

# **Kartonkituotannon materiaalihävikin minimointi stokastisella algoritmilla**

Konsta Palmanto

**Perustieteiden korkeakoulu**

Kandidaatintyö  
Espoo 30.8.2024

**Vastuupettaja**

Prof. Ahti Salo

**Työn ohjaaja**

DI. Leevi Olander

Copyright © 2024 Konsta Palmanto

The document can be stored and made available to the public on the open internet pages of Aalto University.  
All other rights are reserved.

---

**Tekijä** Konsta Palmanto

**Työn nimi** Kartonkituotannon materiaalihävikin minimointi stokastisella algoritmilla

**Koulutusohjelma** Teknistieteellinen kandidaattiohjelma

**Pääaine** Matematiikka ja systeemitieteet **Pääaineen koodi** SCI3029

**Vastuopettaja** Prof. Ahti Salo

**Työn ohjaaja** DI. Leevi Olander

**Päivämäärä** 30.8.2024

**Sivumäärä** 23

**Kieli** Suomi

**Tiivistelmä**

Tässä työssä esitellään optimointimenetelmä kartonkituotannon materiaalihävikin minimoimiseksi. Optimoinnissa on kaksi vaihetta, joista ensimmäisessä arkkitilaukset leikataan varastorullista. Toisessa vaiheessa varastorullat leikataan kartonkikoneen tekemistä jumborullista. Ensimmäinen vaihe on kaksiulotteinen trimmitysongelma, joka ratkaistaan heuristisella menetelmällä. Toinen puolestaan on yksiulotteinen trimmitysongelma, joka ratkaistaan *first fit decreasing* -algoritmilla. Nämä kaksi vaihetta yhdistävän pääongelman optimointitehtävä ratkaistaan stokastisella paikallisen haun algoritmilla, jonka päätösmuuttujina ovat varastorullien leveydet. Optimointimenetelmää testataan todenmukaisella datalla. Tulokset osoittavat, että hävikki vähenee varastorullien määrän kasvaessa. Optimoitu tulos on selvästi parempi, kuin käypien ratkaisujen keskimääräinen hävikki.

---

**Avainsanat** trimmitysongelma, stokastinen algoritmi

---

---

**Author** Konsta Palmanto

---

**Title** Minimizing Material Waste in Cardboard Production with a Stochastic Algorithm

---

**Degree programme** Bachelor's Programme in Science and Technology

---

**Major** Mathematics and Systems Sciences

**Code of major** SCI3029

---

**Teacher in charge** Prof. Ahti Salo

---

**Advisor** M.Sc. (Tech.) Leevi Olander

---

**Date** 30.8.2024

**Number of pages** 23

**Language** Finnish

---

**Abstract**

This thesis presents an optimization method for minimizing material waste in cardboard production. The optimization has two stages, such that sheet orders are cut from stock-sized rolls in the first stage and the stock-sized rolls are cut from jumbo rolls made by the cardboard machine in the second stage. The first stage is a two-dimensional cutting stock problem, which is solved using a heuristic method. The second one is a one-dimensional cutting stock problem, solved using the *first fit decreasing*-algorithm. This main optimization problem, which combines these two stages, is solved using a stochastic local search algorithm, with the decision variables being the widths of the stock-sized rolls. The optimization method is tested with data that resembles reality. The results show that the amount of waste decreases as the number of stock-sized rolls increases. The optimized result produces a clear improvement compared to the average waste in the set of feasible solutions.

---

**Keywords** cutting stock problem, stochastic algorithm

---

# Sisällys

<b>Tiivistelmä</b>	<b>3</b>
<b>Tiivistelmä (englanniksi)</b>	<b>4</b>
<b>Sisällys</b>	<b>5</b>
<b>1 Johdanto</b>	<b>6</b>
<b>2 Aikaisempi tutkimus</b>	<b>7</b>
2.1 Trimmitysongelman typologia . . . . .	7
2.2 Trimmitysongelman ratkaisumenetelmät . . . . .	8
<b>3 Tutkimusmenetelmät</b>	<b>8</b>
3.1 Pääongelma . . . . .	8
3.2 Pääongelman ratkaisumenetelmä . . . . .	12
3.3 Paikallisen haun algoritmi . . . . .	13
3.4 Arkkileikkurioptimointi . . . . .	15
3.5 Kartonkikoneoptimointi . . . . .	18
<b>4 Tulokset</b>	<b>18</b>
<b>5 Yhteenveto</b>	<b>21</b>

# 1 Johdanto

Kartonkituotannossa materiaalitehokkuus parantaa taloudellista katetta, vähentää energiankulutusta ja nopeuttaa prosesseja. Useimmissa tutkimuksissa materiaalitehokkuuden optimointi liittyy vain yhteen tuotantovaiheeseen. Tässä työssä esitellään optimointimenetelmä, jonka tavoitteena on minimoida koko kartonkiarkkituotannon materiaalihävikki.

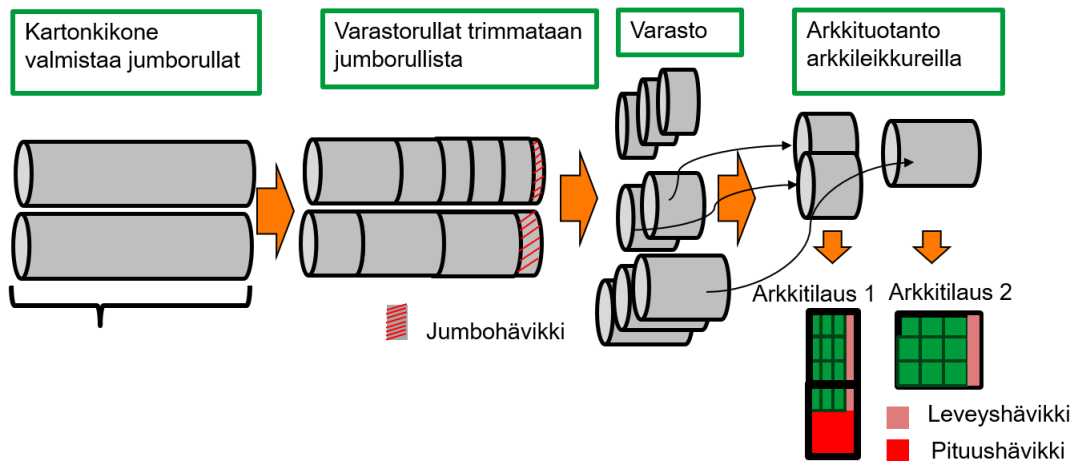
Kartonkituotannossa on kaksi vaihetta. Ensimmäisessä kartonkikoneen valmistamista rullista eli jumborullista leikataan pienempiä varastorullia. Toisessa arkkitilaukset leikataan varastorullista arkkileikkureilla. Materiaalihävikkiä syntyy kummassakin vaiheessa. Tuotanto on havainnollistettu kuvassa 1.

Optimoinnin pääongelman tavoitteena on valita optimaaliset varastorullaleveydet. Pääongelma ratkaistaan stokastisella paikallisen haun algoritmilla. Optimointi tehdään todellisuutta jäljittelevälle arkkitilausdatalle, jossa on annettu arkkitilauksen kokonaispinta-alat, suhteelliset toleranssit sekä arkkikoot. Työssä tarkastellaan, kuinka varastorullaleveyksien optimointi vaikuttaa tuotantohävikkiin verrattuna satunnaisesti valituilla varastorullaleveyksillä toteutettuun tuotantoon.

Ongelma jaetaan tuotannon vaiheiden mukaisesti kahteen osaongelmaan. Optimoinnin rakenteen vuoksi osaongelmat ratkaistaan käänteisessä järjestyksessä verrattuna todellisen tuotannon vaiheisiin. Ensimmäistä osaongelmaa kutsutaan arkkileikkurioptimoinniksi ja toista kartonkikoneoptimoinniksi. Osaongelmat liittyvät laskennallisesti haastavaan trimmitysongelmaan. Trimmitysongelma on yläkäsite optimointiongelmiin, joissa varastomateriaalista leikataan pienempiä yksiköitä. Optimoinnin tavoitteena on prosessissa syntyvän hävikin minimointi. Ongelmaa on tutkittu laajasti ja sille on kehitetty vakiintunut typologia [17].

Kartonkikoneoptimointi on klassinen yksiulotteinen trimmitysongelma (1D-CSP), jossa pyritään sovittamaan tuotantoyksiköt eli varastorullat mahdollisimman pieneen määrään varastomateriaaleja eli jumborullia. Kartonkikoneoptimoinnissa syntyvää hävikkiä kutsutaan jumbohävikiksi. Osaongelma ratkaistaan laskennallisesti tehokkaalla *first fit decreasing* -algoritmilla [11].

Arkkileikkurioptimointi on kaksiulotteinen trimmitysongelma, jossa arkkitilaukset leikataan varastorullista. Ongelmassa on tuotannosta aiheutuvia epälineaarisia rajoitteita, kuten *vierekkäisten arkkien lukumäärä arkkileikkurin leikkauskuvioissa* sekä ulottuvuuksia lisääviä vapausasteita, kuten *valittavien varastorullaleveyksien lukumäärä* sekä *käytettävä pinta-alan toleranssi*. Haastavuutensa vuoksi ongelma ratkaistaan heuristisella menetelmällä. Tämä menetelmä on esimerkinomainen eikä kaikkia tuotannon asettamia rajoitteita eritellä. Esimerkiksi laitteiston tuottamaa teknistä hävikkiä tai arkkileikkurien kapasiteettirajoitteita ei huomioida.



Kuva 1: Kaavio kartonkituotantoprosessista.

## 2 Aikaisempi tutkimus

### 2.1 Trimmitysongelman typologia

Wäscher et al. esittämän typologian mukaan trimmitysongelma on alakategoria leikkaus- ja pakkausongelmista (cutting and packaging problems)[17]. Tämä typologia, joka perustuu Dychoffin [6] vuonna 1990 tekemään typologiaan, ottaa huomioon uudet tutkimustarpeet ja esittelee johdonmukaisen kategorisoinnin erilaisista ongelmatyypeistä. Typologian mukaan leikkaus- ja pakkausongelmissa on joukko pieniä yksiköitä (small items) ja suuria objekteja (large objects). Pieniä yksiköitä asetetaan suuriin objekteihin siten, että pienet yksiköt eivät leikkaa toisiaan. Ongelmat jaetaan viiden kriteerin mukaan alakategorioihin. Kriteerit ovat geometrinen ulottuvuusien määrä (dimensionality), tehtävän tyyppi (kind of assignment), pienten yksiköiden valikoima (assortment of small items), suurten objektien valikoima (assortment of large items), sekä pienten yksiköiden muoto (shape of small items).[17]

Trimmitysongelman ”tehtävän tyyppi” (kind of assignment) on syötteen minimointi, jossa geometrinen ulottuvuusien määrä ei muutu [17]. Syötteen minimointi tarkoittaa, että suurten objektien arvo pyritään minimoimaan (esim. tarvittavien jumborullien määrä kartonkituotannossa). Lisäksi kaikki pienet yksiköt (esim. varastorullat) pitää leikata suurista objekteista. Pienten yksiköiden valikoima on heikosti heterogeeninen (weakly heterogeneous), eli eri muotoisia yksiköitä on suhteellisen vähän ja yksittäiseen yksikköön kohdistuva kysyntä voi olla yli yhden. Leikkauskuvilla tarkoitetaan sitä, kuinka pieniä yksiköitä asetetaan yksittäiselle suurelle objektille, esimerkiksi kuinka varastorullat leikataan yksittäiseltä jumborullalta. Ongelmasta on useita sovelluksia eri teollisuussektoreilla. Sovelluksia on esimerkiksi paperiteollisuudessa [10], metalliteollisuudessa [16] ja tekstiiliteollisuudessa [14].

## 2.2 Trimmitysongelman ratkaisumenetelmät

Trimmitysongelmasta on yhden ja useamman ulottuvuuden sovelluksia. Yhden ulottuvuuden ongelmaan on kehitetty lineaarista optimointia ja kokonaislukuoptimointia yhdistävä sarakegenerointimenetelmä [7][8]. Mikäli pieniin yksiköihin kohdistuva kysyntä on vähäistä, ongelman voi ratkaista testaamalla kaikki mahdolliset leikkauskuviot niin sanotulla brute-force-algoritmillä. Ongelma voidaan ratkaista myös säiliöntäyttöongelmaan (bin-packing problem) kehitetyillä heuristisella menetelmällä, kuten *first fit decreasing* -algoritmillä [13]. Heurististen menetelmien suorituskykyä on tutkittu paljon. Esimerkiksi huonoimmassa tapauksesessa *first fit decreasing* -algoritmi tuottaa tulokseksi  $11/9 \times Opt + 6/9$ , missä  $Opt$  on suurten objektien määrä optimissa [5]. Tämä on noin 22% optimia huonompi, mikäli suuria objekteja on monta.

Kaksiulotteisen trimmitysongelman ovat esittäneet Gilmore ja Gomory [9]. Ongelmasta on useita eri sovelluksia. Esimerkiksi leikkauskuvioihin on useita erilaisia rajoitteita. Yleensä leikkauskuviot ovat "giljotiinileikkauksia". Giljotiinileikkauksessa leikkaukset tehdään kohtisuoraan suorakulmaiseen varastomateriaaliin nähden joko pysty tai vaakasuunnassa. Leikkaus tehdään koko arkin pituudelta. Jos leikatut arkit pitää ajaa giljotiiniin uudestaan, leikkaus on monivaiheinen. Esimerkiksi Silva ja Alvos [15] ovat esittäneet kokonaislukuoptimointiin perustuvan ratkaisumenetelmän kaksi- ja kolmivaiheiseen giljotiinileikkausoptimointiin. Kaksiulotteisen trimmitysongelman voi ratkaista sarakegenerointimenetelmällä yksiulotteisen ongelman tavoin. Sarakegenerointi on kuitenkin haastavampaa erilaisten leikkauskuviorajoitteiden takia. Sarakegenerointiin on kehitetty heuristisia menetelmiä [3], kokonaislukuoptimoinnin menetelmiä [15] sekä geneettisiä algoritmeja [12]. Joissain trimmitysongelman sovelluksissa hävikkiä voidaan vähentää leikkaamalla hävikkipaloista ylimääräisiä standardikokoisia yksiköitä myöhempää hyödyntämistä varten. Tätä menetelmää on hyödynnetty yhden [4] ja kahden [2] ulottuvuuden ongelmissa.

## 3 Tutkimusmenetelmät

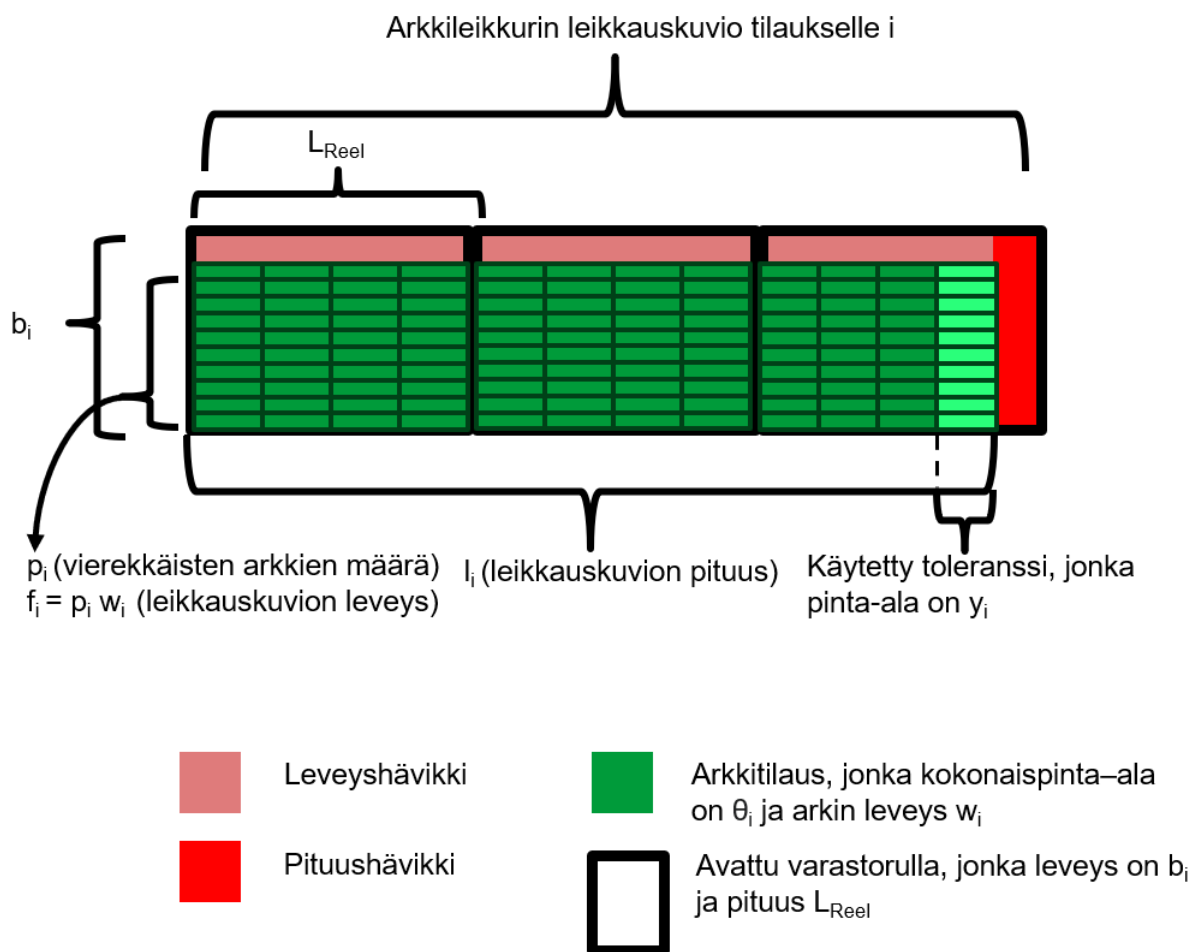
### 3.1 Pääongelma

Pääongelmassa huomioidaan kartonkituotannon kaksi vaihetta, joissa kummassakin syntyy materiaalihävikkiä. Ensimmäisessä vaiheessa varastorullat leikataan kartonkikoneen tuottamista jumborullista. Jumbohävikiksi kutsutaan sitä osaa jumborullasta, jota ei voida hyödyntää. Toisessa vaiheessa arkkitilaukset asetetaan varastorullille ja leikataan arkkileikkurilla. Arkkitilauksessa on annettu tilatun arkin mitat, haluttu kokonaispinta-ala, sekä suhteellinen toleranssi, joka ilmaisee kuinka monta prosenttia tuotettu kokonaispinta-ala saa poiketa tilatusta kokonaispinta-alasta. Arkkitilauksia ajetaan arkkileikkurilla yksi kerrallaan, ja arkkeja voidaan asettaa monta vierekkäin. Varastorullan reunalle jää leveyshävikkiä, ja arkkitilauksen viimeisen rullan loppuun jää joissain tapauksissa pituushävikkiä. Yksittäiselle tilaukselle tehtyä tuotantosuunnitelmaa kutsutaan arkkileikkurin leikkauskuvioksi. Arkkien pituuksia ei tarvitse huomioida optimoinnissa, sillä ne ovat pieniä suhteessa arkkileikkurin leikkauskuvioi-

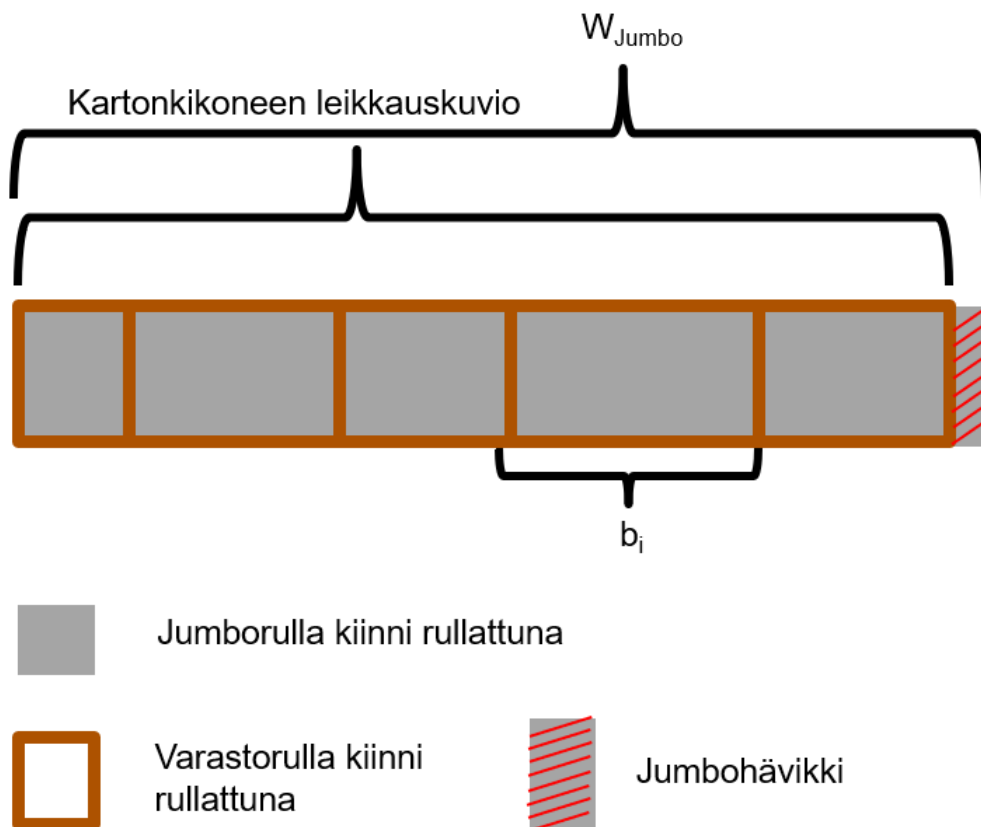


den pituuksiin eivätkä siksi vaikuta tulokseen merkittävästi. (Kartonkituotannossa varastorullan pituus on kokoluokkaa  $10^3m$  ja arkkien pituudet kokoluokkaa  $10^{-1}m$ .)

Pääongelmassa on päätösmuuttujia, jotka vaikuttavat tuotannon eri vaiheissa. Koko tuotantoon vaikuttavia päätösmuuttujia ovat varastorullaleveyksien lukumäärä sekä varastorullaleveydet. Erilaisia varastorullaleveyksiä voidaan tuottaa varastoon valittu lukumäärä, ja samanleveyisiä rullia voidaan tuottaa useita. Arkkileikkurilla yksittäiseen arkkitilaukseen vaikuttavia päätösmuuttujia ovat käytettävä varastorullaleveys, käytettävä toleranssi sekä vierekkäisten arkkien määrä arkkileikkurin leikkauskuviossa. Esimerkki arkkileikkurin leikkauskuviosta on kuvassa 2. Karttonkikoneella yksittäiseen karttonkikoneen leikkauskuvioon vaikuttavia päätösmuuttujia ovat varastorullaleveydet ja -määrät sekä tuotantomäärä. Esimerkki karttonkikoneen leikkauskuviosta on kuvassa 3.



Kuva 2: Esimerkki arkkileikkurin leikkauskuviosta.



Kuva 3: Esimerkki kartonkikoneen leikkauskuviosta.

Pääongelman kohdefunktion voi formuloida yleisesti kaavan (1) mukaisesti ja rajoitusehdot kaavojen (2) - (10) mukaisesti. Täten siis

Päätösmuuttuja	Kuvaus
$y_i$	Pinta-alan toleranssi, joka käytetään arkkitilaukselle $i \in I$
$x_j$	Kartonkikoneen leikkauskuvion $j \in J$ tuotantomäärä
$p_i$	Vierekkäisten arkkioiden lukumäärä arkkileikkurin leikkauskuviossa arkkitilauksella $i \in I$
$n$	Valittavien varastorullaleveyksien lukumäärä
$S = \{s_1, s_2, \dots, s_n\}$	Valittavien varastorullaleveyksien joukko
$B = (b_1, b_2, \dots, b_N)$	Valittujen varastorullien leveydet, jossa $b_i$ on arkkitilaukselle $i \in I$ valittu varastorullan leveys
Parametri	Kuvaus
$L_{Reel}$	Varastorullan ja jumborullan pituus
$W_{Jumbo}$	Jumborullan leveys
$I = \{1, 2, \dots, N\}$	Arkkitilausten joukko, missä $N$ on arkkitilausten lukumäärä
$J$	Mahdollisten kartonkikoneen leikkauskuvioiden joukko
$A$	Jumborullan leikkauskuviomatriisi, jossa $a_{ij}$ on arkkitilauksen $i \in I$ varastorullien määrä kartonkikoneen leikkauskuviossa $j \in J$
$\theta_i$	Arkkitilauksen $i \in I$ kokonaispinta-ala
$w_i$	Arkkitilauksen $i \in I$ arkin leveys
$t_i$	Arkkitilauksen $i \in I$ pinta-alan suurin sallittu suhteellinen toleranssi, $t_i \in [0, 1]$
$S_{\min}$	Varastorullan minimileveys
$S_{\max}$	Varastorullan maksimileveys

Taulukko 1: Pääongelman päätösmuuttujien ja parametrien selitykset.

$$\min \left( \sum_{j \in J} (x_j L_{Reel} W_{Jumbo}) - \sum_{i \in I} (\theta_i + y_i) \right), \quad (1)$$

rajoitusehdoille:

$$L_{Reel} \sum_{j \in J} a_{ij} x_j \geq \frac{\theta_i + y_i}{w_i p_i}, \quad \forall i \in I, \quad (2)$$

$$x_j \in \mathbb{N}, \quad \forall j \in J, \quad (3)$$

$$a_{ij} \in \mathbb{N}, \quad \forall i \in I, \quad (4)$$

$$A_{:,j} \cdot B \leq W_{Jumbo}, \quad \forall j \in J, \quad (5)$$

$$-t_i \theta_i \leq y_i \leq t_i \theta_i, \quad \forall i \in I, \quad (6)$$

$$b_i \geq p_i w_i, \quad \forall i \in I, \quad (7)$$

$$b_i \in S, \quad \forall i \in I, \quad (8)$$

$$S_{\min} \leq s_k \leq S_{\max}, \quad \forall k \in \{1, 2, \dots, n\}, \quad (9)$$

$$s_k \in \mathbb{N}, \quad \forall k \in \{1, 2, \dots, n\}. \quad (10)$$

Pääongelman kohdefunktiossa (1) jumborullien pinta-alasta vähennetään arkkitilausten ja käytettyjen toleranssien pinta-alat. Jäljelle jäävä pinta-ala on kokonaishävikki. Kaava (2) on varastorullien kysyntärajoite. Epäyhtälön vasemmalla puolella on leikattujen varastorullien yhteenlaskettu pituus arkkitilaukselle  $i$  ja oikealla puolella arkkileikkurin leikkauskuvion pituus tilaukselle  $i$ . Kaavassa (5) on kartonkikoneen

leikkauskuvioden leveysrajoite. Epäyhtälön vasemmalla puolella on kartonkikoneen leikkauskuvion  $j$  leveys ja oikealla puolella jumborullan leveys. Kaavassa (6) on toleranssien käyttörajoite. Käytetty toleranssi on arkkitilauksen toleranssien sisällä. Kaavassa (7) on varastorullien leveysrajoite: varastorullan leveys arkkitilaukselle  $i$  on suurempi tai yhtäsuuri kuin arkkileikkurin leikkauskuvion leveys. Kaavoissa (8)-(10) on varastorullien rajoitteet. Kaikki varastorullaleveydet kuuluvat joukkoon  $S$ . Joukon  $S$  alkioit ovat minimi- ja maksimileveyksien sisällä ja ovat luonnollisia lukuja.

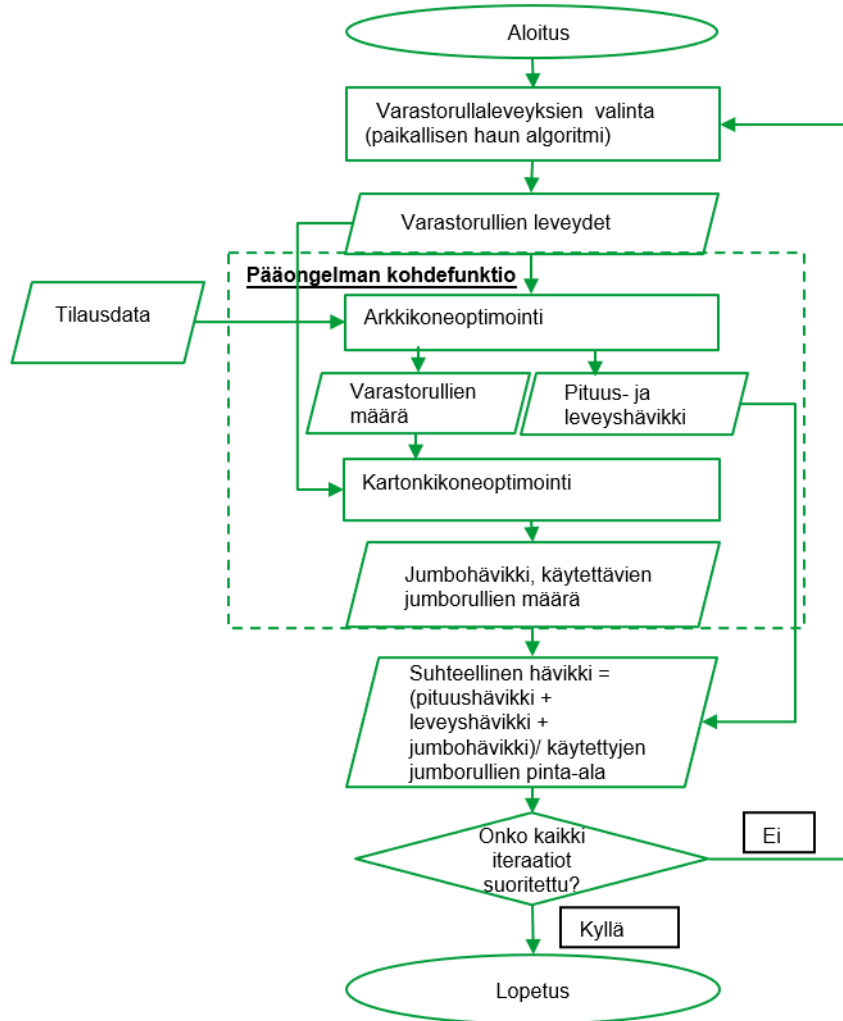
### 3.2 Pääongelman ratkaisumenetelmä

Tässä työssä pääongelma (ks. kuva 4) ratkaistaan siten, että vapausasteena on vain varastorullaleveyksien joukko  $S$ . Heuristinen menetelmä määrittää muiden päätösmuuttujien arvot. Pääongelman kohdefunktiossa on kaksi osaongelmaa: arkkileikkurioptimointi ja kartonkikoneoptimointi. Osaongelmat ratkaistaan erillisinä ja käänteisessä järjestyksessä verrattuna todelliseen tuotantoprosessiin. Arkkileikkurioptimoinnissa jokaiselle arkkitilaukselle valitaan optimaalinen varastorullaleveys ja ongelman ratkaisuna saadaan tarvittavien varastorullien lukumäärä sekä pituus- ja leveyshävikki. Tarvittavien varastorullien lukumäärät ovat lähtötietoina kartonkikoneoptimoinnille. Kartonkikoneoptimoinnissa varastorullat leikataan jumborullilta. Ratkaisuna saadaan jumborullilta jäävä materiaalihävikki eli jumbohävikki sekä tarvittavien jumborullien yhteenlaskettu pinta-ala. Pääongelman kohdefunktio antaa arvona suhteellisen hävikin, joka lasketaan jakamalla arkkileikkurivaiheen ja kartonkikonevaiheen hävikit tuotannon kokonaispinta-alalla,

$$\frac{\Delta_l + \Delta_w + \Delta_j}{L_{Reel} W_{Jumbo} \sum_j x_j}, \quad (11)$$

missä  $\Delta_l$  on pituushävikki,  $\Delta_w$  on leveyshävikki,  $\Delta_j$  jumbohävikki,  $L_{Reel}$  on jumborullan pituus,  $W_{Jumbo}$  on jumborullan leveys ja  $\sum_j x_j$  on tarvittavien jumborullien lukumäärä.

Pääongelman optimointitehtävä on geneerinen globaali optimointiongelma. Ongelma on kombinatorinen, sillä päätösvaihtoehtojen lukumäärä on äärellinen ja päätösvaihtoehdot ovat diskreettejä. Päätösmuuttujien, eli varastorullaleveyksien arvot ovat kokonaislukuja ja yksiköt ovat millimetrejä. Kombinatorisen ongelman ratkaisumenetelmäksi soveltuu stokastinen optimointi. Ratkaisun ei tarvitse lähestyä optimitulosta nopeasti, koska pääongelman kohdefunktion laskeminen kestää i7-1355U-prosessorilla ja Python-ohjelmointikielellä vain noin 0,001 sekuntia, kun käytetään tähän kandidaattityöhön valittuja parametrejä ja esimerkkitietoa. Nopean laskenta-ajan takia ongelma ratkaistaan yksinkertaisella paikallisen haun algoritmilla, jonka toimintaperiaatetta havainnollistaa kuva 5. Ongelman voisi ratkaista myös esimerkiksi metaheuristisella algoritmilla, kuten geneettisellä algoritmilla [1].



Kuva 4: Pääongelman ratkaisumenetelmän vuokaavio.

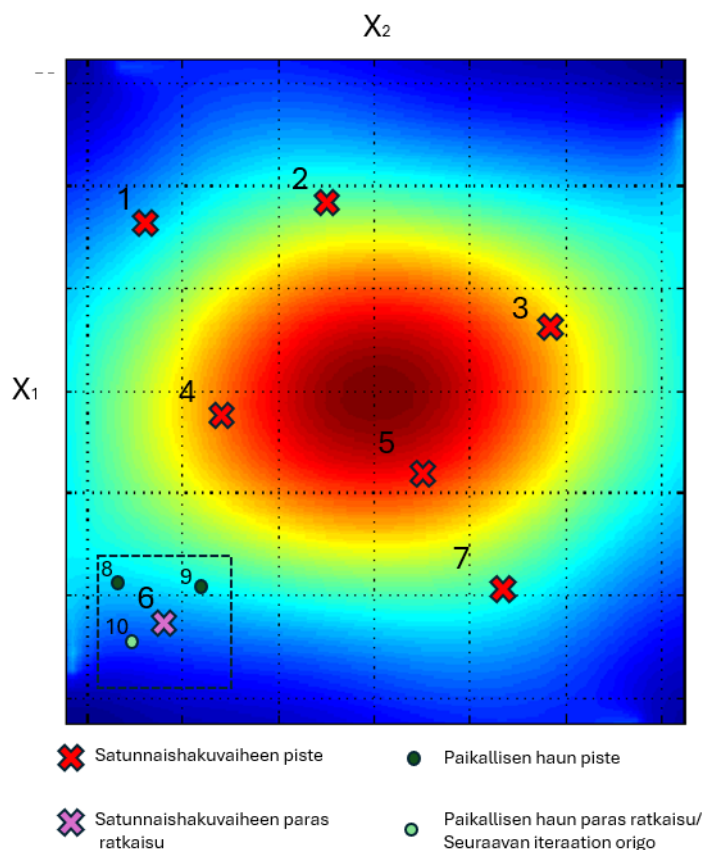
### 3.3 Paikallisen haun algoritmi

Tässä työssä käytettävä paikallisen haun algoritmi on kaksivaiheinen. Ensimmäisessä vaiheessa (satunnaishaku) generoidaan ja testataan ratkaisuavaruuden pisteitä satunnaisesti. Toisessa vaiheessa (paikallinen haku) optimia haetaan parhaiden ratkaisujen lähettäviltä siten, että jokaisella iteraatiolla hakualueen keskipisteeksi valitaan edellisen iteraation paras ratkaisu. Jos parhaan ratkaisun läheltä ei löydy parempaa pistettä, hakualueen säde kasvaa. Kun viimeisimmästä parannuksesta on  $r_{max}$  iteraatiota, hakualue pienenee taas pienimmäksi mahdolliseksi. Hakualueen säde lasketaan kaavalla (12):

$$2^{l_{imp\_i} \bmod r_{max}}, \quad (12)$$

missä  $l_{imp\_i}$  on iteraatioiden lukumäärä edellisestä parannuksesta ratkaisulle  $i$  ja  $r_{max}$  määrittää suurimman mahdollisen hakualueen eksponentin. Kuvassa 5 on havainnollistettu paikallisen haun ensimmäinen iteraatio yhdellä hakupisteellä. Taulukossa 2

on listattu optimoinnissa käytettävät parametrit.



Kuva 5: Kuvaajassa on kuvitteellinen kohdefunktio kahden muuttujan tapauksessa. Lämpökartta kuvastaa funktion arvoa pisteessä. Pisteet 1-7 ovat satunnaishakuvaiheen pisteitä. Piste 6 on satunnaishakuvaiheen paras ratkaisu ja paikallisen haun ensimmäisen iteraation hakualueen keskipiste. Katkoviivalla on rajattu paikallisen haun alue. Pisteet 8-10 ovat paikallisen haun pisteitä. Piste 10 on ensimmäisen iteraation paras ratkaisu ja toisen iteraation hakualueen keskipiste.

Symboli	Kuvaus
$X_{\min}, X_{\max}$	Hakualueen rajat
$N_{RS}$	Satunnaishakuvaiheen hakupisteiden lukumäärä
$N_{LS}$	Valittujen ratkaisujen lukumäärä paikallisen haun vaiheeseessa
$I_{\max}$	Paikallisen haun vaiheen iteraatioiden lukumäärä
$L_{\text{imp}}$	$(l_{\text{imp}_1}, l_{\text{imp}_2}, \dots, l_{\text{imp}_{N_{LS}}})$ , missä $l_{\text{imp}_i}$ on iteraatioiden lukumäärä edellisestä parannuksesta ratkaisulla $i$ , $i \in \{1, 2, \dots, N_{LS}\}$
$N_{\text{neigh}}$	Testattavien pisteiden lukumäärä yksittäisen ratkaisun ympärillä
$r_{\max}$	Suurin eksponentti hakualueen säteessä

Taulukko 2: Paikallisen haun algoritmin parametrit.

### 3.4 Arkkileikkurioptimointi

Arkkileikkurioptimointi on ”usean varastomateriaalin kaksiulotteinen trimmitysongelma” (multiple stock-size cutting stock problem), jossa suuret objektit ovat varastorullia ja näistä leikattavat pienet yksiköt arkkitilauksia. Tavoitteena on minimoida materiaalihävikki. Varastorullia on useita (esim. 10), eli suurten objektien valikoima on heikosti heterogeeninen. Yksittäisessä arkkitilauksessa on ilmoitettu arkin koko sekä kokonaispinta-alan suhteellinen toleranssi. Arkkileikkurit ajavat vain yhtä arkkitilausta kerrallaan, mikä osin yksinkertaistaa ongelmaa. Arkkitilausten toleranssien takia arkkien lukumäärä ei ole kiinnitetty, minkä takia ongelma ei ole Wäscher et al. typologian standardiongelmien mukainen [17].

Optimointitehtävässä varastorullaleveyksien joukko  $S$  on kiinnitetty, sillä varastorullien valinta pääongelman ratkaisumenetelmässä asettaa sen lähtötietona (ks. kuva 4). Päästömuuttujia ovat arkkitilauksen vaatima varastorullatarve  $d_i$ , tilauksen käyttämä varastorullaleveys  $b_i$ , pinta alan toleranssi  $y_i$  sekä vierekkäisten arkkien määrä arkkileikkurin leikkauskuviossa  $p_i$ . Tässä työssä esiteltävä optimointitehtävä on esimerkinomainen eikä sisällä kaikkia tuotannon asettamia rajoitteita. Työssä ei myöskään hyödynnetä arkkileikkurin leikkauskuvion loppuun jäävää pituushävikkiä, jota voisi käyttää muiden arkkitilausten leikkaamiseen. Yksinkertaistettu ongelma on kaavoissa (13)-(18).

Arkkileikkurioptimoinnin kohdefunktio on

$$\min \sum_i (d_i b_i L_{Reel} - (\theta_i + y_i)). \quad (13)$$

Rajoitusehdot ovat

$$d_i L_{Reel} \geq \frac{\theta_i + y_i}{w_i p_i}, \quad \forall i \in I, \quad (14)$$

$$d_i \in \mathbb{N}, \quad \forall i \in I, \quad (15)$$

$$b_i \geq p_i w_i, \quad \forall i \in I, \quad (16)$$

$$-t_i \theta_i \leq y_i \leq t_i \theta_i, \quad \forall i \in I, \quad (17)$$

$$b_i \in S, \quad \forall i \in I. \quad (18)$$

Kaava (13) on arkkileikkurioptimoinnin kohdefunktio. Varastorullien pinta-alasta vähennetään tuotettujen arkkitilausten pinta-alat. Kaavassa (14) on varastorullien kysyntärajoite. Epäyhtälön vasemmalla puolella on leikattujen varastorullien yhteenlaskettu pituus arkkitilaukselle  $i$  ja oikealla puolella arkkileikkurin leikkauskuvion pituus tilaukselle  $i$ . Kaavassa (16) on varastorullan leveysrajoite. Varastorullan leveys arkkitilaukselle  $i$  on suurempi tai yhtäsuuri, kuin arkkileikkurin leikkauskuvion leveys. Kaavassa (6) on toleranssien käyttörajoite.

Ongelma ratkaistaan heuristisella menetelmällä (algoritmi 1), jonka tavoitteena on luoda arkkileikkurin leikkauskuviot (ks. kuva 2) siten, että pituus- ja leveyshävikit minimoituvat. Heuristisessa ratkaisussa (algoritmi 1) on kaksi vaihetta, joilla kokonaishävikkiä voi pienentää:

1. Optimaalisen varastorullaleveyden valinta, joka minimoi leveyshävikin.
2. Pituushävikin minimointi arkkitilaustoleransseja käyttämällä.

Leveyshävikki lasketaan kaavalla

$$(b_i - p_i w_i) \times l_i = \frac{\theta_i (b_i - p_i w_i)}{p_i w_i}. \quad (19)$$

Optimaalinen varastorullaleveys saadaan vertailemalla kaavan

$$\frac{(b_i - p_i w_i)}{p_i} \quad (20)$$

arvoja, sillä  $\theta_i/w_i$  on vakio. Optimaalisen varastorullaleveyden valinta tehdään algoritmin 1 riveillä (1)-(16).

Pituushävikki minimoidaan käyttämällä arkkitilaustoleransseja (rivit (17)-(37)). Pituushävikkiä voi vähentää poistamalla arkkitilauksen viimeinen varastorulla, jolloin pituushävikkiä ei muodostu, tai käyttämällä toleransseja viimeisen rullan maksimaaliseen täyttämiseen. Ensisijaisesti algoritmi yrittää poistaa viimeisen varastorullan, mikäli toleranssi sallii tämän. Muutoin toleranssia käytetään viimeisen varastorullan täyttämiseen (rivit (28)-(34)). Algoritmi 1 laskee lopuksi pituus- ja leveyshävikkien pinta-alat (rivit (38)-(46)). Arkkileikkurioptimoinnin suorittava funktio on kuvattu riveillä (47)-(51).

Parametri	Kuvaus
$W$	$(w_1, w_2, \dots, w_N)$ , missä $w_i$ on arkin leveys arkkitilauksella $i \in I$
$T$	$(t_1, t_2, \dots, t_N)$ , missä $t_i$ on arkkitilauksien $i \in I$ suhteellinen toleranssi
$\Theta$	$(\theta_1, \theta_2, \dots, \theta_N)$ , missä $\theta_i$ on arkkitilauksen $i \in I$ pinta-ala
Johdettu muuttuja	Kuvaus
$\Delta_{w\_opt}$	Pienin leveyshävikki
$b_{opt}$	Varastorullaleveys, jolla saavutetaan pienin leveyshävikki
$f_{opt}$	Arkkileikkurin leikkauskuvion suurin mahdollinen leveys rullaleveydellä $b_{opt}$
$F$	$(f_1, f_2, \dots, f_N)$ , missä $f_i$ on arkkileikkurin leikkauskuvion leveys arkkitilaukselle $i$
$D_p$	Etäisyys edelliseen varastorullaan
$D_n$	Etäisyys seuraavaan varastorullaan
$L$	$(l_1, l_2, \dots, l_N)$ , missä $l_i$ on arkkileikkurin leikkauskuvion pituus arkkitilauksella $i \in I$
Palautusarvo	Kuvaus
$\Delta_l$	Pituushävikki
$\Delta_w$	Leveyshävikki
$D$	$(d_1, d_2, \dots, d_N)$ , missä $d_i$ on tarvittavien varastorullien lukumäärä arkkitilauksella $i \in I$

Taulukko 3: Arkkileikkurialgoritmin (algoritmi 1) parametrien, johdettujen muuttujien ja palautusarvojen kuvaukset.



---

**Algorithm 1** Optimize Middle Width and Minimize Length Loss
 

---

```

1: procedure OPTIMALMIDDLE( $S, w_i$ )
2:    $\Delta_{w\_opt} \leftarrow \infty$ 
3:    $b_{opt} \leftarrow 0$ 
4:    $f_{opt} \leftarrow 0$ 
5:   for  $k = 1$  to  $n$  do
6:      $p_{trial} \leftarrow \lfloor s_k/w_i \rfloor$ 
7:      $f_{trial} \leftarrow p_{trial} \times w_i$ 
8:      $\Delta_{trial} \leftarrow (s_k - f_{trial})/p_{trial}$ 
9:     if  $\Delta_{trial} < \Delta_{w\_opt}$  then
10:       $\Delta_{w\_opt} \leftarrow \Delta_{trial}$ 
11:       $b_{opt} \leftarrow s_k$ 
12:       $f_{opt} \leftarrow f_{trial}$ 
13:     end if
14:   end for
15:   return  $b_{opt}, f_{opt}$ 
16: end procedure
17: procedure MINIMIZELENGTHLOSS( $S$ )
18:    $B \leftarrow$  empty array of size  $(1 \times N)$ , with elements  $b_i$ 
19:    $F \leftarrow$  empty array of size  $(1 \times N)$ , with elements  $f_i$ 
20:    $L \leftarrow$  empty array of size  $(1 \times N)$ , with elements  $l_i$ 
21:    $D \leftarrow$  empty array of size  $(1 \times N)$ , with elements  $d_i$ 
22:   for  $i = 1$  to  $N$  do
23:      $b_i, f_i \leftarrow$  OPTIMALMIDDLE( $S, w_i$ )
24:      $l_i \leftarrow \theta_i/f_i$ 
25:      $d_i \leftarrow \lceil l_i/L_{Reel} \rceil$ 
26:      $D_p \leftarrow l_i - L_{Reel} \times (d_i - 1)$ 
27:      $D_n \leftarrow L_{Reel} \times d_i - l_i$ 
28:     if  $D_p \leq t_i \times l_i$  then
29:        $l_i \leftarrow l_i - D_p$ 
30:        $d_i \leftarrow \lceil l_i/L_{Reel} \rceil$ 
31:     else
32:        $l_i \leftarrow \min((1 + t_i) \times l_i, l_i + D_n)$ 
33:        $d_i \leftarrow \lceil l_i/L_{Reel} \rceil$ 
34:     end if
35:   end for
36:   return  $L, B, F, D$ 
37: end procedure
38: procedure CALCULATELOSSES( $L, B, F$ )
39:    $\Delta_l \leftarrow 0$ 
40:    $\Delta_w \leftarrow 0$ 
41:   for  $i = 1$  to  $N$  do
42:      $\Delta_l \leftarrow \Delta_l + (L_{Reel} \times d_i - l_i) \times b_i$ 
43:      $\Delta_w \leftarrow \Delta_w + (b_i - f_i) \times l_i$ 
44:   end for
45:   return  $\Delta_l, \Delta_w$ 
46: end procedure
47: procedure SHEETCUTTEROPTIMIZATION( $S$ )
48:    $L, B, F, D \leftarrow$  MINIMIZELENGTHLOSS( $S$ )
49:    $\Delta_l, \Delta_w \leftarrow$  CALCULATELOSSES( $L, B, F$ )
50:   return  $\Delta_l, \Delta_w, D$ 
51: end procedure

```

---

### 3.5 Kartonkikoneoptimointi

Kartonkikoneoptimointi on klassinen yhden varastomateriaalin yksiulotteinen trimmitysongelma (single stock material cutting stock problem), joka tunnetaan myös lyhenteellä 1D-CSP. Kartonkikoneoptimoinnissa suuret objektit ovat jumborullia ja pienet yksiköt varastorullia. Kysynnän mukaiset varastorullat leikataan jumborullilta ks. kuva 3. Pääongelman ratkaisumenetelmässä (ks. kuva 4) arkkileikkurioptimointi asettaa tilauksen varastorullatarpeen  $d_i$ , joten se on optimoinnissa kiinnitetty. Päättömuuttujana on kartonkikoneen leikkauskuvion tuotantomäärä  $x_j$ . Kartonkikoneoptimoinnin kohdefunktio on

$$\min \sum_{j \in J} x_j. \quad (21)$$

Rajoitusehdot ovat

$$\sum_{j \in J} a_{ij} x_j \geq d_i, \quad \forall i \in I, \quad (22)$$

$$x_j \geq 0, \quad \forall j \in J, \quad (23)$$

$$x_j \in \mathbb{N}, \quad \forall j \in J. \quad (24)$$

Kaavassa (21) on kartonkikoneoptimoinnin kohdefunktio. Tavoitteena on minimoida käytettävien jumborullien lukumäärä. Kaavassa (22) on leikattavien varastorullien kysyntärajoite. Epäyhtälön vasemmalla puolella on varastorullaleveyksien lukumäärä kartonkikoneen leikkauskuvioissa ja oikealla varastorullaleveyksien kysyntä.

Kartonkikoneoptimointiin käytetään heuristista *first fit decreasing* -algoritmia. Lähtötietoina algoritmi saa listan tarvittavista varastorullaleveyksistä. Parametrina on jumborullan leveys. Tuloksena saadaan tarvittavien jumborullien lukumäärä sekä jumbohävikki, joka on

$$\Delta_j = L_{Reel}(W_{Jumbo} \sum_j x_j - \sum_i d_i b_i). \quad (25)$$

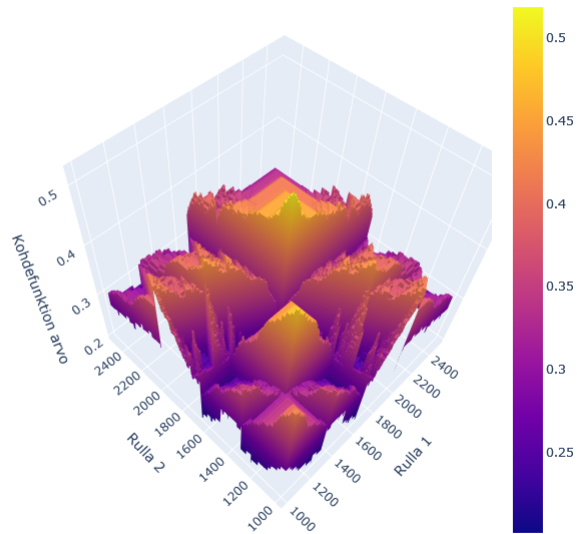
## 4 Tulokset

Pääongelman optimointitehtävää varten generoidaan esimerkkidata, jossa on 100 satunnaisesti luotua arkkitilausta. Taulukossa 4 on esitetty optimoinnissa käytettävät parametrit.

<b>Dataparametri</b>	<b>Kuvaus</b>	<b>Arvo</b>
$L_{Reel}$	Varastorullan ja jumborullan pituus	5000m
$W_{Jumbo}$	Jumborullan leveys	5m
$N$	Tilausten lukumäärä	100
$\theta_i$	Arkkitehtilauksen $i$ kokonaispinta-ala	satunnaisluku tasajakaumasta $\sim \mathcal{U}(3333; 20000)$ (m <sup>2</sup> )
$w_i$	Arkkitehtilauksen $i$ arkin leveys	satunnaisluku tasajakaumasta $\sim \mathcal{U}(0, 2; 0, 7)$ (m)
$t_i$	Arkkitehtilauksen $i$ suhteellinen toleranssi	satunnaisluku tasajakaumasta $\sim \mathcal{U}(0, 02; 0, 1)$
<b>Optimointialgoritmin parametri</b>	<b>Kuvaus</b>	<b>Arvo</b>
$X_{\min} = W_{\min}$	Varastorullan minimileveys	1m
$X_{\max} = W_{\max}$	Varastorullan maksimileveys	2.5m
$N_{RS}$	Satunnaishakuvaiheen hakupisteiden lukumäärä	50000
$N_{LS}$	Parhaiden ratkaisujen lukumäärä paikallisessa haussa	30
$I_{\max}$	Paikallisen haun iteraatioiden lukumäärä	200
$N_{\text{neigh}}$	Testattavien pisteiden lukumäärä yksittäisen ratkaisun ympärillä	100
$r_{\max}$	Suurin eksponentti hakualueen säteessä	10

Taulukko 4: Optimoinnin parametrien arvot.

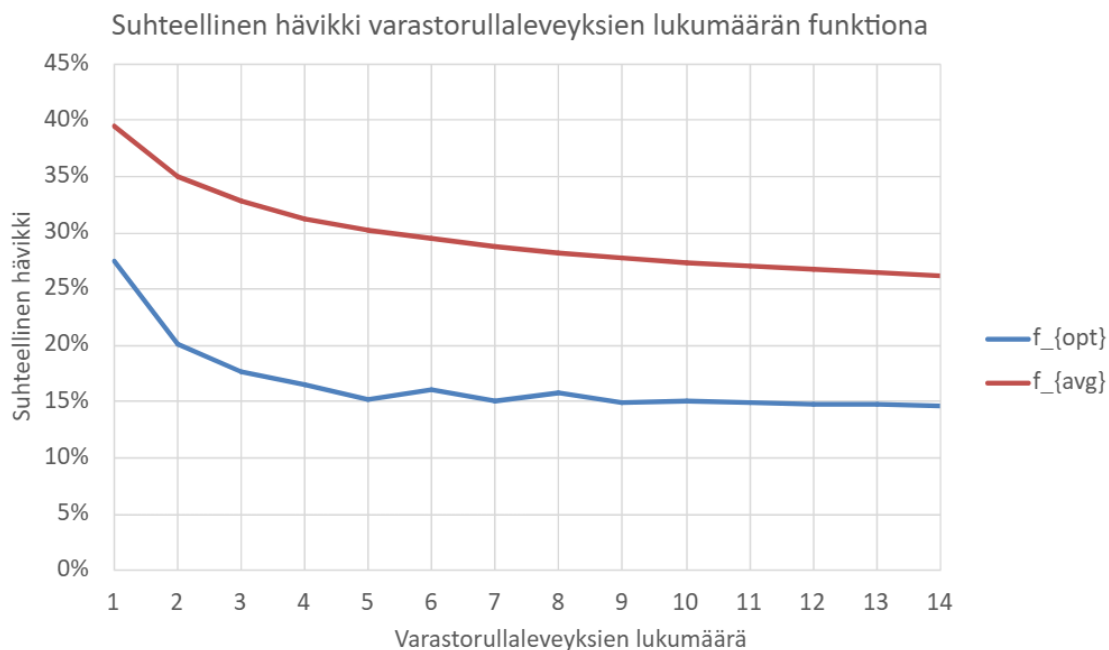
Pääongelman optimointi ajetaan 1-14 varastorullaleveydellä. Taulukosta 5 nähdään, että algoritmin löytämä optimi on selvästi parempi verrattuna satunnaishakuvaiheen keskimääräiseen arvoon. Kuvasta 6 huomataan, että pääongelman kohdefunktion arvot vaihtelevat jyrkästi kahden varastorullan tapauksessa. Kuvasta 7 huomataan, että optimissa suhteellinen hävikki vähenee varastorullaleveyksien lukumäärän kasvaessa. Optimoinnin paras tulos löytyi neljällätoista varastorullaleveydellä, jolloin suhteellinen hävikki oli noin 14,6% ja satunnaishakuvaiheen keskimääräinen suhteellinen hävikki noin 26,2%.

Kohdefunktio,  $n = 2$ 

Kuva 6: Ratkaisuvavaruus kahden varastorullan tapauksessa.

n	$f_{opt}$	Avg(f)	$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$	$s_9$	$s_{10}$	$s_{11}$	$s_{12}$	$s_{13}$	$s_{14}$
1	0.275	0.395	1665													
2	0.201	0.350	1665	1000												
3	0.176	0.328	1745	1502	1000											
4	0.165	0.313	1743	1504	1229	1000										
5	0.152	0.303	1748	1503	1325	1246	1000									
6	0.160	0.295	1803	1716	1374	1507	1304	1000								
7	0.151	0.288	1000	1578	1258	1506	1746	1328	1171							
8	0.158	0.282	1953	1763	1663	1573	1415	1246	1147	1092						
9	0.149	0.278	1796	1717	1624	1581	1503	1374	1327	1070	1000					
10	0.150	0.274	1945	1821	1751	1573	1553	1468	1296	1037	1093	1000				
11	0.149	0.270	1825	1796	1743	1749	1506	1311	1288	1354	1275	1000	1063			
12	0.148	0.267	1953	1799	1741	1483	1445	1319	1349	1269	1163	1089	1000	1041		
13	0.148	0.265	1969	1812	1690	1746	1913	1695	1709	1480	1359	1266	1094	1037	1134	
14	0.146	0.262	1946	1741	1695	1797	1698	1574	1490	1320	1358	1259	1143	1067	1095	1018

Taulukko 5: Tulostaulukossa  $n$  on eri varastorullaleveyksien lukumäärää optimoinnissa.  $f_{opt}$  on löydetty optimaalinen suhteellinen hävikki.  $Avg(f)$  on satunnaishakuvaiheen keskimääräinen suhteellinen hävikki.  $s_k$  on optimiratkaisun varastorullan leveys.



Kuva 7: Suhteellinen hävikki varastorullaleveyksien lukumäärän funktiona.

## 5 Yhteenveto

Tässä työssä esiteltiin kartonkituotantoa simuloiva optimointimenetelmä, joka löysi hyviä tuloksia ratkaisuvaihtelussa, jossa suhteellinen hävikki vaihteli paljon erilaisilla varastorullajoukoilla. Esimerkiksi neljäntoista varastorullaleveyden optimoinnissa satunnaishakuvaiheen keskimääräinen suhteellinen hävikki oli noin 26,2%, kun taas optimissa se oli noin 14,6%. Käytetty data oli satunnaisesti generoitua ja parametrit olivat esimerkinomaisia. Parametrien arvot ja data riippuvat sovelluksesta. Tuotannon rakenne, jossa tuotantomateriaali leikataan ensin suuremmasta raaka-aineesta, varastoidaan ja jalostetaan lopputuotteeksi, on yleinen paperi ja kartonkiteollisuudessa. Esiteltyä ratkaisumenetelmää voi hyödyntää vastaavissa sovelluksissa, sillä menetelmään on helppo lisätä uusia rajoitteita ja toimintalogiikoita. Jatkotutkimuksissa voi tarkastella esimerkiksi varastoinnin mahdollistamaa materiaalitehokkuutta kartonkikoneoptimoinnissa. Kartonkikoneoptimointiin voi myös soveltaa laskennallisesti haastavampia, mutta parempia tuloksia tuottavia menetelmiä.

## Viitteet

- [1] M. Abdel-Basset, L. Abdel-Fatah, ja A. K. Sangaiah. Metaheuristic algorithms: A comprehensive review. *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications*, pages 185–231, 2018.
- [2] R. Andrade, E. G. Birgin, ja R. Morabito. Two-stage two-dimensional guillotine cutting stock problems with usable leftover. *International Transactions in Operational Research*, 23(1-2):121–145, 2016.
- [3] M.-B. Aryanezhad, N. F. Hashemi, A. Makui, ja H. Javanshir. A simple approach to the two-dimensional guillotine cutting stock problem. *Journal of Industrial Engineering International*, 8:1–10, 2012.
- [4] A. C. Cherri, M. N. Arenales, H. H. Yanasse, K. C. Poldi, ja A. C. G. Vianna. The one-dimensional cutting stock problem with usable leftovers—A survey. *European Journal of Operational Research*, 236(2):395–402, 2014.
- [5] G. Dósa. The tight bound of first fit decreasing bin-packing algorithm is FFD (i)  $11/9$  opt (i) +  $6/9$ . Teoksessa *International Symposium on Combinatorics, Algorithms, Probabilistic and Experimental Methodologies*, pages 1–11. Springer, 2007.
- [6] H. Dyckhoff. A typology of cutting and packing problems. *European Journal of Operational Research*, 44(2):145–159, 1990.
- [7] P. C. Gilmore ja R. E. Gomory. A linear programming approach to the cutting-stock problem. *Operations Research*, 9(6):849–859, 1961.
- [8] P. C. Gilmore ja R. E. Gomory. A linear programming approach to the cutting stock problem—Part II. *Operations Research*, 11(6):863–888, 1963.
- [9] P. C. Gilmore ja R. E. Gomory. Multistage cutting stock problems of two and more dimensions. *Operations Research*, 13(1):94–120, 1965.
- [10] J. Kallrath, S. Rebennack, J. Kallrath, ja R. Kusche. Solving real-world cutting stock-problems in the paper industry: Mathematical approaches, experience and challenges. *European Journal of Operational Research*, 238(1):374–389, 2014.
- [11] P. Ongkunaruk. Asymptotic worst-case analyses for the open bin packing problem. PhD thesis, Virginia Tech, 2005.
- [12] G. C. Onwubolu ja M. Mutingi. A genetic algorithm approach for the cutting stock problem. *Journal of Intelligent Manufacturing*, 14:209–218, 2003.
- [13] B. Rieck. Basic analysis of bin-packing heuristics. *arXiv preprint arXiv:2104.12235*, 2021.

- [14] K. H. Salem, E. Silva, J. F. Oliveira, ja M. A. Carravilla. Mathematical models for the two-dimensional variable-sized cutting stock problem in the home textile industry. *European Journal of Operational Research*, 306(2):549–566, 2023.
- [15] E. Silva, F. Alvelos, ja J. M. V. De Carvalho. An integer programming model for two-and three-stage two-dimensional cutting stock problems. *European Journal of Operational Research*, 205(3):699–708, 2010.
- [16] D. Tanir, O. Ugurlu, A. Guler, ja U. Nuriyev. One-dimensional cutting stock problem with divisible items: A case study in steel industry. *TWMS Journal of Applied and Engineering Mathematics*, 9(3):473–484, 2019.
- [17] G. Wäscher, H. Haußner, ja H. Schumann. An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183(3):1109–1130, 2007.