

Heikki Toivo

OPTIMOINTIMENETELMIIN PERUSTU- VAN TEOLLISEN ONGELMANRATKAI- SUKYVYN KEHITYS

Kandidaatintyö
Tekniikan ja luonnontieteiden tiedekunta
Huhtikuu 2020

TIIVISTELMÄ

Heikki Toivo: Optimointimenetelmiin perustuvan teollisen ongelmanratkaisukyvyyn kehitys
Tampereen yliopisto
Konetekniikan tutkinto-ohjelma
Kandidaatintyö
Huhtikuu 2020

Matemaattisen optimoinnin hyödyntämisellä teollisuudessa voi olla hyvin suuri merkitys yrityksen liiketoiminnalle. Hyvin optimoidulla toiminnalla voidaan saavuttaa selvää kilpailuetua muihin toimijoihin verrattuna. Tämän asian merkitys korostuu entisestään globaalissa maailmassa, jossa yrityksen kilpailualue on entistä suurempi. Työn tavoitteena on tutustua erilaisiin matemaattisiin optimointimenetelmiin ja selvittää miten näitä voidaan hyödyntää teollisuudessa.

Työ jakaantuu neljään eri pääosaan. Ensimmäisessä osassa käsitellään matemaattisen optimoinnin taustaa teoreettisella tasolla kirjallisuusselvityksen pohjalta. Toisessa osassa esitellään yleisimpiä teollisuudessa kohdattavia optimointiongelmia. Aineisto toiseen osioon on kerätty suurimmaksi osaksi haastattelututkimuksella ja osittain kirjallisuuden pohjalta. Työn kolmannessa osiossa tarkastellaan kirjallisuusselvityksen pohjalta parhaiten soveltuvia ratkaisumenetelmiä seuraaviin ongelmiin: kokoonpanotuotannon optimointi, juomateollisuuden pullotusongelma, työntöohjatun raaka-ainekäytön optimointi ja kaksiulotteinen leikkausongelma. Viimeisessä pääosiossa esitellään teollisuuden optimointiongelmien ratkaisemiseen soveltuvia ohjelmakirjastoja ja muita työkaluja. Näitä tarkastellessa keskitytään pääasiassa .NET Core -yhteensopiviin sovelluksiin.

Työn perusteella on saatu muodostettua suositukset sille, minkälaisilla ratkaisumenetelmillä erityyppisiä teollisuudessa kohdattavia optimointiongelmia voitaisiin pyrkiä ratkaisemaan. Kokoonpanotuotannon optimointiin suositukseksi saatiin muurahaisyhdyskunta algoritmit, juomateollisuuden pullotusongelmaan erilaiset sekalukuohjelmointimallit sekä näiden yhdistelmä geneettisten algoritmien kanssa. Työntöohjatun raaka-ainekäytön optimointiin suositukseksi saatiin sekalukuohjelmointimallit ja geneettiset algoritmit ja lopuksi kaksiulotteiseen leikkausongelmaan niin sanotut SHP ja RCCG algoritmit. Näitä suosituksia voidaan pitää hyvänä lähtökohtana kyseisten ongelmien ratkaisemisessa, koska ne perustuvat kirjallisuusselvityksen tuloksiin. Kirjallisuusselvityksessä käytetty aineisto koostuu suurimmilta osin tieteellisistä julkaisuista, pääasiassa tutkimusartikkeleista ja optimointiin liittyvistä kirjoista. Täten työn tuloksia voidaan pitää kohtalaisen luotettavina huomioiden toki tietyt rajoitteet, kuten jokaisen optimointiongelman ainutlaatuisuus.

Avainsanat: Kombinatorinen optimointi, teollinen optimointi, heuristiikka

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

ALKUSANAT

Tämä kandidaatintyö koskien matemaattisen optimoinnin hyödyntämistä teollisuudessa on tehty osana tekniikan kandidaatintutkintoani Tampereen yliopistolle.

Iso kiitos mielenkiintoisesta ja haastavasta aiheesta Pinja Oy:lle. Lisäksi erityiskiitos kaikille Pinja Oy:llä haastatteluun osallistuneille henkilöille. Tästä haastattelusta sain paljon materiaalia työhöni ja lisäksi paljon yleistä tietoa käsiteltävästä aihepiiristä.

Lopuksi vielä kiitokset työni ohjaajalle Ville Toivoselle, jolta sain aina hyviä vinkkejä työni eteenpäinviemiseen ja kehittämiseen. Nämä olivat todella tarpeellisia neuvoja tätä työtä tehdessä.

Tampereella, 17.04.2020

Heikki Toivo

SISÄLLYSLUETTELO

1. JOHDANTO	1
1.1 Tutkimusongelma ja -menetelmät	1
1.2 Työn rajaukset ja rakenne	2
2. MATEMAATTINEN OPTIMOINTI	3
2.1 Käytännön sovellukset matemaattisessa optimoinnissa	4
2.2 Kombinatorinen optimointi	5
2.3 Optimointitehtävien ratkaiseminen	5
2.3.1 Eksaktit menetelmät	5
2.3.2 Heuristiikka ja metaheuristiikka	6
3. OPTIMOINTIONGELMAT TEOLLISUUDESSA	8
3.1 Yleisiä optimointiongelmia	9
3.2 Elintarviketeollisuus	9
3.2.1 Juomateollisuuden pullotusongelma	10
3.2.2 Lihan leikkaamisen optimointi	11
3.3 Kokoonpanoteollisuus	12
3.3.1 Kokoonpanotuotannon optimointi	12
3.3.2 Resurssikäytön optimointi	13
3.4 Puu- ja sahateollisuus	14
3.4.1 Tukin sahauksen optimointi	14
3.4.2 Ladontakuvion optimointi	15
4. RATKAISUALGORITMIT TEOLLISUUDEN OPTIMOINTIONGELMIIN	17
4.1 Ratkaisualgoritmit teollisuudessa	17
4.2 Kokoonpanotuotannon optimointi	19
4.3 Juomateollisuuden pullotusongelma	20
4.4 Työntöohjatun raaka-ainekäytön optimointi	22
4.5 Kaksiulotteinen leikkausongelma	24
5. OHJELMAKIRJASTOT JA TYÖKALUT TEOLLISEEN OPTIMOINTIIN	27
5.1 .NET Core -yhteensopivat kirjastot ja työkalut	27
5.2 Muut	28
6. PÄÄTELMÄT	30
LÄHTEET	32

LYHENTEET JA MERKINNÄT

ACO	ant colony algorithms	muurahaisyhdyskunta algoritmit
ALB	assembly line balancing	kokoonpanolinjantasapainotus
ASP	assembly sequence planning	kokoonpanojärjestyksensuunnittelu
CG	column generation	ratkaisumenetelmä
GA	genetic algorithms	geneettiset algoritmit
GAMP	genetic algorithm/mathematical programming	ratkaisualgoritmi
LVL	laminated veneer lumber	viilupuu
MA	memetic algorithms	memeettiset algoritmit
MILP	mixed-integer linear programming	lineaarinen sekalukuohjelmointi
MIP	mixed-integer programming	sekalukuohjelmointi
PSO	particle swarm optimization	partikkelipilvioptimointi
RA	relaxation approach	ratkaisualgoritmi
RCCG	repeated constrained column generation	ratkaisualgoritmi
SA	simulated annealing	simuloitu jäähdytys
SHP	sequential heuristic procedure	ratkaisualgoritmi
SLOPP	single large object placement problem	yhden laajan objektin sijoitteluongelma
TS	tabu search	tabu haku
TSP	travelling salesman problem	kauppamatkustajanongelma

1. JOHDANTO

Optimoinnilla pyritään saavuttamaan mahdollisimman hyvä tulos haluttujen muuttujien suhteen. Jopa yksittäinen ihminen saattaa pyrkiä optimoimaan toimintaansa päivittäin esimerkiksi valitsemalla taloudellisimman reitin työmatkalleen tai pohtimalla miten vuotuisen lomamatkan saisi varatuksi pienimmillä mahdollisilla kustannuksilla. Optimoinnin merkitys korostuu entisestään silloin kun hyvin suoritettulla optimoinnilla voidaan saavuttaa esimerkiksi merkittävästi pienemmät kustannukset kuin ei-optimaalisessa tilanteessa. Teollisuudessa ja muussa liiketoiminnassa optimoinnilla voidaan saavuttaa merkittäviä hyötyjä. Parhaimmillaan se voi tarkoittaa selvää kilpailuetua muihin saman toimialan toimijoihin verrattuna. Tämä kandidaatintyö on toteutettu yhteistyössä suomalaisen ohjelmistoalan yrityksen kanssa.

1.1 Tutkimusongelma ja -menetelmät

Tämän kandidaatintyön tavoitteena on tutustua erilaisiin matemaattisiin optimointimenetelmiin ja näiden hyödyntämiseen teollisuudessa. Työ pyrkii täten tarjoamaan vastauksen tutkimusongelmaan siitä, miten matemaattisia optimointimenetelmiä voidaan hyödyntää teollisen liiketoiminnan kehittämisessä. Tähän kysymykseen pyritään vastaamaan seuraavien alakysymysten avulla:

- Mitkä ovat tyypillisiä eri teollisuudenaloilla kohdattavia ongelmia, joiden ratkaisemisessa kannattaa hyödyntää matemaattisia optimointimenetelmiä?
- Millä ratkaisualgoritmeilla edellä esiteltyjä optimointiongelmia kannattaisi mahdollisesti lähteä ratkaisemaan?
- Mitä kaupallisesti hyödynnettävissä olevia ohjelmistoja, ohjelmakirjastoja tai työkaluja on olemassa, joita voidaan hyödyntää optimointiongelmiensa ratkaisemisessa?

Työssä käytettyjä tutkimusmenetelmiä ovat kirjallisuusselvitys sekä asiantuntijahaastattelut yhteistyöyrityksessä. Haastatteluilla on kerätty pääasiassa tietoa yrityksen kehityskohteista optimointimenetelmiin perustuvassa teollisessa ongelmanratkaisussa sekä tietoa teollisuudessa yleisesti vastaan tulevista optimointiongelmistä. Näiden haastattelujen pohjalta on lähdetty tutustumaan erilaisiin teollisuuden optimointiongelmatyyppeihin

ja pyritty löytämään näihin ongelmiin soveltuvia ratkaisualgoritmeja. Parhaiten soveltuvien ratkaisualgoritmien löytämisessä on tutustuttu aiheesta löytyvään, pääasiassa tieteelliseen, kirjallisuuteen ja tehty vertailua siellä esiteltyjen ratkaisumenetelmien välillä. Ohjelmakirjastojen ja työkalujen löytämiseksi on pääasiassa tutustuttu verkosta löytyvään materiaaliin Google-hakukoneen avulla, sillä tieteelliset julkaisut aiheesta ovat hyvin vähäisiä.

1.2 Työn rajaukset ja rakenne

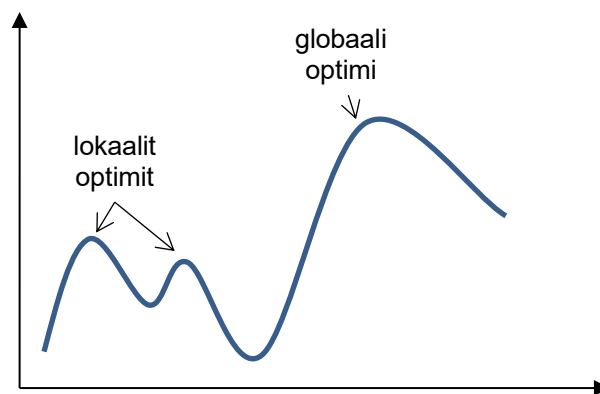
Tässä työssä rajattiin tarkastelu koskemaan erityisesti yhteistyöyrityksen esille tuomia optimointiongelmia teollisuuden parista ja juuri näihin rajattuihin ongelmiin pyrittiin myös löytämään parhaiten soveltuvia ratkaisumenetelmiä. Optimointiin soveltuvien ohjelmistojen ja muiden työkalujen tarkastelussa keskityttiin pääasiassa .NET Core -ympäristöön yhteensopiviin tapauksiin. Lisäksi tuotiin esille myös muita työkaluja ja ohjelmakirjastoja, joista monet olivat yhteensopivia yleisimpien ohjelmointikielien ja kehitysympäristöjen kanssa. Näiden kaikkien ohjelmistojen ja työkalujen kohdalla kuitenkin tarkasteltiin optimointiin yleisellä tasolla soveltuvia tapauksia sen sijaan, että olisi keskitytty vain tarkastelun kohteena olevien ongelmien ratkaisemiseen soveltuviin tapauksiin.

Työ on rakenteeltaan seuraavanlainen. Ensimmäiseksi johdannon jälkeen työssä esitellään yleisesti matemaattisen optimoinnin taustaa ja tutustutaan tarkemmin siihen, miten matemaattista optimointia voidaan hyödyntää erilaisissa käytännön ongelmissa. Seuraavaksi luvussa kolme esitellään teollisuudessa vastaan tulevia tyypillisiä optimointiongelmiä, jotka on jaoteltu teollisuuden eri toimialojen mukaan. Työssä käsitellyt kolme päätoimialaa ovat elintarviketeollisuus, kokoonpanoteollisuus ja puu- ja sahateollisuus. Työssä on tarkoituksena keskittyä erityisesti muutamaan tiettyyn mielenkiintoisimpaan ongelmaan tarkemmin ja esitellä muita ongelmia vain varsin yleisellä tasolla. Neljännessä luvussa nämä tärkeimmät ja mielenkiintoisimmat optimointiongelmat on jaoteltu ongelmatyypin mukaan ja näihin ongelmiin esitellään parhaiten soveltuvia ratkaisualgoritmeja. Tämän jälkeen tutustutaan kohtalaisen yleisellä tasolla optimointiongelmiä ratkaisemiseen soveltuviin ohjelmakirjastoihin ja työkaluihin, jotka ovat hyödynnettävissä kaupallisessa tarkoituksessa. Lopuksi viimeisessä luvussa kuusi esitellään yhteenveto työstä ja käsitellään työstä tehdyt päätelmät.

2. MATEMAATTINEN OPTIMOINTI

Optimoinnista puhuttaessa voidaan tarkoittaa useata eri asiaa. Yhteistä näille kaikille on kuitenkin halutun parhaan mahdollisen vaihtoehdon löytäminen. Monia erilaisia ilmiöitä voidaan kuvailla energia- tai kustannusfunktion perustuvien optimointimallien avulla. Näiden optimointitehtävien ratkaiseminen on varsin keskeinen osa-alue monissa laskennallisen tieteen ja tekniikan tehtävissä. (Haataja 2004, s. 13) Näin on myös teollisuudessa, jonka optimointitehtävien tarkastelemiseen tämä työ keskittyy.

Matemaattinen optimointi on matematiikan osa-alue, jossa etsitään funktiolle $f(\mathbf{x})$ minimi- tai maksimiarvoa riippuen käsiteltävän optimointitehtävän tavoitteista. Kyseinen funktio $f(\mathbf{x})$ on riippuvainen yhdestä tai useammasta muuttujasta x_i , jossa $i = 1, \dots, n$. Tavoitteena on siis löytää sellainen piste \mathbf{x}^* , jossa funktio $f(\mathbf{x}^*)$ saa maksimi- tai minimiarvonsa kaikista mahdollisista $f(\mathbf{x})$:n arvoista. Tätä kyseistä pistettä \mathbf{x}^* kutsutaan funktion $f(\mathbf{x})$ globaaliksi optimiksi. Globaali optimi voi olla joko täysin uniikki piste tai näitä pisteitä voi olla useampia. Vastaavasti funktiolla $f(\mathbf{x})$ on olemassa myös pisteitä \mathbf{x} , joissa funktio saa suurimman arvonsa tämän kyseisen pisteen ympäristössä. Näitä pisteitä kutsutaan lokaaleiksi optimeiksi. Täten voidaankin todeta, että etsittäessä esimerkiksi funktion maksimiarvoa kaikki huiput ovat lokaaleja optimeja mutta ainoastaan korkein huippu on globaali optimi. (Haataja 2004, s. 50–51; Bangert 2012, s. 1) Globaalin ja lokaalin optimin käsitettä on havainnollistettu kuvassa 1.



Kuva 1: Globaali ja lokaali optimi

Yksinkertaisin keino löytää ratkaisu yleisessä tapauksessa matemaattiseen optimointiongelmaan on etsiä annetuista vaihtoehdoista eli käydä läpi jokainen mahdollinen piste annetusta pistejoukosta ja valita tämän jälkeen parhaan tuloksen antanut piste (Bangert 2012, s. 3). Tämän keinon haittana on kuitenkin sen hitaus, joka korostuu entisestään

mitä monimutkaisempi ja laajempi ratkaistava optimointitehtävä on. Tämän vuoksi käytännön sovelluksien ratkaisemiseen käytetään pääasiassa muita menetelmiä, koska ratkaistavat tehtävät ovat usein hyvinkin monimutkaisia erilaisine rajoituksineen ja useine muuttujineen. Näitä soveltuvia ratkaisumenetelmiä ja niiden taustaa käsitellään tässä luvussa hieman myöhemmin.

2.1 Käytännön sovellukset matemaattisessa optimoinnissa

Lähes kaikki käytännön sovellukset matemaattisista optimointiongelmista, joita siis myös tässä työssä käsitellään, ovat joko rajoittuneita tai rajoitettuja. Näitä rajoitteita voivat olla esimerkiksi suurin sallittu maksimikustannus tai jokin fysikaalinen rajoitus, kuten pituus, leveys tai tilavuus (Haataja 2004, s. 57). Näissä tehtävissä kyetään löytämään paras mahdollinen piste tästä kyseisestä rajoitetusta joukosta, muttei sen ulkopuolelta (Bangert 2012, s. 2). Tällöin on mahdollista, että asetettujen rajoitteiden ulkopuolelta löytyisi vielä optimaalisempi ratkaisu kuin niiden sisäpuolelta, mutta sitä ei kyetä löytämään asetettujen rajojen vuoksi.

Käytännön sovellukset ovat usein niin sanottuja monitavoitteisia optimointiongelmia. Näissä on tavoitteena usean eri asian optimointi samanaikaisesti ja usein nämä tavoitteet ovat ristiriitaisia keskenään. Yksinkertaisena esimerkkinä tilanne, jossa tuotteen valmistuskustannukset halutaan minimoida, mutta samalla tuotteen laadun tulisi olla paras mahdollinen. Tämän vuoksi näiden tavoitteiden välille tulee löytää oikeanlainen tasapaino. Ratkaisu tällaisiin tehtäviin on usein kompromissi eri optimointi tavoitteiden välillä. Yhtenä keinona on muuntaa nämä useat erilaiset tavoitteet yhdeksi yhdistelmä tavoitteeksi. Tässä muutoksessa eri muuttujia painotetaan eri tavalla. Tällöin ratkaisun lopputulos riippuu vahvasti siitä, miten tämä muutos on suoritettu eli mitä asioita on painotettu suuremmalla kertoimella kuin toisia. (Yang 2010, s. 17–19, 234–237)

Kokonaislukuoptimoinnissa optimointitehtävän muuttujat saavat arvoikseen ainoastaan kokonaislukuja. Vastaavasti sekalukuoptimoinnissa osa muuttujista on jatkuvia ja osa kokonaislukuja. (Haataja 2004, s. 92) Kokonaisluku tai sekalukuoptimointi voi hyvinkin tulla kyseeseen useissa eri käytännön optimointiongelmassa. Esimerkiksi työntekijöitä ei voi olla 2,3 kappaletta tai valmiita tuotteita ei voida valmistaa 11,7 kappaletta, vaan nämä arvot voivat luonnollisesti olla vain ja ainoastaan kokonaislukuja.

2.2 Kombinatorinen optimointi

Kombinatorinen optimointi on optimoinnin yleisin muoto (Yang 2010, s. 19). Käytännön sovellukset optimoinnista ovat juuri erilaisten rajoitusten ja rajoitteiden vuoksi kombinatorisia optimointitehtäviä. Näissä tehtävissä ratkaisuavaruus on diskreetti eli epäjatkuva. Tämä tarkoittaa sitä, että teoriassa voidaan luetella kaikki mahdolliset ratkaisuvaihtoehdot. Mahdollisia ratkaisuvaihtoehtoja on täten jokin äärellinen määrä. (Haataja 2004, s. 35)

Näiden ratkaisujen määrä kasvaa tyypillisesti kombinatorisessa optimointitehtävässä 2^x tai $x!$ vauhdilla muuttujien lukumäärän x suhteen (Haataja 2004, s. 162). Täten useissa käytännön sovelluksissa, joissa muuttujia x on useita, kasvaa ratkaisuavaruus nopeasti todella suureksi. Haataja (2004, s. 35) toteaaakin, että tämän vuoksi todellisuudessa kaikkia ratkaisuja ei voida käydä läpi, koska se ei ole ajallisesti järkevää.

Kombinatorisen optimointitehtävän ratkaisua voidaan helpottaa saattamalla tehtävän koko mahdollisimman pieneksi. Tämä voidaan toteuttaa esimerkiksi erilaisilla yksinkertaistuksilla sekä muuttujien mahdollisimman tiukalla rajauksella. (Haataja 2004, s. 41) Tästä huolimatta tehtävää ei saa yksinkertaistaa liikaa, jotta se vastaisi edelleen mahdollisimman hyvin todellista optimointitilannetta.

2.3 Optimointitehtävien ratkaiseminen

Joissakin tapauksissa optimointitehtävälle on mahdollista löytää eksakti eli tarkka analyttinen ratkaisu. Näin on kuitenkin varsin harvoin ja sen vuoksi joudutaan turvautumaan numeerisiin menetelmiin. Näin on erityisesti silloin, kun ratkaistavana on useamman muuttujan optimointitehtävä. (Haataja 2004, s. 15)

Haataja (2004, s. 42) tuo myös esille, että optimointitehtävien haasteellisuus muuttuu teknologian kehityksen mukana. Suuren optimointitehtävän määritelmä muuttuu jatkuvasti tietokoneiden laskentatehon parantuessa. Tämä tarkoittaa sitä, että tällä hetkellä mahdollisesti hitaat, mutta hyviä tuloksia tarjoavat ratkaisumenetelmät voivat muuttua tulevaisuudessa ajallisesti kannattavammiksi menetelmiksi teknologian parantuessa.

2.3.1 Eksaktit menetelmät

Lyhyesti tiivistettynä voisi ilmaista, että eksaktit ratkaisumenetelmät antavat täydellisiä vastauksia ilman epävarmuutta (Bangert 2010, s. 5). Näin ei kuitenkaan käytännössä ole, sillä tehtävän lähtötiedot ovat aina rajallisia ja näihin lähtötietoihin liittyy aina epävar-

muutta. Ratkaisua ilman epävarmuutta ei voida saada, jos jo lähtötiedot ovat epävarmoja. Bangert (2012, s. 5) kuitenkin toteaa, että eksaktit menetelmät antavat globaalin optimin sijainnin aina täysin oikein, joten siten mitattuna ratkaisu on aina täydellinen.

Eksakteja ratkaisumenetelmiä ei niiden hyvistä tuloksista huolimatta juurikaan käytetä kuin mahdollisesti hyvin yksinkertaisten optimointitehtävien ratkaisemisessa. Syynä tähän on eksaktien ratkaisumenetelmien käyttämä aika ja tarvitsema laskentateho. Suurten ratkaisujoukkojen vuoksi kombinatoristen optimointitehtävien ratkaisemiseen sovelletaankin usein heuristisia tai likimääräisiä menetelmiä eksaktien ratkaisumenetelmien sijaan (Haataja 2004, s. 162). Näitä menetelmiä käsitellään seuraavaksi.

2.3.2 Heuristiikka ja metaheuristiikka

Suurten ratkaisujoukkojen ja rajallisen ajan sekä laskentatehon vuoksi monimutkaisten optimointitehtävien ratkaisemiseen on kehitetty niin sanottuja heuristisia menetelmiä. Mattila (2018) on kuvannut heuristista ongelmanratkaisua mielestäni varsin hyvin: "Heuristinen ongelmanratkaisu tarkoittaa ratkaisun etsimistä 'valistuneilla' arvauksilla."

Heuristisilla menetelmillä ei siis oikeastaan pyritä edes löytämään globaalia optimia vaan jokin piste riittävän läheltä sitä. Heuristiikka antaakin siis ainoastaan likimääräisiä vastauksia, kun taas eksaktit menetelmät antavat tarkkoja vastauksia (Bangert 2012, s. 5). Usein kuitenkin riittävän hyvä ratkaisu on riittävä, jos se voidaan saavuttaa merkittävästi nopeammin kuin tarkka ratkaisu. Tämä ominaisuus korostuu erityisesti teollisuudessa, jossa esimerkiksi yllättävän konerikon sattuessa tulisi tuotantolinja saada optimoitua uudelleen muutamassa minuutissa 90 % varmuudella optimista sen sijaan että laskentaan käytetään vuorokausi ja saavutetaan 99,9 % varmuudella optimaalinen ratkaisu. Nopeus on siis avainasemassa heurististen menetelmien käytössä ja tämä ominaisuus on tärkeää erityisesti teollisuuden parissa.

Heurististen menetelmien huonona puolena on se, ettei niiden antamista tuloksista aina ole varmuutta. Ne voivat löytää hyvän pisteen, esimerkiksi jonkin lokaalin optimin, mutta eivät onnistu löytämään globaalia optimia. (Bangert 2012, s. 6) Huonona puolena on myös se, ettei heuristisen menetelmän antaman tuloksen optimaalisuutta voida todistaa. Vaikkei saatu tulos olisikaan juuri globaali optimi, sitä ei yleensä voida tietää mitenkään. (Ketonen et al. 2009; Mattila 2018).

Metaheuristiikat ovat sellaisia menetelmiä, jotka yhdistävät korkeamman tason strategioita lähietsinnän kanssa. Metaheuristiikat pyrkivät tutkimaan koko ratkaisuavaruuden jollakin riittävän hyvällä tarkkuudella. Nämä menetelmät pyrkivät hyötymään löydettyjen

hyvin ratkaisujen ominaisuuksista pyrkien kuitenkin samanaikaisesti välttämään jumitumasta samaan tiettyyn lokaaliin optimiin. (Mattila 2018) Monet nykypäivän optimointialgoritmeista ovat juuri erilaisia metaheuristiikoita, sillä ne menestyvät pääsääntöisesti paremmin kuin perinteisemmät heuristiset algoritmit (Yang 2010, s. 21).

3. OPTIMOINTIONGELMAT TEOLLISUUDESSA

Teollisuudessa kohdattavat optimointiongelmat ovat suurimmaksi osaksi jollakin parametreillä rajattuja eli ne ovat kombinatorisia optimointiongelmiä, kuten edellisessä matemaattista optimointia käsittelevässä luvussa on todettu. Hyvin tyypillinen tällainen esimerkki on ongelma, jossa on tiedossa annetut tehtävät ja ainoastaan tehtävien järjestyksestä saa muutella optimin saavuttamiseksi (Jännes, Kanninen, Rovio, Salomäki, Toivonen, haastattelu 7.2.2020).

Ketonen et al. (2009) on todennut hyvin, että ennen kuin lähdetään optimoimaan yhtä tiettyä toimintoa, tulisi varmistua siitä, että toiminta on kokonaisuutena sekä vakiintunutta että taloudellisesti tehokasta. Useimmiten suurimmat toiminnan tehostamismahdollisuudet on mahdollista löytää tarkastelemalla yleisesti koko toiminnan tasoa sen sijaan, että tartuttaisiin johonkin yksittäiseen pieneen toimintoon ja pyrittäisiin optimoimaan sen toimintaa. Tämän lisäksi Ketonen et al. (2009) toteavat, että yksityiskohtainen optimointi vaatii aina myös mahdollisimman kattavat lähtötiedot. Näiden asioiden vuoksi tulisikin ennen yksittäisen toiminnon optimointia pohtia, onko koko muu toiminta niin tehokasta kuin se voisi olla. Olisiko täten toimintaa mahdollista tehostaa joillakin muilla keinoilla, kuin yhden pienen yksittäisen toiminnan optimoinnilla?

Tässä luvussa esitellään erilaisia teollisuudessa vastaan tulevia ongelmia, joiden ratkaisemisessa voidaan hyödyntää matemaattisia optimointimenetelmiä. Optimointiongelmat on jaoteltu ensin yleisen tason ongelmiin, jotka koskettavat monia eri teollisuudenaloja ja nämä ongelmat esitellään hyvin yleisellä tasolla. Tämän jälkeen keskitytään tarkastelemaan seuraavia ongelmatyyppejä eri teollisuudenalojen näkökulmasta:

- Annetuilla työtehtävillä tapahtuva asetusaikojen minimointi.
- Työntöohjatun raaka-ainekäytön optimointi.
- Tuotantosuunnitelman optimointi.
- Resurssikäytön optimointi.
- Kaksiulotteinen leikkausongelma.

Kyzeisiä optimointiongelmiä keskitytään tarkastelemaan jonkin tietyn teollisuudenalan näkökulmasta. Tämä ei kuitenkaan tarkoita sitä, etteikö kyzeisiä ongelmia esiintyisi myös muilla teollisuudenaloilla. Tässä esitellyt ongelmat ovat pääasiassa vain esimerkkejä tietyistä optimointiongelmatyypistä ja ne ovat helposti yleistettävissä koskemaan myös muita teollisuudenaloja.

3.1 Yleisiä optimointiongelmia

On olemassa runsaasti erilaisia optimointiongelmia, jotka ovat merkityksellisiä varsin monilla eri teollisuudenaloilla. Tässä kappaleessa esitellään hyvin yleisellä tasolla näistä muutamia merkittävimpiä ongelmia ja näihin ongelmiin ei ole tarkoitus tutustua enää tämän tarkemmin tämän työn aikana.

Ensimmäisenä mainittakoon erilaiset aikataulutuksen ja sen suunnitteluun liittyvät ongelmat. Nämä ovat hyvin laaja ja monimutkainen perhe kombinatorisessa optimoinnissa. Näihin on olemassa täten runsaasti tosielämän sovelluksia, kuten erilaiset ennakoivan huollon suunnitteluun liittyvät ongelmat. (Bartz-Beielstein et al. 2018)

Yhtenä merkittävänä optimointiongelmana ovat myös erilaiset logistiikkaan ja kuljetukseen liittyvät ongelmat. Hyvin tyypillisenä esimerkkinä tästä on kauppamatkustajanongelma eli englanninkielisessä kirjallisuudessa lyhenteeltään TSP (*travelling salesman problem*). Kyseisestä ongelmasta on olemassa useita eri variaatioita, ja ne ovat helposti yleistettävissä erilaisiin logistiikan ongelmiin. Kauppamatkustajanongelmassa kauppiaan tulee vierailia x määrässä kaupunkeja, siten että jokaisessa kaupungissa saa käydä vain kerran ja kuljetun reitin tulee olla mahdollisimman lyhyt.

Teollisuudessa varastointi on hyvin merkittävä asia, ja tämän vuoksi varastonhallinta on erkaantunut muusta tuotannosuunnittelusta täysin omaksi alueekseen. Varastonhallinnassa pyritään löytämään optimi kolmen pääkustannustyyppin välillä. Nämä kustannustyytit ovat tilauskustannus, hallussapitokustannus ja mahdollisen puutteen kustannus. Näiden perusteella pyritään optimoimaan milloin varastoa tulisi täydentää ja kuinka paljon. (Paschos 2014, s. 79)

3.2 Elintarviketeollisuus

Elintarviketeollisuus on teollisuudenala, joka tuottaa erilaisia elintarvikkeita tai niiden raaka-aineita. Työ- ja elintarvikeministeriö (2019, s. 11-12) on todennut julkaisussaan elintarviketeollisuuden olevan yksi suurimpia teollisuudenaloja Suomessa tuotannon arvolla mitattuna, joten sitä voidaan pitää varsin merkittävänä teollisuudenalana. Elintarviketeollisuus jaetaan kahteen päätoimialaan, jotka ovat elintarvikkeiden valmistus ja juomien valmistus (Työ- ja elintarvikeministeriö 2019, s. 11-12).

3.2.1 Juomateollisuuden pullotusongelma

Ensimmäisenä tarkemmin käsiteltävänä optimointiongelma esitellään juomateollisuuden parissa esiintyvä pullotusongelma. Tämä on hyvä esimerkki annetuilla työtehtävillä tapahtuvasta vaihtoaikojen minimoinnista (Jännes et al. 2020). Ongelmaa tarkastellaan aluksi hieman laajemmasta näkökulmasta ja tämän jälkeen keskitytään itse pullotuksen optimointiin.

Elintarviketeollisuudessa yksi merkittävä optimointiongelma on tuotantomäärien ja aikataulutuksen suunnittelun optimointi panimoteollisuudessa. Tämä voidaan vielä jakaa kahteen eri päätasoon eli nesteen valmistukseen ja tämän jälkeiseen valmiin nesteen pullotukseen. Nämä kaksi samaa päätasoa löytyvät myös muusta juomateollisuudesta kuten virvoitusjuomateollisuudesta. Merkittävimpana erona kuitenkin panimoteollisuuden ja muun juomateollisuuden välillä on nesteen valmistukseen kuluva aika. Panimoteollisuudessa nesteen käyminen on aikaa vievä prosessi, jolloin ainoastaan odotetaan kyseisen nesteen valmistumista. Nesteen valmistusta joudutaankin helposti odottamaan päiviä tai useimmiten jopa viikkoja, kun taas muussa juomateollisuudessa nesteen valmistumiseen kuluu aikaa korkeintaan muutamia tunteja. (Almada-Lobo et al. 2014) Tämän vuoksi panimoteollisuuden tuotannosuunnittelussa tulee huomioida erityisesti nesteen valmistuksen aloitusajankohta, jotta tilattu tuotantoerä saadaan toimitettua luvatussa aikataulussa.

Toinen päävaihe eli pullotus taas on helposti yleistettävissä koskemaan mitä tahansa juomateollisuutta. Pullotusvaiheessa optimoinnin tavoitteena on minimoida vaihtoajat eri tuotantoerien välillä. Tässä vaiheessa on kaksi muuttujaa, jotka voivat vaihtua. Nämä ovat pullotettava tai yleisemmin ilmaistuna pakattava neste ja pakkauskoko. Yhdellä linjastolla voidaan pakata ainoastaan yhtä nestettä yhdellä tietyllä ajanhetkellä. Täten nesteen tai pakkauksen tai molempien vaihtuessa linjastolla tulee suorittaa erilaisia asetus- ja pesutoimenpiteitä. Nämä toimenpiteet voivat olla joko hieman pienempiä ja lyhyempiä tai hieman suurempia, esimerkiksi nesteen vaihtuessa tummemmasta kirkkaampaan tulee linjaston pesu toteuttaa huolellisemmin kuin jos neste vaihtuisi kirkkaasta tummempaan. (Almada-Lobo et al. 2014; Jännes et al. 2020) Nämä toimenpiteet ovat aikaa vieviä ja asiakkaalle arvoa tuottamattomia toimenpiteitä, joten niistä aiheutuu ainoastaan turhia kustannuksia. Optimoinnin tuloksena tulisivin saada tieto siitä, milloin vaihdetaan nestettä ja milloin taas pakkaustyyppiä, jotta asetusajat saadaan minimoitua (Jännes et al. 2020).

Tarkasteltaessa taas koko tuotantoketjua panimo- tai juomateollisuudessa tulee optimointia suoritettaessa huomioida erilaiset linjastot ja koneet sekä niiden välillä olevat

eroavaisuudet. Jollakin linjalla kyetään mahdollisesti pakkamaan ainoastaan tiettyjä pakkaustyyppisiä, kun taas toisella linjalla voidaan pakata kaikkia pakkausvaihtoehtoja. Yksinkertaistetussa tilanteessa voidaankin tarkastella vain yhtä linjaa ja pyrkiä optimoimaan sen tuotantojärjestystä, mutta todellisuudessa linjoja on useita ja niissä eroavaisuuksia (Jännes et al. 2020). Lisäksi tulisi huomioida eri tuotantovaiheiden välinen yhteensovittaminen. Ideaalitulanteessa välittömästi edellisen pullotuksen päätyttyä päästäisiin pullottamaan uutta juuri valmistunutta nestettä. Todellisuudessa kuitenkin joko neste joutuu odottamaan pullotukseen pääsyä tai pullotus joutuu odottamaan nesteen valmistumista. Lisäksi erilaiset yllättävät tilanteet aiheuttavat tarpeen uudelleen suunnittelulle kesken tuotannon. Näitä tilanteita voisivat olla esimerkiksi yllättävät laiterikot tai raaka-aineiden saantivaikeudet. (Almada-Lobo et al. 2014)

3.2.2 Lihan leikkaamisen optimointi

Seuraavaksi tarkastellaan elintarviketeollisuuden toista päätoimialaa eli elintarvikkeiden valmistusta ja erästä siellä kohdattavaa optimointiongelmia. Tämä optimointiongelma liittyy raaka-aineen käytön katteen optimointiin ja tässä tarkastelun kohteena on lihan leikkaamisen optimointi. Tämä voidaan nähdä yhtenä esimerkkinä työntöohjatun raaka-ainekäytön optimoinnista.

Eläin tulee teurastaa sen saavuttaessa tietyn pisteen, jolloin raaka-ainetta saapuu elintarviketeollisuuden käyttöön työntöohjauksella. Työntöohjauksessa raaka-ainetta saapuu tietyllä ennalta määrättyllä tahdilla riippumatta varsinaisesti sen tarpeesta. Tästä raaka-aineesta tulisi kyetä hyödyntämään niin paljon kuin mahdollista ja lisäksi pyrkiä maksimoimaan saatu tuotto. Jokaisella eläimellä on tietty niin sanottu primäärituote, joka voi olla esimerkiksi naudan sisäfilee tai vaikkapa broilerin rintafilee. Tätä samaa primäärituotetta voidaan saada usealla erilaisella leikkaustavalla primäärituotteen määrän hie-man vaihdellessa. Se, johon erilaiset leikkaustavat merkittävimmin vaikuttavat, ovat niin sanotut sekundäärituotteet eli primäärituotteen lisäksi saatavat muut lihatuotteet. (Jännes et al. 2020) Lisäksi erilaiset leikkuutavat voivat viedä aikaa eri tavalla, jolloin eri leikkauksista aiheutuu erilainen kustannus (Albornoz et al. 2015).

Lihan leikkaamisen optimoinnissa eli lihan leikkuutavan valinnassa tuleekin huomioida eri tuotteiden sen hetkiset kysynät, sillä huonoin vaihtoehto on mahdollinen hukka eli käytännössä lihan pilaantuminen. Primäärituotteillakin voi olla erilaisia muotoja kuten luullinen tai luuton versio broilerin rintafileestä. Primäärituotteiden ja sekundäärituotteiden lisäksi liha on mahdollista jauhaa jauhelihaiksi. Tällöin hukkaa minimoidaan, mutta samalla menetetään mahdollisuus saada parempaa tuottoa hyväkatteisen primäärituot-

teen myynnistä. (Jännes et al. 2020) Lihan leikkuun optimointi onkin siis eläimestä saatavan tuoton optimointia samalla pyrkien vastaamaan annettuun tai mahdollisesti arvioituun kysyntään.

3.3 Kokoonpanoteollisuus

Kokoonpanoteollisuus on teollisuudenala, jossa erilaisista komponenteista nimensä mukaisesti kootaan joko valmiita lopputuotteita tai osakokoonpanoja muihin lopputuotteisiin. Kokoonpanoteollisuus on hyvin laaja teollisuudenala, joten sinne mahtuu runsaasti erilaisia ja erityyppisiä optimointiongelmia. Lisäksi globalisaation lisääntyessä on kilpailu kokoonpanoteollisuudessa kasvanut, minkä seurauksena tuotannon tehokkuus tulee kyetä maksimoimaan pitäen kustannukset kuitenkin mahdollisimman matalalla (Hutabarat et al. 2012).

3.3.1 Kokoonpanotuotannon optimointi

Kokoonpanoteollisuuden parissa yhtenä merkittävänä optimointiongelmanä voidaan nähdä varsinaisen kokoonpanotuotannon optimointi eli kokoonpanotuotannon tuotannosuunnittelu. Tämä ongelma voidaan nähdä yhtenä esimerkkinä tuotantosuunnitelman optimoinnista.

Tällä alueella on kaksi hyvin merkittävää ja optimoitavissa olevaa ongelmaa. Ensimmäinen on nimeltään kokoonpanojärjestyksensuunnittelu, josta käytetään kirjallisuudessa vakiintunutta lyhennettä ASP (*assembly sequence planning*) (Lu et al. 2013). Tämän tavoitteena on saada määritettyä tuotteen optimaalinen kokoonpanojärjestys siten, että kokoonpano saadaan suoritettua mahdollisimman tehokkaasti. Tällä on myös merkitystä yleiseen tuottavuuteen, sillä se määrittää myös sen, kuinka nopeasti ja tarkasti kyseinen tuote on kokoonpantu. (Hutabarat et al. 2012) Yhtenä merkittävänä osana tämän kokoonpanojärjestyksensuunnittelua on jo edellisessä luvussa juomateollisuuden yhteydessä esitelty asetusaikojen optimointi. Erilaisen tuotteen vaihtuessa tulee koneita useimmiten säätää, jolloin kuluu aikaa ja tämä aika on asiakkaalle arvoa tuottamatonta. (Jännes et al. 2020) Yhtenä merkittävänä tavoitteena on täten ASP:ssä asetusaikojen minimointi.

Toinen merkittävä optimointiongelma kokoonpanoteollisuuden tuotannosuunnittelussa on kokoonpanolinjantasapainotus eli ALB (*assembly line balancing*) (Lu et al. 2013). Tässä tavoitteena on jakaa työtehtäviä siten, että työmäärät eri työryhmillä ovat täysin samat tai hyvin lähellä sitä (Hutabarat et al. 2012). Tämä ongelma liittyy hyvin vahvasti myös erilaisten resurssin käyttöön, jota tarkastellaan lisää hieman myöhemmin.

Lisäksi tulee huomioida se, että vaikka tässä ASP ja ALB onkin käsitelty toisistaan erillisinä kokonaisuuksina, todellisuudessa ne liittyvät hyvin vahvasti toisiinsa. Optimoinnissakin nämä kaksi tulisi mielellään käsitellä yhtenä kokonaisuutena, sillä jos ensin pyritään optimoimaan ASP ja sitten vasta ALB, niin niiden antamat suositukset ovat helposti ristiriitaisia keskenään (Lu et al. 2013).

3.3.2 Resurssikäytön optimointi

Kokoonpanoteollisuudessa tulee huomioida erilaiset resurssit, ja se missä niitä milloinkin käytetään. Tämä on yksi merkittävä osa tuotannosuunnittelua kokoonpanoteollisuudessa ja liittyy vahvasti myös edellä esitettyihin ongelmiin. Kyseistä ongelmaa on tarkasteltu kirjallisuudessa myös erikseen erityisesti palvelualoilla. Kyseinen ongelma voidaan nähdä yhtenä esimerkkinä resurssikäytön optimoinnista.

Tämän ongelman kohdalla tulee miettiä sitä, minne tulee lisätä kapasiteettia ja kuinka paljon sitä tulee lisätä, jotta ennalta määritelty tuotanto saadaan tehtyä aikataulussa valmiiksi (Jännes et al. 2020). Harvoin kapasiteetin lisääminen vain yhteen hitaimpaan prosessiin eli pullonkaulaan riittää, sillä seuraava pullonkaula syntyy välittömästi tämän jälkeen jonnekin muualle. Yleisimmät pullonkauloja aiheuttavat ja täten kapasiteetin lisäys-tarvetta aiheuttavat resurssit ovat henkilöstö tai koneet. Useimmiten ne nämä molemmat yhdessä samanaikaisesti. Nämä samat resurssit aiheuttavat myös rajoituksia kapasiteetin lisäämisessä. Kapasiteettiakaan ei siis kyetä lisäämään rajattomasti. Esimerkkinä tästä toimii kokoonpanotuotannossa jokaisen tuotteen kohdalla tarvittava hyvin kallis kone, joita on vain muutama kappale tehtaalla. Tällöin jos näiden koneiden käyttöaste on jo maksimissaan, ei sitä voida enää nostaa enempää ja täten kapasiteettia ei kyetä siltä osin lisäämään.

Teoriassa kapasiteettia voitaisiin lisätä kaikille työpisteille siten, että se varmasti riittää mutta tällöin aiheutuu tarpeettomia kustannuksia, kun osan ajastaan työntekijät ovat täysin toimeettomina. Tämän vuoksi optimointi tässäkin asiassa olisi tärkeää, jotta toiminta saataisiin mahdollisimman tehokkaaksi. Jottei tämä tilanne olisi liian yksinkertainen, tuovat oman haasteensa erilaiset jaettavissa olevat resurssit. Tällöin esimerkiksi sama työntekijä voi käyttää kahta konetta samanaikaisesti ja optimoinnin yhteydessä tulee pohtia missä suhteessa tämä käytettävissä oleva aika tulee jakaa. (Jännes et al. 2020)

3.4 Puu- ja sahateollisuus

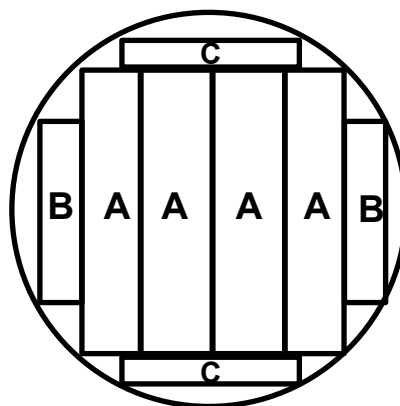
Puu- ja sahateollisuus ovat yksi merkittävimpiä teollisuudenaloja Suomen viennissä. Pelkkä sahatavara oli Suomen viidenneksi tärkein vientitavara vuonna 2019 (Sahateollisuus ry 2020). Täten pärjätäkseen globaaleilla markkinoilla tulee suomalaisten sahajen toiminnan olla tehokasta ja tuottavaa. Tuottavuuden parannusta voidaan saavuttaa hyvin tehdyllä tuotannon optimoinnilla.

3.4.1 Tukin sahauksen optimointi

Yhtenä merkittävimpana optimoinnin kohteena sahateollisuudessa voidaan pitää tukin sahauksen optimointia. Puutavara saapuu samalla periaatteella sahalle kuin liha saapui teurastamolle eli työntönä. Täten tämä ongelma toimii toisena esimerkkinä työntöohjatun raaka-ainekäytön optimoinnista.

Koska raaka-ainekustannukset ovat merkittävin osa sahan tuotantokustannuksista, on täten raaka-aineen käytön optimointi tärkeää (Ketonen et al. 2009). Tällöin annetusta raaka-aineesta tulee saada aikaiseksi niin paljon tuottoa kuin on mahdollista. Tähän kyseiseen optimointiongelmaan pätee pääsääntöisesti täysin samat asiat ja tavoitteet kuin lihanleikkuun tapauksessa, mutta tarkastellaan ongelmaa vielä erikseen juuri sahateollisuuden näkökulmasta.

Sahalle saapuvat tukit mitataan ja ne lajitellaan koneellisesti erilaisiin ennalta määritettyihin luokkiin, joita on tyypillisesti joitakin kymmeniä. Kun tiettyyn luokkaan on saatu vähimmäismäärä tukkeja, ne voidaan sahata yhtenä sahauseränä kaikki samalla sahausasetteella. (Ketonen et al. 2009) Kuvassa 2 on havainnollistava päätykuva tukista, jossa on esitettyä mitä sahausasete tarkoittaa.



Kuva 2: Havainnollistava kuva sahausasetteesta

Kuten kuvasta 2 havaitaan, saadaan siis yhdestä tukista aina yhtä tai useampaa erilaista puutavaraa ja ylijäävä osuus tukista pyritään minimoimaan. Tämä tarkoittaa käytännössä

sitä, että joissakin tapauksessa tiettyyn luokkaan joudutaan, esimerkiksi kysynnän täyttämiseksi, sijoittamaan joko liian pieniä tai liian suuria tukkeja. (Ketonen et al. 2009)

Pääasiallisena tavoitteena on siis maksimoida tukista saatava myyntituotto ja samalla täyttää asiakastilaukset (Ketonen et al. 2009). Sahausasetteen määrityksessä tuleekin siis ennen kaikkea huomioida annettu ja ennustettu kysyntä. Ongelmia voivat kuitenkin aiheuttaa kysynnän täyttämiseksi sivussa syntyvät puutavaradimensiot eli sivutuotteet, joille ei olekaan kysyntää siinä määrin kuin niitä sahauksen seurauksena syntyy. Tämä puutavara aiheuttaa helposti ongelmia myöhemmissä tuotantoketjunvaiheissa. Merkittävimpiä ongelmia aiheutuu kuivaamossa ja loppuvarastoinnissa, joihin mahtuu ainoastaan rajallinen määrä puutavaraa kerralla. Jos nämä sivutuotteet joutuvat esimerkiksi odottamaan kuivaamoon pääsyä, ne saattavat vahingoittua sillä aikaa ja niistä saatava kate heikkenee. Vastaavasti ne saattavat viedä runsaasti tilaa varastosta, jonne pitäisi saada mahtumaan kysytyintä puutavaraa odottamaan kuivauksen ja toimituksen välillä. (Jännes et al. 2020)

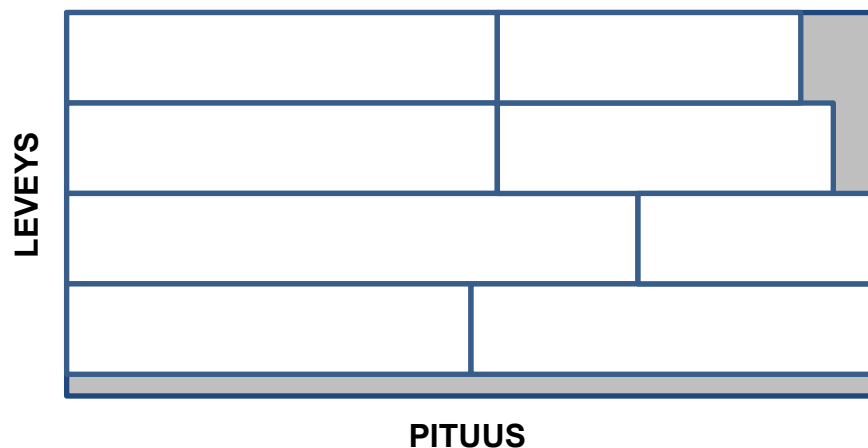
Kuten lihan tapauksessa, niin myös tukin tapauksessa on mahdollista pilkkoa koko tukki kokonaisuudessaan pienemmän katteen tuotteeksi. Tukin tapauksessa tämä tarkoittaa puun hakkaamista hakkeeksi. Tällöin tukista saatava kate on pienempi kuin mahdollisena lautana, mutta mahdollinen hukka on minimoitu. (Jännes et al. 2020) Täten tukin sahauksen optimoinnissa eli sahausasetteen valinnassa tuleekin huomioida pelkän sahauksen hukan minimoimisen lisäksi myös tuotantoketjun myöhemmät vaiheet.

3.4.2 Ladontakuvion optimointi

Toinen mielenkiintoinen sahateollisuudessa esiintyvä optimointiongelma on viilupuu eli LVL (*laminated veneer lumber*) aihion sahauksen optimointi. Tämä voidaan nähdä eräänlaisena esimerkkinä kaksiulotteisesta leikkausongelmasta.

Tässä tavoitteena on minimoida valmiista aihioista asiakastilauksien mukaisia tuotteita sahatessa syntyvä hukka eli levyaihiosta ylijäävä osuus. (Jännes et al. 2020) Tärkeimpänä tavoitteena onkin täten annetun kysynnän täyttäminen ja samalla pyrkiä minimoimaan aihion hukka. Tämä ongelma onkin helposti yleistettävissä esimerkiksi metalliteollisuuden metallilevyjen leikkaamiseen liittyen tai tekstiiliteollisuuden kankaiden leikkaamiseen liittyen.

Haasteena kuitenkin LVL:n tapauksessa on se, että ahiolla on kaksi dimensiota, jotka ovat ennalta määrättyjä. Nämä ovat aihion paksuus, joka riippuu asiakastilauksista sekä aihion leveys, joka määräytyy koneen mukaan jolla aihio tehdään. Aihion pituuteen voidaan kuitenkin vaikuttaa, sen ollessa jotakin tietyltä annetulta väliltä esimerkiksi 6–24 metriä. (Jännes et al. 2020) Tämä mahdollisuus vaikuttaa aihion pituuteen on merkittävin erottava tekijä muista vastaavista kaksiulotteisista leikkausongelmista. Tämän seurauksena optimoitavana onkin pelkän ladontakuvion lisäksi myös aihion pituus. Yleisemmissä tapauksissa taas aihion kaikki dimensiot ovat tiedossa ja optimoinnin tavoitteena on vain asetella leikattavat kohteet tälle ahiolle siten, että hukkan osuus minimoidaan. Tätä ladontakuvion asettelua on havainnollistettu kuvassa 3.



Kuva 3: Karkea havainnekuva ladontakuvioista, jossa harmaa alue on hukkaa

Lisähaasteen tuovat myös LVL-aihion leikkaamiseen käytetyt koneet, sillä osa näistä koneista sahaa levyn ensin pituussuunnassa ja sitten vasta leveyssuunnassa. Toiset koneet taas tekevät juuri päinvastoin eli ensin aihiota sahataan leveyssuunnassa ja sitten vasta pituussuunnassa. (Jännes et al. 2020) Tällaisesta leikkaustavasta, jossa aihio sahataan reunasta reunaan ensin joko pituus- tai leveyssuunnassa, käytetään nimitystä giljotiinileikkaus (*guillotine cut*). Täten myös tämä asia tulisi kyetä huomioimaan aihion pituutta ja ladontakuvioita optimoitaessa.

4. RATKAISUALGORITMIT TEOLLISUUDEN OPTIMOINTIONGELMIIN

Optimointiongelmien ratkaisemiseksi on kehitetty ja kehitetään jatkuvasti uusia ratkaisualgoritmeja. Yksi tunnetuimmista ja hyvin moniin erilaisiin optimointiongelmiin sovellettu algoritmi on geneettinen algoritmi, josta käytetään englanninkielisessä kirjallisuudessa lyhennettä GA (*genetic algorithm*). Tämä on varsin yleispätevä algoritmi, joka oikein sovellettuna kykenee ratkaisemaan lähes minkä tahansa optimointiongelman (Bangert 2012, s.6). Toinen yleispätevä optimointialgoritmi on simuloitu jäähdytys, josta kirjallisuudessa käytetään lyhennettä SA (*simulated annealing*). Bangert (2012, s. 7) toteaa SA:n kykenevän ratkaisemaan lähes minkä tahansa optimointiongelman ja toteaakin sen olevan riittävän hyvä ratkaisemaan teollisuudessa vastaan tulevat tapaukset. Lisäksi Yang (2010, s. 241) toteaa simuloitun jäähdytyksen sekä hiukkasparvioptimoinnin olevan hyvin lupaavia menetelmiä monitavoitteisten optimointiongelmien ratkaisemisessa. Täten ainakin simuloitua jäähdytystä voitaisiin pitää hyvänä lähtökohtana uuden optimointiongelman ratkaisemisessa, jos mitään muuta lähtökohtaa ratkaisumenetelmän valinnalle ei ole.

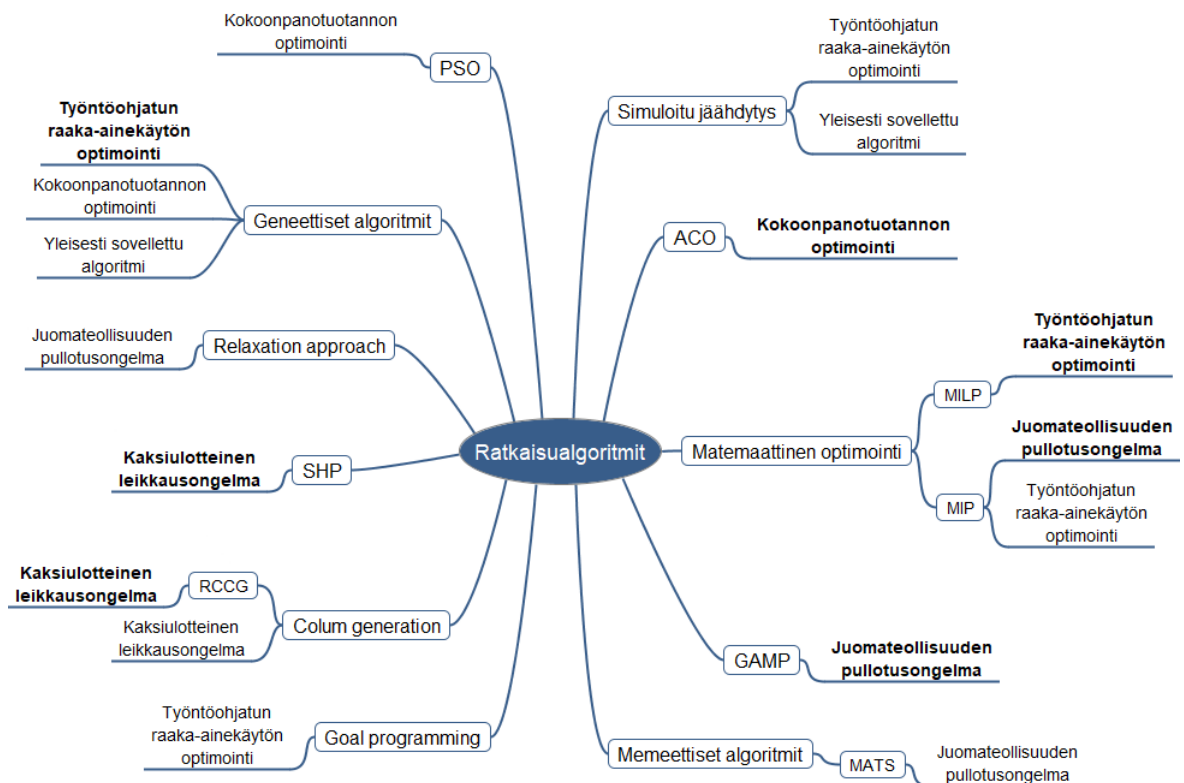
Haataja (2004, s. 32) taas tuo esille sen, ettei ole olemassa yhtä yleispätevää ratkaisumenetelmää vaan eri optimointitehtäville tulisivatkin käyttää erilaisia ratkaisualgoritmeja. Jokaiselle optimointialgoritmille voidaan kehittää sellainen tehtävä, jota kyseinen algoritmi ei kykene ratkaisemaan tehokkaasti (Haataja 2004, s. 38). Täten optimointialgoritminkin valinta on yksi optimointiongelma. Tähän valintaan vaikuttavia tekijöitä ovat muun muassa ongelman tyyppi, käytettävissä oleva laskentateho, käytettävä aika ja halutun ratkaisun laatu (Yang 2010, s. 243). Geneettiset algoritmit ovat monissa tapauksissa kyenneet ratkaisemaan annetun ongelman, mutta ratkaisun laatu ja nopeus ovat useimmiten olleet selvästi heikompia kuin muilla käytetyillä algoritmeilla.

4.1 Ratkaisualgoritmit teollisuudessa

Jännes et al. (2020) toivat esiin ratkaisun saamisen nopeuden olevan ehdottomasti yksi tärkeimpiä asioita teollisuuden optimointiongelmien parissa. Teollisuudessa on tärkeää kyetä reagoimaan nopeasti muuttuneisiin tilanteisiin ja saada esimerkiksi konerikon seurauksena uusi optimoitu tuotantosuunnitelma mahdollisimman nopeasti. Täten algoritmin ratkaisunopeuteen on kiinnitetty erityistä huomiota ratkaisualgoritmien valinnassa, jos kyseinen tieto on ollut saatavilla.

Muita tarkastelun kohteita eri ratkaisualgoritmeja vertailtaessa ovat olleet ratkaisun laatu sekä ratkaisun uutuus. Kuten aiemmin on todettu, teknologian kehityksen myötä myös ratkaisualgoritmit ja niiden käytettävyys kehittyvät. Kaikkein uusimpia ja harvinaisempia algoritmeja ei kuitenkaan tarkastelussa ole huomioitu, sillä niistä saattaa löytyä ainoastaan yksi tai kaksi tutkimusta. Tällöin niiden sovellettavuutta muihin ongelmiin kuin juuri kyseisessä tutkimuksessa käsiteltyyn on hyvin hankalaa arvioida.

Seuraavaksi esitellyt algoritmit suositukset perustuvat siihen käsitykseen, joka läpikäydyn kirjallisuuden pohjalta on muodostunut. Lisäksi optimointiongelmien lähtötietoihin liittyy aina oletuksia ja epävarmuuksia, ja monissa tapauksissa esitetyt ratkaisumenetelmät ovat heuristisia eli ne tarjoavat ainoastaan riittävän hyviä ratkaisuja. Täten voidaankin olettaa myös näiden suositusalgoitmien olevan aivan riittävän hyviä kyseisten ongelmien ratkaisemiseen. Työssä käsitellyt ratkaisualgoritmit ovat esitettynä kuvassa 4.



Kuva 4: Työssä käsitellyt ratkaisualgoritmit

Kuvaan 4 on koottuna työssä käsiteltyjä ratkaisualgoritmeja. Ratkaisualgoritmin perässä lihavoidulla olevat optimointiongelmat ovat niitä ongelmia, joiden ratkaisemiseen tässä työssä suositellaan käytettävän kyseistä ratkaisualgoritmia. Yleisesti sovellettu algoritmi tarkoittaa, että kyseistä ratkaisumenetelmää voi hyödyntää lähes minkä tahansa optimointiongelman ratkaisemiseen.

4.2 Kokoonpanotuotannon optimointi

Monimutkaisuutensa vuoksi kokoonpanojärjestyksensuunnittelu ja kokoonpanolinjantasapainotus ongelmia ei kyetä ratkaisemaan eksaktisti järkevissä ajoissa edes kaikkein tehokkaimmilla tietokoneilla (Hutabarat et al. 2012). Tämän vuoksi ratkaisussa tuleekin käyttää heuristisia tai metaheuristisia algoritmeja. Tässä luvussa esitellään kirjallisuuden pohjalta parhaiten ASP:n ja ALB:n tyyppisten optimointiongelmiä ratkaisemiseen soveltuvia algoritmeja.

Käytetyimpiä menetelmiä näiden ongelmien ratkaisemiseen ovat geneettiset algoritmit, muurahaisyhdyskunta algoritmit eli ACO (*ant colony algorithms*) sekä hiukkasparvioptimointi eli PSO (*particle swarm optimization*) (Hutabarat et al. 2012). Lisäksi käytössä on myös useita muita algoritmeja, mutta näitä on käytetty ja täten tutkittu merkittävästi vähemmän, joten niitä ei ole huomioitu tässä tapauksessa.

Kokoonpanojärjestyksensuunnittelun ja kokoonpanolinjantasapainotuksen ongelmat ratkaistaan tyypillisesti peräkkäin eli ensin ratkaistaan näistä toinen ja sitten vasta toinen. Parempi tapa olisi kuitenkin ratkaista nämä kaksi ongelmaa rinnakkain, koska muuten näiden kahden ongelman ratkaisujen välille muodostuu helposti ristiriitoja. Geneettisten algoritmien on todistettu kykenevän ratkaisemaan yhdistetty ASP ja ALB ongelma saaden hyviä tuloksia. Toisaalta on myös todistettu, etteivät geneettiset algoritmit pysty ratkaisemaan optimia tai edes löytämään toteuttamiskelpoista ratkaisua silloin, kun ASP ongelma on riittävän monimutkainen. (Lu et al. 2013; Bahubalendruni et al. 2018) Tämä riittävä monimutkaisuus tulee helposti vastaan ongelmassa, jotka ovat hyvin lähellä oikeita todellisia tilanteita. Lisäksi Hutabarat et al. (2012) toteaa, ettei geneettisiä algoritmeja useinkaan voida käyttää suoraan tämän tyyppisten tehtävien ratkaisemiseen.

Myöskään hiukkasparvioptimointi ei ollut sovellettavissa suoraan tämän tyyppiin ongelmiin ja lisäksi sillä oli taipumuksena jumiutua ratkaisussa lokaaliin optimiin (Hutabarat et al. 2012; Bahubalendruni et al. 2018). Täten näistä kolmesta dominoivasta ratkaisumenetelmästä jäljelle jäi ainoastaan muurahaisyhdyskunta algoritmit, jotka soveltuvat suoraan ASP:n ja ALB:n ratkaisemiseen (Hutabarat et al. 2012; Bahubalendruni et al. 2018). Muurahaisalgoritmit ovat toimineet tehokkaasti myös aiemmin mainitun yhdistetyn ASP:n ja ALB:n ratkaisemisessa. Lu et al. (2013) ovatkin käyttäneet ASP:n ja ALB:n ratkaisemisessa rinnakkain erästä versiota muurahaisyhdyskunta algoritmeista, joka kykenee sovittamaan mahdolliset ristiriidat näiden kahden eri optimointitehtävän tavoitteiden välillä. Tämä ratkaisu kykeni löytämään optimaalisen ratkaisun myös monimutkai-

sempiin tehtäviin (Lu et al. 2013). Ongelmana ACO algoritmeissa on kuitenkin riski enenaikaiseen suppenemiseen ja täten ratkaisu voi jäädä lokaaliin optimiin (Bahubalendruni et al. 2018).

Tämän perusteella kokoonpanojärjestyksensuunnittelun ja kokoonpanolinjantasapainoituksen optimointiin voidaan suositella käytettävän muurahaisyhdyskunta algoritmeja. Tilanteessa, jossa epäillä ratkaisun jäävän liian heikoksi voisi kokeilla hybridiratkaisua, jossa muurahaisalgoritmit ovat yhdistettynä toiseen algoritmiin. Nämä hybridialgoritmit ovat rakenteeltaan monimutkaisempia, joten niiden toteutukseen kuluu enemmän aikaa ja lisäksi monimutkaisuutensa vuoksi ne käyttävät enemmän laskenta-aikaa. Näiden asioiden vuoksi ne eivät ole ensisijainen suosituksen kohde. Tutkimuksissa käytettyjä ja menestyneitä hybridialgoritmeja ovat ACO:n ja GA:n yhdistelmä sekä ACO:n ja PSO:n yhdistelmä (Bahubalendruni et al. 2018). Nämä vaihtoehdot vaikuttavat tämänhetkisen tiedon perusteella kykenevän varsin hyvin ratkaisuihin tämän tyyppisten optimointiongelmiä ratkaisemisessa.

4.3 Juomateollisuuden pullotusongelma

Pelkkä juomateollisuuden pullotusongelma on hyvin vähän tutkittu aihepiiri tieteellisissä tutkimuksissa. Juomateollisuuden koko kaksivaiheiseen prosessiin keskittyviä tutkimuksia sen sijaan on runsaasti ja tämän vuoksi sopivimman algoritmin etsimisessä onkin keskitytty näihin tutkimuksiin. Erona pelkän pullotusvaiheen optimointiin näissä on se, että ne pyrkivät optimoimaan myös eräkokoja sekä synkronoimaan nämä kaksi eri prosessivaihetta toisiinsa mahdollisimman tehokkaasti. Ne siis tarjoavat ratkaisuja huomattavasti laajempaan ongelmaan kuin pelkkään pullotukseen, joten tällöin uskoisin niiden kykenevän ratkaisemaan tarvittaessa pelkän pullotukseenkin liittyvän optimointiongelman.

Ferreira et al. (2009) ovat tutkineet edellä esitetyn kaltaista ongelmaa ja keränneet dataa brasilialaisesta virvoitusjuomatehtaasta. Tämän perusteella he ovat luoneet 15 erilaista laskentatilannetta testatakseen kehittämiään algoritmeja. Näitä samoja laskentatilanteita ovat käyttäneet myöhemmin myös muut tutkijat, joten vertailu eri algoritmien välillä on onnistunut kohtuullisen hyvin. Ferreiran et al. (2009) tutkimuksessa parhaiten menestynyt algoritmi oli heuristinen *relaxation approach* -algoritmi eli RA. Tässä muokattiin yhtäloista siten, että ensin saatiin ratkaistua pullotusongelma ja vasta tämän jälkeen ratkaistiin nesteen valmistamiseen ja näiden vaiheiden väliseen synkronointiin liittyvä ongelma.

Almada-Lobo et al. (2012) kehittivät samaan ongelmaan neljä uutta algoritmia, joita he vertailivat toisiinsa sekä Ferreiran tutkimusryhmineen kehittämään RA-algoritmiin käyttäen samoja 15:tä laskentatilannetta. Tässä tutkimuksessa kaksi uutta algoritmia pärjäsivät selvästi paremmin kuin RA. Arantes et al. (2012) taas käyttivät virvoitusjuomateollisuudessa samassa ongelmassa erilaisia variaatioita memeettisistä algoritmeista eli MA:sta (*memetic algorithms*). Tässä tutkimuksessa parhaiten menestynyt algoritmi oli memeettinen algoritmi, jossa hyödynnettiin tabu hakua. Tabu hausta käytetään englanniksi lyhennettä TS (*tabu search*) ja kyseisestä ratkaisualgoritmista käytettiin lyhennettä MATS. Myös tämä MATS menetelmä menestyi selvästi paremmin kuin aiempi *relaxation approach* -algoritmi.

Franca et al. (2014) käyttivät myös samoja 15:tä laskentatilannetta vertaillessaan uutta kehittämäänsä menetelmää aiempiin menetelmiin. Usein aiemmin tämä tyyppisissä ongelmissa käytetyt ratkaisumenetelmät pohjautuivat sekalukuohjelmointiin, lyhenteeltään MIP (*mixed integer programming*), ja nämä olivat kyennyt pääsääntöisesti ratkaisemaan vain pienempiä ongelmia. Tämän vuoksi nämä matemaattiset mallit oli nyt päätetty yhdistää heuristisiin menetelmiin ratkaisun tehostamiseksi (Franca et al. 2014). Tämä uusi menetelmä yhdisti geneettiset algoritmit ja matemaattisen ohjelmoinnin ja siitä käytettiin lyhennettä GAMP (*genetic algorithm/mathematical programming approach*). Vertailussa olivat GAMP:in lisäksi mukana Arantesin ja kumppaneiden (2012) esittämä MATS menetelmä sekä Almada-Lobon ja kumppaneiden (2012) kaksi menestyneintä algoritmia, joista käytettiin vertailun yhteydessä nimiä F1 ja PS. GAMP menestyi selvästi paremmin kuin pelkkä GA tai MATS. Lisäksi GAMP menestyi juuri ja juuri paremmin kuin F1 ja PS, mutta tässä GAMP käytti laskenta-aikaa ainoastaan yhden neljäsosan siitä mitä F1 ja PS käyttivät.

Almada-Lobo et al. (2014) ovat käyttäneet panimoteollisuudessa samantyyppisen ongelman ratkaisemiseen erilaisia MIP pohjaisia heuristiikoita. Tässä tutkimuksessa laskennallinen vertailu suoritettiin kuitenkin 20:llä heidän omallaan laskentatilanteella. Nämä laskentatilanteet perustuivat oikean panimon dataan. Parhaiten menestynyt algoritmi tässä tutkimuksessa oli MIP pohjainen heuristiikka, joka hyödynsi niin sanottua *relax-and-fit* -lähestymistapaa.

Pienemmissä laskentaongelmissa pelkkä MIP pärjää hyvin, mutta ongelman laajuuden kasvaessa sen käyttämä laskenta-aika nousee liian suureksi. Laskentaa pystytään nopeuttamaan yhdistämällä tähän heuristiikat. Edellä esitettyjen tutkimusten pohjalta voitaisiinkin suositella koko juomateollisuuden kaksivaiheisen prosessin optimointiin matemaattisen ohjelmoinnin ja heuristiikoiden yhdistelmiä. Näiden tulisi soveltua myös pelkän pullotuksen optimointiin, sillä se on toinen vaihe tätä kaksivaiheista prosessia. Erityisesti

geneettisen algoritmin ja matemaattisen ohjelmoinnin yhdistelmä GAMP tarjosi luotettavaa vertailudataa muihin ratkaisualgoritmeihin. Hyvin samantyyppinen on toki rakenteeltaan *relax-and-fix* -strategiaan perustuva MIP pohjainen algoritmi, jonka voisi olettaa menestyvän hyvin samalla tavalla kuin GAMP:in.

4.4 Työntöohjatun raaka-ainekäytön optimointi

Raaka-ainekäytön optimointi voidaan nähdä yhtenä variaationa niin sanotusta *cutting-stock* ongelmasta. Kyseistä ongelmaa on tutkittu varsin vähän lihateollisuuden yhteydessä. Sahateollisuudessa kyseiseen ongelmaan on useampia erilaisia lähestymistapoja, mutta tässä tapauksessa on keskitytty niihin, jotka soveltuvat myös lihateollisuuteen kuin yleisemmälläkin tasolla vastaavan tyyppisiin ongelmiin.

Albornoz et al. (2015) ovat tutkineet juuri *cutting-stock* ongelman ratkaisua lihateollisuudessa. Merkittävimpanä päätöksenä niin tässä kuin muissakin saman tyyppin ongelmassa on se mitä leikkuuasetetta käytetään ja montako kertaa tätä kyseistä asetetta tulee käyttää. Albornoz et al. (2015) ovat käyttäneet ratkaisussaan lineaarista sekalukuohjelmointia, josta käytetään englanninkielisessä kirjallisuudessa lyhennettä MILP (*mixed integer linear programming*). Tämä malli pyrkii maksimoimaan saadut tuotot huomioiden muun muassa annetun kysynnän, puutteesta aiheutuvat kustannukset ja varastoinnin. Tässä ongelman kokoluokka oli varsin maltillinen sisältäen 17 eri leikkuuasetetta ja 40 eri lopputuotetta. Ratkaisut erilaisiin ongelmatilanteisiin saatiin kaikissa tapauksissa noin kahdessa minuutissa käytettäessä *branch&bound* menetelmää IBM ILOG CPLEX -ohjelmistolla.

Borenstein et al. (2018) ovat hyödyntäneet samaa MILP lähestymistapaa sahausasetteiden optimoinnissa. Tässä tapauksessa tavoitteena oli optimoida käytettävät sahausasetteet kysynnän täyttämiseksi samalla huomioiden varastossa olevat tuotteet. Borenstein et al. (2018) vertailivat mallinsa toimivuutta myös suuremmilla ongelmilla, joista suurimmassa oli keskimäärin 311 596 muuttujaa ja 57 rajoitusta. Tämän kokoluokan ongelma ratkaisemiseen kului keskimäärin hieman alle puoli tuntia.

Myös Broz et al. (2018) ovat käyttäneet samaa MILP lähestymistapaa tukkien sahausasetteiden optimoinnissa. He ovat huomioineet laskennassaan pääasiassa täysin samoja asioita kuin Albornoz et al. (2015) edellä. Broz et al. (2018) ovat kuitenkin tarkastelleet ongelmaa monijaksoisessa viitekehysessä, joka tuo ongelmaan lisää laajuutta. He ovat rajoittaneet laskenta-ajan 1500 s eli 25 minuuttiin. Tässä ajassa MILP mallin parhaan ratkaisun ero optimiin oli 5-7 %, joka on aivan riittävän hyvä tarkkuus teollisuuden käyttöön.

Seuraavaksi on esiteltyä muutama muu ratkaisumenetelmä MILP-mallien lisäksi. Yksi näistä on lihanleikkauksen yhteydessä käytetty lineaarinen ohjelmointi (Cammel & Whittaker (1990). Broz et al. (2019) ovat käyttäneet tavoiteohjelmointia (*goal programming*) suuremman kokoluokan tukkien sauhauksen optimoimisessa. Gronalt & Huka (2017) ovat käyttäneet sekalukuohjelmointia tukkien sauhauksessa. Ratkaistavana heillä on ollut suuren kokoluokan ongelma, jossa on ollut 1039 leikkuuasetetta, 597 lopputuotetta ja 66 erilaista raaka-ainetta. Tutkimuksessa optimi löytyi keskimäärin kahden tunnin laskennan jälkeen, mutta Gronalt & Huka (2017) totesivat, että ero parhaan löytyneen ratkaisun ja optimin välillä oli vain noin 1 % jo 10 sekunnin laskennan jälkeen.

Heuristiikoista käytettyjä menetelmiä tämä tyyppisiin ongelmiin ovat geneettiset algoritmit ja simuloitu jäähditys. Ferland et al. (2013) ovat soveltaneet heuristisia ratkaisumenetelmiä tukkien sauhauksen optimointiin. Heidän käyttämänsä heuristiikat ovat juuri geneettiset algoritmit sekä simuloitu jäähditys. Tutkimuksessaan he ovat käyttäneet pelkkiä heuristiikoita ongelman ratkaisemiseen sekä heuristiikoita yhdistettyinä kokonaisluohjelmointiin. Ratkaistavana toimi 11 ongelman joukko ja paras tulos seitsemään ongelmaan yhdestätoista saatiin geneettisen algoritmin ja kokonaisluohjelmoinnin yhdistelmällä. Loppuun neljään ongelmaan paras tulos saavutettiin pelkällä GA:lla. Laskentaajat pysyivät keskimäärin alle minuutissa kaikilla käytetyillä algoritmeilla. Tutkimuksessa tarkasteluajanjakso oli kuitenkin varsin lyhyt, joka osaltaan voi vaikuttaa laskenta-aikoihin. SA pohjaiset tulokset olivat vain hieman heikompia kuin GA pohjaiset, joten myös SA ja SA:n ja kokonaisluohjelmoinnin muodostama hybridialgoritmi ovat täysin käytökelpoisia kyseisten ongelmien ratkaisemiseen.

Wikborg (2008) on soveltanut myös heuristiikkaa lihateollisuuden raaka-ainekäytön optimoinnissa. Hänen käyttämänsä heuristiikka on geneettinen algoritmi, jota on vertailtu lisäksi lineaariseen ohjelmointimalliin. Vertailussa GA:lla saadut tulokset ovat pärjänneet varsin hyvin matemaattisella mallilla saaduille tuloksille. Wikborg (2008) toteaaakin, että käytännön tasolla teollisuudessa ero näiden menetelmien välillä on merkityksetön. Etuna geneettisillä algoritmeilla on niiden suurempi joustavuus sekä soveltuvuus myös suuremman kokoluokan ongelmien ratkaisemiseen (Wikborg 2008). Lisäksi Wikborg (2008) tuo esille sen, että vaikka tässä tarkastelun kohteena onkin lihan leikkauksen optimointi ovat samat asiat sovellettavissa helposti myös tukin sauhukseen.

Edellä esiteltyjen tietojen pohjalta voidaan tehdä johtopäätöksiä siitä, mitkä algoritmit voisivat soveltua erityisen hyvin juuri työntönä saapuvan raaka-aineen käytön optimointiin. Erilaiset sekalukuohjelmointimallit ovat hyvin käytettyjä kirjallisuudessa ja näyttö niiden toiminnasta monenlaisissa tilanteissa on varsin hyvin vahvistettu. Täten niitä ja eri-

tyisesti lineaarista ratkaisumallia eli MILP-mallia voidaan suositella tämän tyyppisten optimointiongelmiin ratkaisemiseen. Lisäksi geneettiset algoritmit joko yksinään tai yhdistettynä edellä esiteltäisiin matemaattisiin malleihin ovat hyviä ratkaisualgoritmeja näihin leikkausongelmiin. Geneettisten algoritmien hyödyt korostuvat entisestään suuremmissa ja monimutkaisemmissa ongelmissa, joissa riittävän hyvä ratkaisu halutaan saada mahdollisimman lyhyessä ajassa.

4.5 Kaksiulotteinen leikkausongelma

Kirjallisuudessa kaksiulotteisiin leikkausongelmiin on esitettyä pääasiassa kaksi eri lähestymistapaa. Ensimmäisessä on valmiina tietyn kokoinen aihio, johon pyritään asettelemaan leikattavat objektit siten, että ylijäävä hukan osuus minimoituu. Toisessa lähestymistavassa taas asiaa käsitellään pakkausongelmana. Tässä pyritään annetulla leveydellä olevaan suorakaiteeseen asettelemaan pienempiä suorakaiteita siten, että isomman suorakaiteen (aihion) korkeus minimoidaan. Tässä ongelmana vain on se, että kyseinen ongelma ei yleensä huomioi giljotiinileikkaustapaa leikattavien kohteiden asettelussa. Tämän vuoksi keskitytäänkin tarkastelemaan ratkaisualgoritmeja erityisesti kaksiulotteiseen kaksivaiheiseen giljotiinileikkausongelmaan (*two-dimensional two-staged guillotine cutting stock problem*).

Suliman (2006) sekä Cui & Zhao (2013) esittävät tutkimuksissaan kaksiulotteisen leikkausongelman ratkaisemiseen tyypillisesti käytettyjä menetelmiä. Näitä ovat lineaarinen kokonaislukuohjelmointi sekä kokonaislukuohjelmointi niin sanotun *column generation* -menetelmän kanssa. Kyseinen *column generation* -menetelmä, lyhenteeltään CG, laskee iteratiivisesti lineaarisen ohjelmointimallin ratkaisuja ja luo uusia ladontakuvioita ratkaisemalla kaksiulotteisia yhden laajan objektin sijoitteluongelmia, lyhenteeltään SLOPP (*single large object placement problem*). Tämä uusi luotu ladontakuviokuva korvaa jo olemassa olevan ja päivittää siten ratkaisua. (Cui & Zhao 2013).

Arenales & Morabito (2000) esittivät ratkaisumenetelmän huonekaluteollisuuden kaksiulotteiseen leikkausongelmaan, joka perustuu vahvasti CG-menetelmään. He ovat ratkaisussaan yhdistäneet lineaarisen ohjelmoinnin ja *column generation* -menetelmän. Ratkaisun toteutus on kaksivaiheinen, jossa ensin määritetään leikkauskuviot jokaiselle pitkittäiselle suikaleelle, joilla on tietty pituus ja eri leveyksiä. Tämän jälkeen määritetään, kuinka monta kertaa jokaista näistä edellä määritetyistä valmiista suikaleista käytetään. Jotta laskenta-aika ei venyisi tarpeettoman pitkäksi, on suosituksena pysäyttää iterointi jollakin säännöllä tilanteissa, joissa tuloksissa ei tapahdu enää merkittäviä muutoksia.

Cui & Zhao (2013) ovat hyödyntäneet esittämässään ratkaisussa myös *column generation* -lähestymistapaa. He ovat käyttäneet tästä ratkaisustaan nimeä RCCG (*repeated constrained column generation*). Kyseinen menetelmä käyttää nimensä mukaisesti *column generation* -menetelmää toistuvasti jäännösongelmien ratkaisemiseen, kunnes kaikkien tuotteiden annettu kysyntä on saatu täytetyksi. Merkittävimpinä eroina yleiseen *column generation* -menetelmään verrattuna RCCG:ssä on se, että se ratkaisee jäännösongelmia toistuvasti, kunnes kysyntä on täytetty. Lisäksi yleinen *column generation* käyttää ratkaisussa rajoittamatonta SLOPP:ia, kun taas RCCG:ssä ratkaisussa käytetään rajoitettua SLOPP:ia.

Cui & Zhao (2013) ovat vertailleet ratkaisumenetelmäänsä kolmeen muuhun ratkaisuun. Vertailussa ovat olleet Cintran et al. (2008) esittämä heuristinen menetelmä, Silvan et al. (2010) esittämä eksakti ratkaisumenetelmä sekä kaupallinen ladontakuvion optimiointiohjelmisto. Tilanteissa, joissa ladottavia kuvioita oli mahdollista kääntää 90 astetta, RCCG pärjasi selvästi paremmin kuin Cintran et al. (2008) malli. Toisin taas tilanteessa, joissa tämä kääntäminen ei ollut mahdollista, pärjasi Cintran et al. (2008) malli hieman paremmin kuin RCCG. Laskenta-ajoissa näillä kahdella heuristiikalla ei ollut mainittavaa eroa. Vertailussa Silvan et al. (2010) eksaktiin menetelmään RCCG pärjasi selvästi paremmin suuremman kokoluokan ongelmassa. Lisäksi laskenta-aika eksaktilla menetelmällä oli huomattavasti pidempi. Kaupalliseen ohjelmistoon verrattuna RCCG vaati vähemmän laskenta-aikaa sekä tarjosi parempia tuloksia. Näiden vertailujen perusteella voitaisiinkin todeta RCCG olevan varsin varteenotettava vaihtoehto erilaisten kaksiulotteisten leikkausongelmien ratkaisemisessa.

Suliman (2006) esittelee yhden heuristisen lähestymistavan kaksiulotteisten leikkausongelmien ratkaisemiseen. Tämä kyseinen lähestymistapa on kolmivaiheinen ja siitä käytetään nimeä peräkkäinen heuristinen menetelmä eli SHP (*sequential heuristic procedure*). Nämä vaiheet ovat seuraavanlaiset:

1. Määritellään yksiulotteisessa tilanteessa leveysuuntainen kuvio, joka minimoi luonnollisesti hukan leveys suunnassa.
2. Määritellään pituussuuntainen kuvio, yhdessä leveysuuntaisen kanssa, joka pyrkii minimoimaan pituussuuntaisen hukan.
3. Määritetään, montako kertaa kyseistä edellä määritettyä leikkauskuviota käytetään annetun kysynnän täyttämiseksi. Kun kysyntä on saatu täytettyä, iteroidaan tilanne uudelleen tälle uudelle kysynnälle.

Tässä mallissa yhtenä kontrolloitavista muuttujista on pituus, joka voi joissakin optimointitilanteissa olla varsin hyödyllinen muuttuja.

Yhteenvetona kaksiulotteisen leikkausongelman ratkaisemisesta voidaan todeta, että varsin hyviä ratkaisumenetelmiä ovat erilaiset toteutukset kokonaislukuohjelmoinnista yhdessä *column generation* -menetelmän kanssa. Erityisesti Cui & Zhaon (2013) esittämä RCCG algoritmi vaikuttaa suoritetun vertailun perusteella tarjoavan varsin hyviä tuloksia kaksiulotteisten leikkausongelmien ratkaisemisessa. Lisäksi Sulimanin (2006) esittämä SHP menetelmä tarjoaa erilaisen lähestymistavan muihin esiteltyihin algoritmeihin verrattuna huomioimalla aihion pituuden eri tavalla kuin muissa tapauksissa. Kyseinen lähestymistapa voi joissakin optimointitapauksissa olla varsin hyödyllinen.

5. OHJELMAKIRJASTOT JA TYÖKALUT TEOLLI- SEEN OPTIMOINTIIN

Tässä luvussa tutustutaan erilaisiin tarjolla oleviin ohjelmakirjastoihin sekä työkaluihin, joita voidaan hyödyntää teollisuudessa vastaan tulevien optimointiongelmien ratkaisemisessa. Tarkoituksena ei ole keskittyä ainoastaan niihin vaihtoehtoihin, joilla voidaan ratkaista juuri tässä työssä esiteltyjä ongelmia tai käyttää juuri niitä algoritmeja, joita edellisessä luvussa on tuotu esille. Tässä luvussa keskitytään erityisesti esittelemään ohjelmakirjastoja ja muita työkaluja, jotka ovat .NET Core -ympäristöön yhteensopivia. Tämä valinta on tehty sen vuoksi, että yhteistyöyrityksen käyttämä kehitysympäristö on .NET Core. Tämän lisäksi esitellään myös muita ohjelmia, ohjelmistoja ja työkaluja, joista voisi olla hyötyä teollisuuden parissa tapahtuvassa matemaattisessa optimoinnissa ja jotka ovat yhteensopivia varsin laajasti eri ympäristöjen ja ohjelmointikielien kanssa.

5.1 .NET Core -yhteensopivat kirjastot ja työkalut

Ensimmäisenä käsitellään .NET Core -ympäristöön yhteensopivia optimointiin soveltuvia kirjastoja, ratkaisijoita sekä muita vastaavan tyyppisiä työkaluja. Monet näistä ovat yhteensopivia myös useiden muiden kehitysympäristöjen kanssa. Nämä ovat koottuna taulukkoon 1. Taulukossa 1 nimi tarkoittaa työkalun nimeä ja kuvauksessa on lyhyesti kuvattu kyseistä työkalua. Taulukon 1 kolmanteen sarakkeeseen ovat koottuna työkalujen hinnat yhdelle käyttäjälle kaupallisessa käytössä, jos kyseinen tieto on ollut julkisesti saatavilla.

Taulukko 1: .NET Core -yhteensopivat kirjastot ja työkalut optimointiin

Nimi	Kuvaus	Hinta, jos tiedossa
Extreme Optimization Numerical Libraries for .NET	Kokoelma yleiskäyttöisiä matemaattisia ja tilastollisia työkaluja	999 USD + mahdollisuus ilmaiseen kokeiluun
Math.NET Numerics	Numeerinen kirjasto, josta voi olla apua joidenkin optimointitehtävien ratkaisemisessa	Ilmainen käyttää
Google OR-Tools	Kombinatoriseen optimointiin tarkoitettu ohjelmisto, yhteensopivuus .NET Core kanssa	Ilmainen käyttää
GUROBI Optimizer	Ratkaisija yleisimpiin ongelmatyyppeihin kuten LP, MILP ja QP	–
LINDO API	Kutsuttava kirjasto, jossa ratkaisijat muun muassa LP, NLP, MILP ja QP ongelmiin	395 USD + mahdollisuus ilmaiseen kokeiluun
LocalSolver	Ratkaisija, jossa myös oma mallinuskikieli, rajapinta .NET Core kanssa	–
LpSolveDotNet	MILP-ratkaisija	Ilmainen käyttää
NAG Library for .NET	Kutsuttava algoritmi kirjasto, muun muassa LP, MILP ja QP ongelmiin	–
NMath	Matematiikka ja statistiikka kirjasto .NET (Core) ympäristöön	1 595 USD + mahdollisuus ilmaiseen kokeiluun
MOSEK	Optimointiohjelmisto, muun muassa LP, MILP ja QP ongelmiin, rajapinta .NET kanssa	Mahdollisuus ilmaiseen kokeiluun

5.2 Muut

Tässä luvussa käsitellään muita optimointiin soveltuvia ohjelmistoja ja muita työkaluja, jotka eivät välttämättä tarjoa suoraa rajapintaa tai muuta yhteensopivuutta .NET Core -ympäristön kanssa. Nämä ovat koottuna taulukossa 2. Suurinta osaa kyseisistä ohjelmistoista ja muista työkaluista voi hyödyntää yleisimpien kehitysympäristöjen ja ohjelmointikielien kanssa. Näitä kaikkia eri vaihtoehtoja ei ole lueteltuna tässä taulukossa,

mutta mainittakoon muutamina esimerkkeinä C++, C# ja Java. Taulukon 2 hinta sarakkeeseen ovat koottuna ohjelmistojen hinnat yhdelle käyttäjälle, jos kyseinen tieto on ollut julkisesti saatavilla.

Taulukko 2: Muita optimointiin soveltuvia ohjelmistoja ja ratkaisijoita

Nimi	Kuvaus	Hinta, jos tiedossa
ALGLIB	Numeerinen kirjasto, joka sisältää ominaisuuksia mm. optimointiin	1 470 USD + mahdollisuus ilmaiseen kokeiluun
Artelys Knitro	Ratkaisija erityisesti epälineaarisiin ongelmiin, mutta soveltuu myös muihin optimointiongelmiin	–
FICO Xpress optimization	Ratkaisija muun muassa LP, MILP ja QP ongelmiin, rajapinta muun muassa .NET kanssa	–
Frontlinesolvers Solver SDK	Suurien optimointiongelmiin ratkaisemiseen	1. vuoden lisenssi 1 995 USD + mahdollisuus ilmaiseen kokeiluun
GAMS	Mallinnuskieli, joka sisältää myös perusratkaisijoita, lisäksi runsas valikoima muita ratkaisijoita lisähinnalla	Perusmoduuli 3 200 USD
IBM ILOG CPLEX	Matemaattiseen optimointiin tarkoitettu ohjelmisto, toimii myös kutsuttavana kirjastona	–
LINDO API	Kutsuttava kirjasto, jossa ratkaisijat muun muassa LP, NLP, MILP ja QP ongelmiin	395 USD + mahdollisuus ilmaiseen kokeiluun
LINDO LINGO	Mallinnuskieli ja -ohjelmisto ratkaisijoilla, toimii myös kutsuttavana kirjastona	495 USD + mahdollisuus ilmaiseen kokeiluun
MIDACO	Ratkaisija, joka perustuu ACO algoritmiin, saatavilla muun muassa C# kielellä	3 600 EUR elinikäinen lisenssi + mahdollisuus ilmaiseen kokeiluun
NAG Library	Kutsuttava algoritmi kirjasto monille yleisimmille ohjelmointikielille ja ympäristöille, muun muassa LP, MILP ja QP ongelmiin	–
Scilab	MATLAB tyyppinen ohjelmisto numeeriseen laskentaan	Ilmainen käyttää

6. PÄÄTELMÄT

Tässä kandidaatintyössä käsiteltiin matemaattisen optimoinnin hyödyntämistä teollisuuden optimointiongelmien ratkaisemisessa. Ensimmäiseksi työssä käsiteltiin matemaattisen optimoinnin taustaa. Tämän jälkeen luvussa 3 esiteltiin yleisimpiä eri teollisuudenaloilla kohdattavia optimointiongelmia, joita olivat: annetuilla työtehtävillä tapahtuva asetusaikojen minimointi, työntöohjatun raaka-ainekäytön optimointi, tuotantosuunnitelman optimointi, resurssikäytön optimointi ja kaksiulotteinen leikkausongelma.

Luvussa 4 käsiteltiin eri ratkaisualgoritmeja ja pyrittiin kirjallisuusselvityksen perusteella muodostamaan suosituksia eri ongelmiin parhaiten soveltuvista ratkaisumenetelmistä. Saadut tulokset ovat koottuna taulukkoon 3.

Taulukko 3: Yhteenveto ratkaisualgoritmeista

Optimointiongelma	Suosittelavat ratkaisualgoritmit
Kokoonpanotuotannon optimointi	ACO
Juomateollisuuden pullotusongelma	MIP, MILP ja GAMP
Työntöohjatun raaka-ainekäytön optimointi	MILP ja GA
Kaksiulotteinen leikkausongelma	SHP ja RCCG

Työn tekemiseen käytettävissä oleva aika on ollut jossain määrin rajallinen sekä tutkimusta matemaattisen optimoinnin hyödyntämisestä teollisuudessa on tehty todella runsaasti. Täten selvitettäessä parhaiten soveltuvia ratkaisualgoritmeja ei ole ollut mahdollista käydä läpi kaikkia mahdollisia tutkimuksia ja vertailla näiden tarjoamia tuloksia. Täten tämän työn suositukset perustuvatkin siihen näkemykseen, joka on muodostunut läpikäydyn materiaalin perusteella. Työtä varten tutkittua materiaalia on varsin runsaasti, joten usko suositusalgoritmien toimivuudesta kyseisien ongelmien ratkaisemiseen on kohtalaisen vahva. Jokainen ongelma on kuitenkin aina ainutlaatuinen tapaus, joten myös optimointia suorittavalta taholta vaaditaan asiantuntemusta aihepiiristä ja juuri kyseisen ongelmatilanteen ymmärtämistä.

Lopuksi luvussa 5 esiteltiin erilaisia ohjelmakirjastoja sekä muita työkaluja, joita voidaan hyödyntää teollisessa optimoinnissa. Kyseisessä luvussa keskityttiin pääasiassa tarkastelemaan .NET Core -yhteensopivia työkaluja, mutta myös muita tyypillisimmille ohjelmointikielille ja kehitysympäristöille saatavia ohjelmistoja ja muita työkaluja käsiteltiin.

Kandidaatintyöstä saatuja tuloksia voidaan hyödyntää kehitettäessä teollista optimointikykyä. Ratkaisualgoritmi suositusten perusteella voidaan lähteä etsimään juuri kyseiseen ongelmaan parhaiten soveltuvia ratkaisumenetelmiä ja ohjelmakirjastosuositukset tarjoavat hyvän lähtökohdan etsittäessä valmiiksi tarjolla olevia optimointiin soveltuvia työkaluja.

Teolliseen optimointiin liittyy myös tulevaisuuden tavoitteita. Jännes et al. (2020) totesivat, että nykypäivänä kyetään esimerkiksi optimoimaan tuotannossa yksi avainprosessi. Tavoitteena olisi kuitenkin kyetä optimoimaan suurempia kokonaisuuksia ja lisäksi kyetä sovittamaan nämä yhteen. Lisäksi Jännes et al. (2020) toivat esiin sen, että nykyiset optimointimallit osaavat kyllä toimia sääntöjen mukaan, mutta joissain tapauksissa näitä sääntöjä tulisi kyetä myös rikkomaan. Optimointimallien kyky rikkoa sääntöjä olisikin toinen merkittävä tulevaisuuden tavoite teollisen optimoinnin parissa. Tekoälyn avulla tällainen toiminta voisi olla mahdollista. Tekoälyn ja optimoinnin yhdistäminen voisi täten olla mielenkiintoinen tutkimuksen kohde tulevaisuudessa.

LÄHTEET

- Albornoz, V., González-Araya, M., Gripe, M., Rodriguez, S. & Trevino, E. (2015). An optimization model for planning operations in a meat packing plant, in de Werra, D., Parlier, G. & Vitoriano, B. (ed.) *Operations Research and Enterprise Systems*, 4th International Conference, ICORES 2015, Springer International Publishing Switzerland, pp. 136–146.
- Almada-Lobo B., Baldo, T., Morabito, R. & Santos, M. (2014). An optimization approach for the lot sizing and scheduling problem in brewery industry, *Computers & Industrial Engineering*, Vol. 72, pp. 58–71.
- Almada-Lobo, B., Clark, A., Ferreira, D. & Morabito, R. (2012). Single-stage formulations for synchronised two-stage lot sizing and scheduling in soft drink production, *International Journal of Production Economics*, Vol. 136 (2), pp. 255–265.
- Arantes M., Franca, P., Morabito, R. & Toledo, C. (2012). A Memetic Framework for Solving the Lot Sizing and Scheduling Problem in Soft Drink Plants, in Chiong, R., Michalewicz, Z. & Weise, T. (ed.) *Variants of Evolutionary Algorithms for Real-World Applications*, Springer-Verlag Berlin Heidelberg, Berliini, pp. 59–93.
- Arenales, M. & Morabito, R. (2000). Optimizing the cutting of stock plates in a furniture company, *International Journal of Production Research*, Vol. 38 (12), pp. 2725–2742.
- Bahubalendruni, M., Bala Murali, G., Biswal, B. & Deepak, B. (2018). Assembly sequence planning using soft computing methods: A review, *Proceedings of the Institution of Mechanical Engineers Part E: Journal of Process Mechanical Engineering*, pp. 1–31.
- Bangert, P. (2012). *Optimization for Industrial Problems*, 1. ed., Springer-Verlag Berlin Heidelberg, Berliini.
- Bartz-Beielstein, T., Filipič B., Korošec P., Melab N., Naujoks B. & Talbi, E. (2018). D1.3 Potential complex optimization problems in science and industry, *Synergy*, vii-tattu: 23.1.2020. Saatavilla: <http://www.synergy-twinning.eu/files/synergy-d13-v24-180413-0642.pdf>
- Borenstein, D., da Silveira Farias, E. & Parra Galvez, J. (2018). Application of optimization for solving a sawing stock problem with a cant sawing pattern, *Optimization Letters*. Vol.12 (8), pp. 1755–1772.
- Broz, D., Corsano, G., Montagna, J. & Vanzetti, N. (2018). An optimization approach for multiperiod production planning in a sawmill, *Forest Policy and Economics*, Vol. 97, pp. 1–8.
- Broz, D., Corsano, G., Montagna, J. & Vanzetti, N. (2019). Goal programming application for the decision support in the daily production planning of sawmills, *Forest Policy and Economics*, Vol. 102 (102), pp. 29–40.
- Cammel, S. & Whitaker, D. (1990). A Partitioned Cutting-stock Problem Applied in the Meat Industry, *Journal of the Operational Research Society*, Vol. 41 (9), pp. 801–807.

- Cintra, G., Miyazawa, F., Wakabayashi, Y. & Xavier, E. (2008). Algorithms for two-dimensional cutting stock and strip packing problems using dynamic programming and column generation, *European Journal of Operational Research*, Vol. 191 (1), pp. 61–85.
- Cui, Y. & Zhao, Z. (2013). Heuristic for the rectangular two-dimensional single stock size cutting stock problem with two-staged patterns, *European Journal of Operational Research*, Vol. 231 (2), pp. 288–298.
- Farnca., P., de Freitas Pereira, R., Morabito, R., de Oliveira L. & Toledo, C. (2014). A genetic algorithm/mathematical programming approach to solve a two-level soft drink production problem, *Computers and Operations Research*, Vol. 48, pp. 40–52.
- Ferreira, D., Morabito, R. & Rangel, S. (2009). Solution approaches for the soft drink integrated production lot sizing and scheduling problem, *European Journal of Operational Research*, Vol. 196(2), pp. 697–706.
- Ferland, J., Garces, J., Parada, V. & Pradenas, L. (2013). Genotype-phenotype heuristic approaches for a cutting stock problem with circular patterns, *Engineering Applications of Artificial Intelligence*, Vol.26(10), pp. 2349–2355.
- Gronalt, M. & Huka, M. (2017). Model development and comparison of different heuristics for production planning in large volume softwood sawmills, *Engineering Optimization*, Vol. 49(11), pp. 1829–1847.
- Haataja, J (2004) Optimointitehtävien ratkaiseminen, 3. painos, Picaset Oy, Helsinki.
- Hutabarat, W., Rashid, M. & Tiwari, A. (2012). A review on assembly sequence planning and assembly line balancing optimization using soft computing approaches, *International Journal of Advanced Manufacturing Technology*, Vol. 59 (1–4), pp. 335–349.
- Jännes, T., Kanninen, K., Rovio, T., Salomäki T. & Toivonen, V. (2020). Pinja Oy, Tampere, haastattelu 7.2.2020.
- Ketonen, M., Korvola, T. & Rummukainen, H. (2009). Kehittyneiden optimointimenetelmien sovellus sahan tuotannosuunnittelussa, teoksessa Ketonen, M. (toim.) *Automaatio XVIII Seminaari 17.-18.3.2009 Helsinki*, SAS julkaisusarja nro 36., Suomen Automaatioseura ry., Helsinki, viitattu 18.1.2020. Saatavilla: https://www.vtt.fi/inf/julkaisut/muut/2009/AutomaatioXVIII_seminaari2009_artikkeli.pdf
- Lu, C., Yang Z. & Zhao, H. (2013). An ant colony algorithm for integrating assembly sequence planning and assembly line balancing, *Applied Mechanics and Materials*, Vol. 397–400, pp. 2570–2573.
- Mattila, M. (2018). MS-C2105 Optimoinnin perusteet: luento 13 heuristinen optimointi – luentokalvot, Aalto yliopisto.
- Paschos, V. (2014). *Application of combinatorial optimization*, 2. ed., ISTE, Lontoo.
- Sahateollisuus ry, Sahateollisuus-verkkosivusto, viitattu 4.3.2020. Saatavilla: <https://sahateollisuus.com>
- Silva, E., Alvelos, F. & Valério de Carvalho, J. (2010). An integer programming model for two- and three-stage two-dimensional cutting stock problems, *European Journal of Operational Research*, Vol. 205 (3), pp. 699–708.

Suliman, S. M. (2006). A sequential heuristic procedure for the two-dimensional cutting-stock problem, *International Journal of Production Economics*, Vol. 99 (1), pp. 177–185.

Työ- ja elinkeinoministeriö (2019). Toimialaraportit: Elintarviketeollisuus 365 ruoan päivää, Työ- ja elinkeinoministeriön julkaisuja 2019:61, Helsinki.

Wikborg, U. (2008). *Online Meat Cutting Optimisation*, thesis, Norwegian University of Technology.

Yang, X. (2010). *Engineering optimization: An Introduction with Metaheuristic Applications*, John Wiley & Sons, Hoboken.