



**TURUN  
YLIOPISTO**

SIMPLEX-ALGORITMISTA

Akseli Hulkko

LuK-tutkielma  
Helmikuu 2025

Ohjaaja:  
Stefan Emet

MATEMATIIKAN JA TILASTOTIETEEN LAITOS

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

TURUN YLIOPISTO  
Matematiikan ja tilastotieteen laitos

AKSELI HULKKO: Simplex-algoritmista  
LuK-tutkielma, 13 s.  
Matematiikka  
Helmikuu 2025

---

Tässä tutkielmassa esitetään matemaattisen optimoinnin yksi tunnetuimmista algoritmeista, simplex-algoritmi, sekä sen pohjalta kehitetty parannettu simplex-menetelmä. Tutkielmassa esitetään miten lineaarista optimointiongelmää voi muokata. Tämän jälkeen esitetään kaksivaiheinen simplex-algoritmi, jota voidaan käyttää sallitun kantaratkaisun etsimiseen. Lisäksi esitetään simplex-algoritmin ja parannetun simplex-menetelmän algoritmit, ja ratkaistaan niiden avulla lineaarinen optimointiongelma.

Asiasanat: simplex-algoritmi, parannettu simplex-menetelmä, redusoitu kustannus, kantamuuttuja.



# Sisällys

<b>1</b>	<b>Johdanto</b>	<b>1</b>
<b>2</b>	<b>Lineaarisesta optimoinnista</b>	<b>1</b>
2.1	Kaksivaiheinen simplex-algoritmi . . . . .	2
<b>3</b>	<b>Simplex-algoritmi</b>	<b>4</b>
3.1	Simplex-algoritmin askeleet . . . . .	4
3.2	Ongelman ratkaisu simplex-algoritmilla . . . . .	5
<b>4</b>	<b>Parannettu simplex-menetelmä</b>	<b>8</b>
4.1	Parannetun simplex-menetelmän algoritmi . . . . .	9
4.2	Ongelman ratkaisu parannetulla simplex-menetelmällä . . . . .	9
<b>5</b>	<b>Yhteenveto</b>	<b>13</b>



# 1 Johdanto

Lineaarinen optimointi on keskeinen matematiikan ja operaatioanalyysin osa-alue, jossa pyritään löytämään paras mahdollinen ratkaisu kohdefunktiolle yhtälörajoitteiden määräämässä sallitussa alueessa. Yksi tunnetuimmista algoritmeista näiden ongelmien ratkaisemiseen on George Dantzigin 1940-luvulla kehittämä simplex-algoritmi [5]. Se kehitettiin juuri teollistuvaan maailmaan ja varsinkin sodan jälkeinen aika-kausi esitti monimutkaisia logistiikka- ja tuotantohaasteita, jotka edellyttivät matemaattisia ratkaisuja. Simplex-algoritmi ratkaisi nämä haasteet ja sen käyttö levisi yhä laajemmalle. Tietotekniikan kehittyessä simplex-algoritmi saatiin ohjelmoitua tietokoneelle, jolloin se pystyi ratkaisemaan yhä suurempia ja monimutkaisempia ongelmia. Tällöin kuitenkin huomattiin sen rajoitukset, ja varsinkin sen vaatima suuri laskentakapasiteetti todettiin ongelmaksi. Näiden rajoitteiden selvittämiseksi vuonna 1965 simplex-algoritmista kehitettiin parannettu simplex-menetelmä [2], jota lineaariseen optimointiin keskittyneet ohjelmistot, kuten CPLEX, saattavat edelleen käyttää. Tämän lisäksi muutkin valmisohjelmistot saattavat käyttää simplex-algoritmia pohjana, johon voidaan yhdistää muita optimointikeinoja. Tämän avulla saadaan valmisohjelmisto, joka on todella tehokas tietyn tyyppisissä optimointiongelmissa.

Tässä työssä käydään läpi simplex-algoritmi, parannettu simplex-menetelmä ja ratkaistaan lineaarinen optimointiongelma algoritmeja käyttäen. Aluksi luvussa 2 käydään läpi lineaarista optimointia ja esitellään kaksivaiheinen simplex-algoritmi, joiden avulla kaikki lineaariset optimointiongelmat saadaan muotoon, johon simplex-algoritmia voi soveltaa. Luvussa 3 esitetään simplex-algoritmi ja luvussa 4 esitetään parannettu simplex-menetelmä. Lopuksi luvussa 5 on työn yhteenveto.

## 2 Lineaarista optimoinnista

Linearisessa optimoinnissa kohdefunktio on aina unimodaalinen ja lineaariset rajoitukset määräävät konveksin joukon. Tästä seuraa, että lokaali optimi on aina globaali optimi [1]. Tällöin ei tarvitse keskittyä optimaalisuusehtoihin niin tarkasti kuin muissa matemaattisen optimoinnin osa-alueissa. Linearisessa optimointitehtävässä määrätään päätösmuuttujille  $x_j$  arvot, joilla kohdefunktio

$$c_1x_1 + c_2x_2 + \dots + c_nx_n$$

maksimoidaan tai minimoidaan. Lineaarisen optimointitehtävän rajoitukset, jotka sitovat päätösmuuttujia, ovat muotoa

$$\begin{aligned} a_1x_1 + a_2x_2 + \dots + a_nx_n &\geq b \\ a_1x_1 + a_2x_2 + \dots + a_nx_n &= b \\ a_1x_1 + a_2x_2 + \dots + a_nx_n &\leq b. \end{aligned}$$

Näiden lisäksi on etumerkkirajoitusehto  $x_j \geq 0$ , joka yleensä merkitään erikseen. Simplex-algoritmia käytetään, kun tehtävä on standardimuodossa, joka yleisesti saa-

daan lisäämällä epäyhtälörajoituksiin ylimääräiset muuttujat, eli muoto

$$\begin{aligned} \max \quad & \sum_{j=1}^n c_j x_j \\ \text{s. t.} \quad & \sum_{j=1}^n a_{ij} x_j \leq b_j, \quad i = 1, \dots, m \\ & x_j \geq 0, \quad j = 1, \dots, n \end{aligned}$$

voidaan muuttaa standardimuotoon

$$\begin{aligned} \max \quad & \sum_{j=1}^n c_j x_j \\ \text{s. t.} \quad & \sum_{j=1}^n a_{ij} x_j + s_j = b_j, \quad i = 1, \dots, m \\ & x_j, s_j \geq 0, \quad j = 1, \dots, n, \end{aligned} \tag{1}$$

joka on matriisimuotoon kirjoitettuna

$$\begin{aligned} \max \quad & \mathbf{c}^T \mathbf{x} \\ \text{s. t.} \quad & \mathbf{A} \mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq 0. \end{aligned}$$

Standardimuodossa simplex-algoritmia on helpoin käyttää, sillä siitä on helpoin selvittää, onko jokin epäyhtälörajoitus aktiivinen vai ei.

Lineaarisen optimointitehtävän sallittu piste on reunapiste, jos ainakin yksi epäyhtälörajoituksista toteutuu siinä yhtäsuuruutena, ja sisäpiste, jos kaikki epäyhtälörajoitukset toteutuvat aidosti [1]. Simplex-algoritmi hakee tehtävän ratkaisua käymällä läpi näitä reunapisteitä. Reunapisteissä jokin epäyhtälörajoitus on aktiivinen, ja näistä voidaan muodostaa lineaarisia yhtälöryhmiä, joiden avulla saadaan ratkaistua tuntemattomat muuttujat. Nollaan kiinnitettyjä muuttujia kutsutaan ei-kantamuuttujiksi, ja muuttujia, jotka selvitetään yhtälöryhmistä kutsutaan kantamuuttujiksi.

Standardimuotoisen lineaarisen optimointitehtävän kantaratkaisun pitää olla sallittu, eli ratkaisun jokaisen komponentin pitää olla ei-negatiivinen. Tämä pätee myös ensimmäiseen ratkaisuun, josta simplex-algoritmia lähdetään iteroimaan. Yleensä ensimmäinen sallittu kantaratkaisu saadaan asettamalla kohdefunktion muuttujat ei-kantamuuttujiksi, eli lähdetään hakemaan ratkaisua origosta, jolloin ylimääräiset muuttujat saavat yhtälörajoituksen oikeanpuoleisen arvon. Ensimmäisen kantaratkaisun selvittäminen ei kuitenkaan aina ole niin yksinkertaista, jolloin voidaan käyttää kaksivaiheista simplex-algoritmia.

## 2.1 Kaksivaiheinen simplex-algoritmi

Kaikissa optimointitehtävissä ei ole selvää, mikä on ensimmäinen sallittu kantaratkaisu. Sen selvittämiseksi voidaan luoda toinen lineaarinen optimointitehtävä, jonka

ratkaisuksi saadaan alkuperäisen lineaarisen optimointitehtävän jokin sallittu kantaratkaisu. Tämä on kaksivaiheisen simplex-algoritmin perusidea. Tarkastellaan ajatusta vielä tarkemmin esimerkin (1) avulla.

**Esimerkki 1.** Olkoon lineaarinen optimointitehtävä (lähde: [4])

$$\begin{aligned} \min \quad & 6x_1 + 3x_2 \\ \text{s. t.} \quad & x_1 + x_2 \geq 1 \\ & 2x_1 + x_2 \geq 1 \\ & 3x_2 \leq 2 \\ & x_1, x_2 \geq 0. \end{aligned}$$

Muutetaan tehtävä maksimointitehtäväksi ja otetaan käyttöön lisämuuttujat  $s_1, s_2$  ja  $s_3$ , jotta saadaan muutettua tehtävä standardimuotoon:

$$\begin{aligned} \max \quad & -6x_1 - 3x_2 \\ \text{s. t.} \quad & x_1 + x_2 - s_1 = 1 \\ & 2x_1 + x_2 - s_2 = 1 \\ & 3x_2 + s_3 = 2 \\ & x_1, x_2, s_1, s_2, s_3 \geq 0. \end{aligned}$$

Nyt tehtävän kantaratkaisu on  $x^0 = (x_1, x_2, s_1, s_2, s_3) = (0, 0, -1, -1, 2)$ , joka ei ole sallittu kantaratkaisu, koska jokainen komponentti ei ole ei-negatiivinen. Tehtävään voidaan lisätä ylimääräinen muuttuja  $y$  yhtälöihin, joissa yli-alijäämämuuttujilla on eri etumerkki kuin oikean puolen vakioilla. Ylimääräiset muuttujat myös korvaavat kohdefunktion. Näin ollen saadaan tehtävä muotoon

$$\begin{aligned} \min \quad & y_1 + y_2 \\ \text{s. t.} \quad & x_1 + x_2 - z_1 + y_1 = 1 \\ & 2x_1 + x_2 - z_2 + y_2 = 1 \\ & 3x_2 + z_3 = 2 \\ & x_1, x_2, z_1, z_2, z_3, y_1, y_2 \geq 0. \end{aligned}$$

Nyt tehtävän kantaratkaisu on  $x^0 = (0, 0, 0, 0, 2, 1, 1)$ , joka on sallittu kantaratkaisu. Jos alkuperäinen optimointitehtävä on toteutettavissa, niin on olemassa kantaratkaisu, jolla  $y_1 = y_2 = 0$  ja siinä pisteessä on alkuperäisen optimointitehtävän sallittu kantaratkaisu.

Tiivistettynä kaksivaiheinen simplex-algoritmi toimii seuraavalla tavalla.

### Algoritmi 1. Kaksivaiheinen simplex-algoritmi

**Askel 0.** (*Aputehtävä*) Jos alkuperäisellä optimointitehtävällä on helposti löytyvä sallittu lähtökanta, niin muodostetaan se ja ratkaistaan tehtävä tavallisesti simplex-algoritmillä. Muuten muodostetaan aputehtävä esimerkin(1) tavalla.

**Askel 1.** (*Vaihe 1*) Minimoidaan aputehtävän apumuuttujien summaa simplex-algoritmillä (2) alkaen aputehtävän sallitusta lähtökannasta.

**Askel 2.** (*Ei-sallittavuus*) Jos aputehtävän kohdefunktion arvo on iteroinnin jälkeen suurempi kuin 0, lopetetaan, sillä alkuperäisellä optimointitehtävällä ei ole sallittuja pisteitä. Muuten muodostetaan alkuperäisen optimointitehtävän sallittu lähtökanta aputehtävän ratkaisusta.

**Askel 3.** (*Vaihe 2*) Muodostetun lähtökannan avulla palataan optimoimaan alkuperäistä standardimuotoista optimointitehtävää.

### 3 Simplex-algoritmi

Simplex-algoritmin toimintaperiaate perustuu iteratiiviseen prosessiin, jossa aluksi valitaan sallittu aloituspiste, ja sitten edetään askel askeleelta kohti optimia. Jokaisessa iteroinnissa tarkastellaan yhtälörajoitteiden muodostaman polyedrin naapurikärkeä ja valitaan niistä se, joka parhaiten parantaa tavoitefunktion arvoa. Tätä prosessia jatketaan, kunnes optimaalinen ratkaisu saavutetaan tai todetaan, että ongelma ei ole ratkaistavissa. Simplex-algoritmi on todella käytännöllinen, ja se toimii parhaiten pienille ja keskikokoisille ongelmille. Vaikka simplex-algoritmi on käytännöllinen ja tehokas monissa tilanteissa, sen suorituskykyä rajoittavat tietyt tekijät. Esimerkiksi algoritmin vaatima muistin ja laskentatehon määrä kasvaa nopeasti ongelman koon ja muuttujien lukumäärän kasvaessa. Tämä tekee algoritmista vähemmän sopivan erittäin suurille lineaarisille optimointitehtäville.

#### 3.1 Simplex-algoritmin askeleet

Tiivistettynä simplex-algoritmi toimii alla esitetyllä tavalla.

##### Algoritmi 2. Simplex-algoritmi

**Askel 0.** (*Alustus*) Muodostetaan kantaratkaisu  $x^0$  valitsemalla jokin sallittu lähtökanta ja asetetaan kierroslaskuriin arvo  $t \leftarrow 0$ .

**Askel 1.** (*Simplex-suunnat*) Parannetaan jokaista ei-kantamuuttujaa muodostamalla simplex-suunta  $\Delta \mathbf{x}_j = (x_j, 0, 0, \dots, \Delta x_{n-1}, \Delta x_n)$ , jossa jokainen ei-kantamuuttuja  $x_j$  on vuorollaan 1 ja muut ei-kantamuuttujat ovat 0. Kantamuuttujat merkittyinä  $\Delta x_n$  selvitetään lineaarisista yhtälöryhmistä. Lopuksi jokaiselle simplex-suunnalle lasketaan sen redusoitu kustannus kaavalla  $\bar{c}_j = \mathbf{c}^T \Delta \mathbf{x}_j$ .

**Askel 2.** (*Optimaalisuus*) Maksimointitehtävän parantava suunta on jokin  $c_j > 0$ , ja minimointitehtävässä se on  $c_j < 0$ . Jos parantavaa suuntaa ei ole, sen hetkinen kantaratkaisu  $\mathbf{x}^t$  on optimaalinen ja kierto lopetetaan. Muuten valitaan jokin parantava suunta  $\Delta \mathbf{x}_j$  ja vaihdetaan suuntaa vastaan muuttujan indeksi  $x_j \leftarrow x_p$ .

**Askel 3.** (*Askelpituus*) Jos kaikki kantamuuttujia vastaavat suunnan  $\Delta \mathbf{x}_j$  komponentit ovat negatiivisia, tehtävällä ei ole äärellistä ratkaisua ja kierto lopetetaan. Muuten valitaan kannasta poistuva muuttuja kaavalla

$$\frac{x_r^t}{-\Delta x_r^{t+1}} = \min \left\{ \frac{x_j^t}{-\Delta x_j^{t+1}} \mid \Delta x_j^{t+1} < 0, x_j^t \text{ kantamuuttuja} \right\}.$$

Asetetaan valittu arvo askelpituudeksi,

$$\lambda = \frac{x_r^t}{-\Delta x_r^{t+1}}.$$

**Askel 4.** (*Uusi kärki ja kanta*) Uusi kärkipiste lasketaan kaavalla

$$\mathbf{x}^{t+1} = \mathbf{x}^t + \lambda \Delta \mathbf{x}^{t+1}.$$

Korvataan myös kannassa oleva muuttuja  $x_r$  muuttujalla  $x_p$ . Lopuksi kasvatetaan kierroslaskuria  $t \leftarrow t + 1$  ja palataan askeleeseen 1.

### 3.2 Ongelman ratkaisu simplex-algoritmillä

Käydään seuraavaksi läpi algoritmin toiminta esimerkin avulla.

**Esimerkki 2.** Tarkastellaan lineaarista optimointitehtävää (lähde: [3])

$$\begin{aligned} \max \quad & 5x_1 + 4x_2 + 3x_3 \\ \text{s. t.} \quad & 2x_1 + 3x_2 + x_3 \leq 5 \\ & 4x_1 + x_2 + 2x_3 \leq 11 \\ & 3x_1 + 4x_2 + 2x_3 \leq 8 \\ & x_1, x_2, x_3 \geq 0. \end{aligned}$$

Muutetaan tehtävä standardimuotoon ottamalla käyttöön lisämuuttujat  $x_4, x_5$  ja  $x_6$ , joiden avulla tehtävä saadaan muotoon

$$\begin{aligned} \max \quad & 5x_1 + 4x_2 + 3x_3 \\ \text{s. t.} \quad & 2x_1 + 3x_2 + x_3 + x_4 = 5 \\ & 4x_1 + x_2 + 2x_3 + x_5 = 11 \\ & 3x_1 + 4x_2 + 2x_3 + x_6 = 8 \\ & x_1, x_2, x_3, x_4, x_5, x_6 \geq 0. \end{aligned}$$

Nyt tehtävää voidaan alkaa iteroimaan simplex-algoritmillä. Ensin muodostetaan kantaratkaisu, joka nyt saadaan asettamalla  $x_1, x_2, x_3 = 0$ . Sijoitetaan nämä arvot yhtälöön  $\mathbf{Ax} = \mathbf{b}$ , josta saadaan ratkaistua arvot  $x_4 = 5$ ,  $x_5 = 11$  ja  $x_6 = 8$ . Sallittu kantaratkaisu on siis  $x^0 = \{0, 0, 0, 5, 11, 8\}^T$ , ja kohdefunktion arvo kantaratkaisulla on  $x^0 c^T = \{0, 0, 0, 5, 11, 8\}^T \{5, 4, 3, 0, 0, 0\}^T = 0$ . Kantamuuttujat ovat  $\mathbf{B} = \{4, 5, 6\}$  ja ei-kantamuuttujat ovat  $\mathbf{N} = \{1, 2, 3\}$ .

Seuraavaksi etsitään simplex-suunnat muodostamalla jokaiselle ei-kantamuuttujalle yhtälö  $\Delta x_1 = (1, 0, 0, \Delta x_4, \Delta x_5, \Delta x_6)$ . Tuntemattomat muuttujat selvitetään yhtälöstä  $\mathbf{A} \Delta x_1 = 0$ , joka on auki kirjoitettuna

$$\mathbf{A} \Delta x_1 = \begin{cases} 2 \cdot 1 + 3 \cdot 0 + 1 \cdot 0 + 1 \cdot \Delta x_4 + 0 \cdot \Delta x_5 + 0 \cdot \Delta x_6 = 0 \\ 4 \cdot 1 + 1 \cdot 0 + 2 \cdot 0 + 0 \cdot \Delta x_4 + 1 \cdot \Delta x_5 + 0 \cdot \Delta x_6 = 0 \\ 3 \cdot 1 + 4 \cdot 0 + 2 \cdot 0 + 0 \cdot \Delta x_4 + 0 \cdot \Delta x_5 + 1 \cdot \Delta x_6 = 0. \end{cases}$$

Yhtälöstä saadaan arvot

$$\begin{cases} \Delta x_4 = -2 \\ \Delta x_5 = -4 \\ \Delta x_6 = -3. \end{cases}$$

Simplex-suunnaksi saadaan  $\Delta x_1 = (1, 0, 0, -2, -4, -3)$ . Simplex-suunnille  $\Delta x_2 = (0, 1, 0, \Delta x_4, \Delta x_5, \Delta x_6)$  ja  $\Delta x_3 = (0, 0, 1, \Delta x_4, \Delta x_5, \Delta x_6)$  suoritetaan sama toimenpide. Niille saadaan arvoksi  $\Delta x_2 = (0, 1, 0, -3, -1, -4)$  ja  $\Delta x_3 = (0, 0, 1, -1, -2, -2)$ . Lasketaan simplex-suuntien avulla redusoidut kustannukset eli, kuinka kannattava muuttujaa vastaava suunta on. Ne lasketaan kaavalla  $\bar{c}_j = c^T \Delta x_j$ , eli tässä tapauksessa

$$\begin{aligned} \bar{c}_1 &= c^T \Delta x_1 = (5, 4, 3, 0, 0, 0)^T \cdot (1, 0, 0, -2, -4, -3) = 5 \\ \bar{c}_2 &= c^T \Delta x_2 = (5, 4, 3, 0, 0, 0)^T \cdot (0, 1, 0, -3, -1, -4) = 4 \\ \bar{c}_3 &= c^T \Delta x_3 = (5, 4, 3, 0, 0, 0)^T \cdot (0, 0, 1, -1, -2, -2) = 3. \end{aligned}$$

Kaikki redusoidut kustannukset ovat positiivisia, eli jokainen suunta on parantava. Valitaan suunnaksi  $\Delta x_1$ . Lasketaan kannasta poistuva muuttuja ja askelpituus. Askelpituudeksi saadaan

$$\lambda = \min \left\{ \frac{5}{-(-2)}, \frac{11}{-(-4)}, \frac{8}{-(-3)} \right\} = \frac{5}{2}.$$

Valittu arvo vastaa muuttujaa  $x_4$ , se poistuu kannasta ja  $x_1$  tulee kantaan sen tilalle. Uusi kärkipiste tehtävälle on

$$\begin{aligned} x^1 &= x^0 + \lambda \Delta x_1 \\ x^1 &= (0, 0, 0, 5, 11, 8) + \frac{5}{2} \cdot (1, 0, 0, -2, -4, -3) \\ x^1 &= \left( \frac{5}{2}, 0, 0, 0, 1, \frac{1}{2} \right). \end{aligned}$$

Kohdefunktion arvo uudessa kärkipisteessä on

$$c^T x^1 = (5, 4, 3, 0, 0, 0) \cdot \left( \frac{5}{2}, 0, 0, 0, 1, \frac{1}{2} \right) = \frac{25}{2}.$$

Uusi iteraatiokierros: Kasvatetaan kierroslaskuria  $t \leftarrow 2$ . Kantamuuttujat ovat  $\mathbf{B} = \{1, 5, 6\}$  ja ei-kantamuuttujat  $\mathbf{N} = \{2, 3, 4\}$ . Uudet simplex-suunnat ovat:

$$\begin{aligned} \Delta x_2 &= (\Delta x_1, 1, 0, 0, \Delta x_5, \Delta x_6) \\ \Delta x_3 &= (\Delta x_1, 0, 1, 0, \Delta x_5, \Delta x_6) \\ \Delta x_4 &= (\Delta x_1, 0, 0, 1, \Delta x_5, \Delta x_6). \end{aligned}$$

Niille saadaan arvoiksi:

$$\begin{aligned} \Delta x_2 &= \left( -\frac{3}{2}, 1, 0, 0, 5, \frac{1}{2} \right) \\ \Delta x_3 &= \left( -\frac{1}{2}, 0, 1, 0, 0, -\frac{1}{2} \right) \\ \Delta x_4 &= \left( -\frac{1}{2}, 0, 0, 1, 2, \frac{3}{2} \right). \end{aligned}$$

Joista saadaan laskettua seuraavat redusoidut kustannukset:

$$\begin{aligned}\bar{c}_2 &= c^T \Delta x_2 = (5, 4, 3, 0, 0, 0)^T \cdot \left(-\frac{3}{2}, 1, 0, 0, 5, \frac{1}{2}\right) = -\frac{7}{2} \\ \bar{c}_3 &= c^T \Delta x_3 = (5, 4, 3, 0, 0, 0)^T \cdot \left(-\frac{1}{2}, 0, 1, 0, 0, -\frac{1}{2}\right) = \frac{1}{2} \\ \bar{c}_4 &= c^T \Delta x_4 = (5, 4, 3, 0, 0, 0)^T \cdot \left(-\frac{1}{2}, 0, 0, 1, 2, \frac{3}{2}\right) = -\frac{5}{2}.\end{aligned}$$

Ainoa parantava suunta on  $\Delta x_3$ . Askelpituudeksi tulee

$$\lambda = \min \left\{ \frac{\frac{5}{2}}{-(-\frac{1}{2})}, \frac{\frac{1}{2}}{-(-\frac{1}{2})} \right\} = \min\{5, 1\} = 1.$$

Kantamuuttuja  $x_6$  vastaa valittua arvoa, se poistuu kannasta,  $x_3$  tulee kantaan. Uusi kärkipiste on nyt

$$\begin{aligned}x^2 &= x^1 + \lambda \Delta x_3 = \left(\frac{5}{2}, 0, 0, 0, 1, \frac{1}{2}\right) + 1 \left(-\frac{1}{2}, 0, 1, 0, 0, -\frac{1}{2}\right) \\ x^2 &= (2, 0, 1, 0, 1, 0).\end{aligned}$$

Kohdefunktion arvo uudessa kärkipisteessä on

$$c^T x^2 = (5, 4, 3, 0, 0, 0)^T (2, 0, 1, 0, 1, 0) = 13.$$

Kantamuuttujat ovat  $\mathbf{B} = \{1, 3, 5\}$  ja ei-kantamuuttujat ovat  $\mathbf{N} = \{2, 4, 6\}$ . Simplex-suunnat ovat:

$$\begin{aligned}\Delta x_2 &= (\Delta x_1, 1, \Delta x_3, 0, \Delta x_5, 0) \\ \Delta x_4 &= (\Delta x_1, 0, \Delta x_3, 1, \Delta x_5, 0) \\ \Delta x_6 &= (\Delta x_1, 0, \Delta x_3, 0, \Delta x_5, 1).\end{aligned}$$

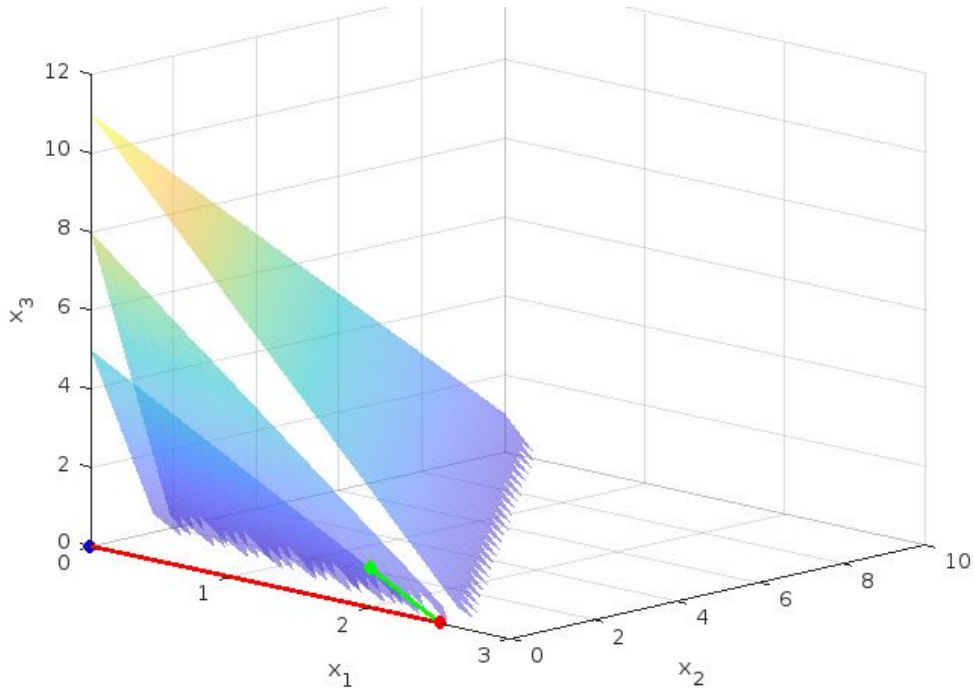
Niiden arvoiksi saadaan:

$$\begin{aligned}\Delta x_2 &= (-2, 1, 1, 0, 5, 0) \\ \Delta x_4 &= (-2, 0, 3, 1, 2, 0) \\ \Delta x_6 &= (1, 0, -2, 0, 0, 1).\end{aligned}$$

Redusoidut kustannukset ovat:

$$\begin{aligned}\bar{c}_2 &= c^T \Delta x_2 = (5, 4, 3, 0, 0, 0)^T (-2, 1, 1, 0, 5, 0) = -3 \\ \bar{c}_4 &= c^T \Delta x_4 = (5, 4, 3, 0, 0, 0)^T (-2, 0, 3, 1, 2, 0) = -1 \\ \bar{c}_6 &= c^T \Delta x_6 = (5, 4, 3, 0, 0, 0)^T (1, 0, -2, 0, 0, 1) = -1.\end{aligned}$$

Kaikki redusoidut kustannukset ovat negatiivisia, jolloin parantavaa suuntaa ei ole, eli kantaratkaisu  $x^2$  on globaali optimi, jossa kohdefunktion arvo on 13. Kuvasta 1 voidaan hahmottaa, kuinka simplex-algoritmi etenee. Rajoitteet muodostavat tasoja, joiden reunalta voidaan löytää optimi.



Kuva 1: Simplex-algoritmin eteneminen

## 4 Parannettu simplex-menetelmä

Parannettu simplex-menetelmä on matemaattisesti sama kuin perinteinen simplex-algoritmi, toteutus on vain hiukan erilainen. Perinteinen simplex-algoritmi vaatii lineaarisen yhtälön ratkaisua jokaisen kantaratkaisun ja simplex-suunnan määrittämiseksi. Parannetussa simplex-menetelmässä sen sijaan käytetään kantamatriisia  $\mathbf{B}$ , joka on muodostettu alkuperäisen matriisin kantamuuttujia vastaavista sarakkeista. Kantamatriisi ei paljon muutu iteraation aikana, joten uutta kantamatriisin kääntematriisia ei tarvitse aina muodostaa. Sen sijaan jokaisen iteraation lopuksi kantamatriisia päivitetään päivitysmatriisilla. Päivitysmatriisi on  $n \cdot n$  matriisi, jossa  $n$  on yhtä suuri kuin kantamuuttujien lukumäärä. Siinä jaetaan valitun simplex-suunnan jokainen komponentti kannasta poistuvan  $x_r$  muuttujan simplex-suunnan arvolla. Jakolaskut laitetaan valitun simplex-suunnan indeksin sarakkeeseen. Esimerkiksi, jos valittu suunta on  $\Delta x^1$ , jakolaskut laitetaan ensimmäiseen sarakkeeseen. Kannasta poistuva muuttuja ei myöskään jaa itseään, vaan sen kohdalle laitetaan arvo  $-1$ . Päivitysmatriisi voi olla esimerkiksi seuraavanlainen:

$$\mathbf{E} = \begin{pmatrix} 1 & 0 & \cdots & 0 & -\frac{\Delta x_1}{\Delta x_r} & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & -\frac{\Delta x_2}{\Delta x_r} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 & -\frac{\Delta x_{r-1}}{\Delta x_r} & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & -\frac{-1}{\Delta x_r} & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & -\frac{\Delta x_{r+1}}{\Delta x_r} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & -\frac{\Delta x_j}{\Delta x_r} & 0 & \cdots & 1 \end{pmatrix}.$$

## 4.1 Parannetun simplex-menetelmän algoritmi

Tiivistettynä parannettu simplex-menetelmä toimii alla esitetyllä tavalla.

### Algoritmi 3. Parannettu simplex-menetelmä

**Askel 0.** (*Alustus*) Valitaan jokin sallittu lähtökanta ja tehdään sitä vastaavista matriisin  $\mathbf{A}$  pystyriveistä kantamatriisiin käänteismatriisi  $\mathbf{B}^{-1}$ . Sitä käyttämällä ratkaistaan yhtälön  $\mathbf{B}\mathbf{x}^B = \mathbf{b}$  toteuttavat kantamuuttujien  $\mathbf{x}^0$  arvot. Ei-kantamuuttujien arvoiksi asetetaan nolla,  $x_j^0 \leftarrow 0$  ja asetetaan kierroslaskurin arvoksi myös nolla,  $t \leftarrow 0$ .

**Askel 1.** (*Pricing*) Lasketaan arvo  $\mathbf{v}^T = (\mathbf{c}^B)^T \mathbf{B}^{-1}$ , jonka avulla lasketaan jokaisen ei-kantamuuttujan redusoitu kustannus  $\bar{c}_j = c_j - \mathbf{v}^T \mathbf{a}^j$ .

**Askel 2.** (*Optimaalisuus*) Maksimointitehtävän parantava suunta on jokin  $\bar{c}_j > 0$ , ja minimointitehtävässä se on  $\bar{c}_j < 0$ . Jos parantavaa suuntaa ei ole, sen hetkinen kantaratkaisu  $\mathbf{x}^t$  on optimaalinen ratkaisu. Muuten valitaan uudeksi kantamuuttujaksi jokin  $x_j$ , jolla on parantava suunta  $\bar{c}_j$ . Lopuksi vaihdetaan valitun kantamuuttujan indeksi,  $x_j \leftarrow x_p$ .

**Askel 3.** (*Simplex-suunta*) Määrätään kantaan otettavaa muuttujaa  $x_p$  vastaavan simplex-suunnan  $\Delta\mathbf{x}^{t+1}$  kantaa vastaavat komponentit  $\Delta\mathbf{x} = -\mathbf{B}^{-1}\mathbf{a}^p$ . Muuttujat, jotka eivät kuulu kantaan ovat nollia, paitsi indeksiä  $p$  vastaava komponentti, joka on yksi.

**Askel 4.** (*Askelpituus*) Jos saadun simplex-suunnan  $\Delta\mathbf{x}^{t+1}$  kaikki komponentit ovat ei-negatiivisia, lopetetaan, sillä kohdefunktion arvo ei ole rajoitettu. Muuten valitaan kannasta poistuva muuttuja  $x_r$  ehdosta

$$\frac{x_r^t}{-\Delta x_r^{t+1}} = \min_{j \in B} \left\{ \frac{x_r^t}{-\Delta x_j^{t+1}} \mid \Delta x_j^{t+1} < 0 \right\}.$$

Lopuksi asetetaan saatu arvo askelpituudeksi

$$\lambda = \frac{x_r^t}{-\Delta x_r^{t+1}}.$$

**Askel 5.** (*Uusi kärki ja kanta*) Määrätään uusi kärkipiste  $\mathbf{x}^{t+1} \leftarrow \mathbf{x}^t + \lambda \Delta\mathbf{x}^{t+1}$  ja asetetaan  $x_p$  kantaan muuttujan  $x_r$  tilalle. Tehdään päivitysmatriisi  $\mathbf{E}$  ja päivitetään kantamatriisiin käänteismatriisia sen avulla,  $\mathbf{E}\mathbf{B}^{-1}$ . Lopuksi kasvatetaan kierroslaskuria  $t \leftarrow t + 1$  ja palataan askeleeseen yksi.

## 4.2 Ongelman ratkaisu parannetulla simplex-menetelmällä

Ratkaistaan sama lineaarinen optimointiongelma, kuin esimerkissä (2) parannetun simplex-algoritmin avulla.

**Esimerkki 3.** Standardimuotoinen tehtävä on muotoa

$$\begin{aligned} \max \quad & 5x_1 + 4x_2 + 3x_3 \\ \text{s. t.} \quad & 2x_1 + 3x_2 + x_3 + x_4 = 5 \\ & 4x_1 + x_2 + 2x_3 + x_5 = 11 \\ & 3x_1 + 4x_2 + 2x_3 + x_6 = 8 \\ & x_1, x_2, x_3, x_4, x_5, x_6 \geq 0. \end{aligned}$$

Lähtökanta saadaan samalla tavalla kuin aiemmin, eli asetetaan  $x_1, x_2, x_3 = 0$ . Kantamuuttujat ovat  $\mathbf{B} = \{x_4, x_5, x_6\}$ . Alussa on matriisi  $\mathbf{A}$  ja kantamuuttujia vastaava matriisi  $\mathbf{B}$ , jotka ovat arvoltaan

$$\mathbf{A} = \begin{pmatrix} 2 & 3 & 1 & 1 & 0 & 0 \\ 4 & 1 & 2 & 0 & 1 & 0 \\ 3 & 4 & 2 & 0 & 0 & 1 \end{pmatrix} \quad \text{ja} \quad \mathbf{B} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \mathbf{B}^{-1}.$$

Ratkaistaan kantamuuttujien arvot yhtälöstä  $\mathbf{B}\mathbf{x}^{\mathbf{B}} = \mathbf{b}$ ,

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} 5 \\ 11 \\ 8 \end{pmatrix}.$$

Arvoiksi saadaan  $x_4 = 5, x_5 = 11, x_6 = 8$ , eli lähtökanta on  $x^0 = (0, 0, 0, 5, 11, 8)$ . Lasketaan arvo  $v^T$ :

$$v^T = (\mathbf{c}^{\mathbf{B}})^T \mathbf{B}^{-1} = (0, 0, 0) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = (0, 0, 0).$$

Tämän avulla lasketaan jokaiselle ei-kantamuuttujalle redusoitu kustannus:

$$\begin{aligned} \bar{c}_1 &= c_1 - v^T a^1 = 5 - (0, 0, 0) \begin{pmatrix} 2 \\ 4 \\ 3 \end{pmatrix} = 5 \\ \bar{c}_2 &= c_2 - v^T a^2 = 4 - (0, 0, 0) \begin{pmatrix} 3 \\ 1 \\ 4 \end{pmatrix} = 4 \\ \bar{c}_3 &= c_3 - v^T a^3 = 3 - (0, 0, 0) \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix} = 3. \end{aligned}$$

Jokainen suunta on parantava, joten valitaan eniten parantava suunta eli  $x_1$ . Lasketaan simplex-suunnan kantaa vastaavat komponentit

$$\Delta \mathbf{x} = -\mathbf{B}^{-1} \mathbf{a}^1 = - \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 4 \\ 3 \end{pmatrix} = \begin{pmatrix} -2 \\ -4 \\ -3 \end{pmatrix}.$$

Simplex-suunta on siis  $\Delta x^1 = (1, 0, 0, -2, -4, -3)$ . Valitaan kannasta poistuva muuttuja ehdosta

$$\min \left\{ \frac{x_4^0}{-\Delta x_4^1}, \frac{x_5^0}{-\Delta x_5^1}, \frac{x_6^0}{-\Delta x_6^1} \right\} = \min \left\{ \frac{5}{-(-2)}, \frac{11}{-(-4)}, \frac{8}{-(-3)} \right\} = \frac{5}{2}.$$

Valittu arvo vastaa muuttujaa  $x_4$ , se poistuu kannasta ja  $x_1$  menee tilalle. Uusi kanta on  $\mathbf{B} = \{x_1, x_5, x_6\}$ . Asetetaan askelpituudeksi valittu arvo,  $\lambda = \frac{5}{2}$ . Uudeksi kantaratkaisuksi saadaan

$$\begin{aligned} x^1 &= x^0 + \lambda \Delta x^1 = (0, 0, 0, 5, 11, 8) + \frac{5}{2}(1, 0, 0, -2, -4, -3) \\ x^1 &= \left(\frac{5}{2}, 0, 0, 0, 1, \frac{1}{2}\right). \end{aligned}$$

Kohdefunktion arvo uudella kantaratkaisulla on  $c^T x^1 = 12.5$ . Lopuksi tehdään päivitysmatriisi ja päivitetään kantamatriisia  $\mathbf{B}$  sillä:

$$\begin{aligned} E &= \begin{pmatrix} -\frac{1}{\Delta x_4^1} & 0 & 0 \\ -\frac{\Delta x_5^1}{\Delta x_4^1} & 1 & 0 \\ -\frac{\Delta x_6^1}{\Delta x_4^1} & 0 & 1 \end{pmatrix} = \begin{pmatrix} -\frac{1}{-2} & 0 & 0 \\ -\frac{-4}{-2} & 1 & 0 \\ -\frac{-3}{-2} & 0 & 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ -2 & 1 & 0 \\ -\frac{3}{2} & 0 & 1 \end{pmatrix} \\ EB^{-1} &= \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ -2 & 1 & 0 \\ -\frac{3}{2} & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ -2 & 1 & 0 \\ -\frac{3}{2} & 0 & 1 \end{pmatrix}. \end{aligned}$$

Uusi iteraatiokierros: Kasvatetaan kierrosalaskuria,  $t \leftarrow 2$ . Lasketaan uusi  $v^T$ :

$$v^T = (5, 0, 0) \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ -2 & 1 & 0 \\ -\frac{3}{2} & 0 & 1 \end{pmatrix} = \left(\frac{5}{2}, 0, 0\right).$$

Lasketaan redusoidut kustannukset:

$$\begin{aligned} \bar{c}_2 &= c_2 - v^T a^2 = 4 - \left(\frac{5}{2}, 0, 0\right) \begin{pmatrix} 3 \\ 1 \\ 4 \end{pmatrix} = -\frac{7}{2} \\ \bar{c}_3 &= c_3 - v^T a^3 = 3 - \left(\frac{5}{2}, 0, 0\right) \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix} = \frac{1}{2} \\ \bar{c}_6 &= c_6 - v^T a^6 = 0 - \left(\frac{5}{2}, 0, 0\right) \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = 0. \end{aligned}$$

$\bar{c}_3$  on ainoa positiivinen, joten valitaan  $x_3$  suunnaksi. Lasketaan kantaa vastaavat komponentit

$$\Delta x_3 = -\mathbf{B}^{-1} a^3 = - \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ -2 & 1 & 0 \\ -\frac{3}{2} & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix} = \begin{pmatrix} -\frac{1}{2} \\ 0 \\ -\frac{1}{2} \end{pmatrix}.$$

Simplex-suunta on siis  $\Delta x^2 = (-\frac{1}{2}, 0, 1, 0, 0, -\frac{1}{2})$ . Valitaan kannasta poistuva muuttuja

$$\min \left\{ \frac{x_1^1}{-\Delta x_1^2}, \frac{x_6^1}{-\Delta x_6^2} \right\} = \min \left\{ \frac{\frac{5}{2}}{-(-\frac{1}{2})}, \frac{\frac{1}{2}}{-(-\frac{1}{2})} \right\} = 1.$$

Valittu arvo vastaa muuttujaa  $x_6$ , joka poistuu kannasta ja  $x_3$  menee sen tilalle. Uusi kanta on  $\mathbf{B} = \{x_1, x_5, x_3\}$ . Asetetaan valittu arvo askelpituudeksi,  $\lambda = 1$ , ja lasketaan uusi kantaratkaisu

$$\begin{aligned} x^2 &= x^1 + \lambda \Delta x^2 = \left(\frac{5}{2}, 0, 0, 0, 1, \frac{1}{2}\right) + 1\left(-\frac{1}{2}, 0, 1, 0, 0, -\frac{1}{2}\right) \\ x^2 &= (2, 0, 1, 0, 1, 0). \end{aligned}$$

Kohdefunktion arvo uudella kantaratkaisulla on  $c^T x^2 = 13$ . Esimerkistä (2) tiedetään, että tämä on globaali optimi. Kierretään kuitenkin vielä algoritmia, kunnes algoritmin mukaan voidaan lopettaa kierto. Tehdään päivitysmatriisi ja päivitetään kantamatriisia

$$\begin{aligned} E &= \begin{pmatrix} 1 & 0 & -\frac{-\frac{1}{2}}{-\frac{1}{2}} \\ 0 & 1 & -\frac{0}{-\frac{1}{2}} \\ 0 & 0 & -\frac{\frac{1}{2}}{-\frac{1}{2}} \end{pmatrix} = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{pmatrix} \\ EB^{-1} &= \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ -2 & 1 & 0 \\ -\frac{3}{2} & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & -1 \\ -2 & 1 & 0 \\ -3 & 0 & 2 \end{pmatrix}. \end{aligned}$$

Uusi iteraatiokierto: Kasvatetaan kierrosalaskuria,  $t \leftarrow 3$  ja lasketaan uusi  $v^T$ :

$$v^T = (5, 0, 3) \begin{pmatrix} 2 & 0 & -1 \\ -2 & 1 & 0 \\ -3 & 0 & 2 \end{pmatrix} = (1, 0, 1).$$

Lasketaan ei-kantamuuttujien redusoidut kustannukset:

$$\begin{aligned} \bar{c}_2 &= c_2 - v^T a^2 = 4 - (1, 0, 1) \begin{pmatrix} 3 \\ 1 \\ 4 \end{pmatrix} = -3 \\ \bar{c}_4 &= c_4 - v^T a^4 = 0 - (1, 0, 1) \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = -1 \\ \bar{c}_6 &= c_6 - v^T a^6 = 0 - (1, 0, 1) \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = -1. \end{aligned}$$

Yksikään redusoitu kustannus ei ole positiivinen, joten lopetetaan kierto. Kantaratkaisu  $x^2$  on optimaalinen ratkaisu, ja kohdefunktion arvoksi saadaan  $c^T x^2 = 13$ . Matemaattisesti parannettu simplex algoritmi ja simplex-algoritmi ovat samat, eli parannettu simplex-algoritmi etenee samalla tavalla kuin simplex-algoritmi kuvassa 1.

## 5 Yhteenveto

Yllä olevista esimerkeistä nähdään, että parannetulla simplex-menetelmällä ja simplex-algoritmillä saadaan aina sama lopputulos lineaarisille optimointitehtäville. Simplex-algoritmi on näistä kahdesta helpompi ymmärtää ja käyttää, mutta parannettu simplex-menetelmä on matemaattisesti tehokkaampi, varsinkin kun yhtälörajoitusten määrä kasvaa. Jokaista yhtälörajoitusta kohden simplex-algoritmillä joudutaan ratkaisemaan lineaarinen yhtälöryhmä ja muuttuja yhtälöryhmästä. Parannetun simplex-menetelmän laskuaika ei kasva niin rajusti. Kantamuuttujan lukumäärän kasvaessa kantamatriisi kasvaa yhdellä ulottuvuudella, jonka lisäksi joudutaan laskemaan yksi uusi redusoitu kustannus ja valitun suunnan kantamuuttujan arvo.

Pahimmassa tapauksessa simplex-algoritmin laskuaika on  $O(n^m)$ , jossa  $n$  on muuttujien lukumäärä ja  $m$  on epäyhtälörajoitteiden lukumäärä [7]. Keskimäärin sen laskuaika on  $O((n+m)n)$  [6]. Näistä molemmat ovat teoreettisia laskuaikoja, joten todellisuudessa laskuaika on luultavasti pienempi. Parannetun simplex-menetelmän laskuaika on noin  $O(m^3)$ . Reaalimaailman tilanteissa käytetään erilaisia ohjelmistoja optimointiongelmiin ratkaisuun, kuten CPLEX tai MATLAB. Niitä käytetään, sillä reaalimaailman optimointiongelmat ovat huomattavasti laajempia kuin mitä yleensä käytetään esimerkkeinä.

Simplex-algoritmi on yksi lineaarisen optimoinnin kulmakivistä ja se on edelleen laajasti käytössä sovellettavuutensa ansiosta. Parannettu simplex-menetelmä puolestaan tuo tehokkuutta erityisesti laajoihin ongelmiin vähentämällä ylimääräisiä laskuja.

## Viitteet

- [1] Marko M. Mäkelä: Matemaattinen Optimointi I. Luentomoniste, Turun Yliopisto, 2020
- [2] Fiasco A. & McCormick B.: Advancements in Computational Techniques for Optimization. Operations Research 1965
- [3] Václav Chvátal: Linear Programming, 1983
- [4] Felix Fischer: Optimization. Luentomoniste, osa 8, Cambridgen Yliopisto 2015
- [5] George B. Dantzig: Origins of the Simplex Method, 1990
- [6] Volkan Erol & Mert side: Applying Quantum Optimization Algorithms for Linear Programming. 2017
- [7] A. H. G. Rinnoy Kan, J. Telgen: The complexity of linear programming. Statistica Neerlandica, nr. 2. 1981