-techniques-for-mobile-devices-in-lte/>
- [4] G. Blewitt: Basics of the GPS Technique: Observation Equations. Luku kokoelmasta Geodetic Applications of GPS, National Land Survey, Sweden, 1997
- [5] A. Bensky: Short-range Wireless Communication. Elsevier, kolmas painos, 2019
- [6] P. Nieminen: Tilastollinen päättely I & II, syksy 2021