

Katsaus modulaaristen piirilevyldontakoneiden käyttöön  
liittyvistä optimointiongelmista

TURUN YLIOPISTO  
Pro gradu -tutkielma  
Tulevaisuuden teknologioiden laitos  
Tietojenkäsittelytiede  
Oskari Kivinen  
2019

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

TURUN YLIOPISTO  
Tulevaisuuden teknologioiden laitos

OSKARI KIVINEN: Katsaus modulaaristen piirilevyldontakoneiden rakenteeseen ja käyttöön liittyviin optimointiongelmiin

Pro gradu -tutkielma, 70 s., 6 liites.  
Tietojenkäsittelytiede  
Huhtikuu 2019

---

Työssä luodaan katsaus modulaaristen ladontakoneiden rakenteeseen, toimintaan ja uudelleenkonfigurointiin. Tämän lisäksi tarkastellaan kyseiseen ladontakoneentyypiin keskeisesti liittyvää optimointiongelmaa, eli ladontakonekonfiguraation ja linjaston tasapainotuksen ongelmaa. Tutkielmassa kehitetään simuloitua jäähdystystä hyödyntävä ratkaisumenetelmä modulaarisen ladontakoneen uudelleenkonfigurointiin.

Modulaariset ladontakoneet edustavat modernia ladontakoneityyppejä. Sen tärkeimpänä erona moniin muihin ladontakoneityyppeihin nähden on moduulien uudelleenkonfigurointimahdollisuus. Toisin sanoen, koneen osia pystytään vaihtamaan siten, että sillä pystytään tehokkaasti latomaan erilaisia piirilevyjä ilman, että koko ladontakone jouduttaisiin vaihtamaan toiseen. Tästä syystä tämäntyyppinen ladontakone sopii erityisen hyvin nykyaikaisen elektroniikkateollisuuden tarpeisiin, jossa ladottavana olevat piirilevyt vaihtuvat tiuhaan tahtiin esimerkiksi tuotteiden lyhentyneiden elinkaarien sekä tuotannossa olevien lukuisten varianttien vuoksi.

Modulaaristen ladontakoneiden optimointiongelmissa ladontakonekonfiguraation määrääminen ja linjaston tasapainotusongelma, eli lyhyemmin MCLB-ongelma on tyypillinen kyseiseen koneentyypin liittyvä optimointiongelma. Nimensä mukaisesti optimointitehtävä koostuu kahdesta tiukasti yhteen nivoutuvasta laajemmasta ongelmasta: ladontakone-moduulien konfiguroinnista sekä näiden moduulien muodostaman linjaston tasapainotuksesta. Ongelman ratkaisu ei ole triviaali, sillä siihen liittyy useita muuttujia. Valmistettavana oleva piirilevy määrittelee ladontakoneissa käytettävät komponenttien käsittelyyn tarvittavat työkalut, eli käytännössä ladontapäät ja niihin asennettavat suuttimet. Konekonfiguraatiossa joudutaan huomioimaan keskinäisiä yhteensopivuuksia sekä suuttimien ja ladontapäiden, että komponenttien ja suuttimien välillä. Linjaston tasapainotuksen tarkoituksena on allokoida valmistuksessa tarvittavat komponentit eri ladontamoduuleille siten, että piirilevyn valmistusaika minimoituu. Käytännössä MCLB-ongelma on laskennallisesti vaativa, jolloin joudutaan tyytymään alioptimaalisiin ratkaisuihin.

Työssä MCLB-ongelman ratkaisemiseksi on esitetty simuloitua jäähdystystä hyödyntävä ratkaisumenetelmä. Simuloitu jäähdystys on metaheuristinen ja todennäköisyyspohjainen optimointimenetelmä, jota käytetään yleisesti nimenomaan vaikeiden kombinatoristen optimointiongelmiin ratkaisuun. Se on saanut vaikutteita metallin karkaisusta, jossa käsiteltävänä olevaa metallia jäähdytetään hallitusti. Simuloidun jäähdystyksen suorituskykyyn vaikuttaa merkittävästi nk. jäähdystysaikataulu, jonka vaikutusta esitellään myös työn suorituskykymittauksia käsittelevässä osiossa.

Tutkimuskysymys: Ladontakonekonfiguraation ja linjaston tasapainotuksen ongelman ratkaiseminen käyttäen simuloitua jäähdystystä.

Avainsanat: Modulaariset ladontakoneet, optimointi, simuloitu jäähdystys

UNIVERSITY OF TURKU  
Department of Future Technologies

OSKARI KIVINEN: A survey on the optimization problems regarding the structure and operation of modular placement machines

Master's Thesis, 70 p., 6 app. p.  
Computer Science  
April 2019

---

This work studies the structure, operation and reconfiguration of modular PCB placement machines. Machine configuration and line balancing, a fundamental optimization problem regarding this machine type, is studied in detail. A simulated annealing -based method to solve the reconfiguration of the placement machine is presented in this thesis.

Modular placement machines represent a modern type of placement machines. Their main difference compared to other placement machine types is the possibility of reconfiguring its placement modules. In other words, many parts on the machine can be easily replaced in order to manufacture different kinds of PCBs without replacing the entire machine. For this reason, the modular placement machine is particularly well suited to the needs of contemporary electronics manufacturing industry, where manufactured PCBs are constantly changing due to short product lifespans and broad variety of nearly-identical variants of similar products.

The machine configuration and line balancing problem, commonly abbreviated the MCLB-problem, is a typical combinatorial optimization problem related to modular placement machines. As the name suggests, the optimization problem consists of two tightly intertwined problems: determining the configuration of the individual placement modules and balancing the assembly line consisting of those modules. Finding the solution to this problem is not trivial because the problem entails several variables. The manufactured PCB determines which mounting tools, or more specifically, which placement heads and nozzles are needed to mount the electrical components on the PCB. There are several compatibility factors between the nozzles and the heads and between the components and the nozzles, which all need to be considered when the machine is configured. The purpose of the line balancing is to allocate the components between different modules in a way that minimizes the time needed to manufacture a single PCB. Practical-sized MCLB-problems are computationally demanding, which means that one has to settle for suboptimal solutions.

A simulated annealing -based solution method for solving the MCLB-problem is presented in this work. Simulated annealing is a metaheuristic and probability-based optimization method that is commonly used in solving hard combinatorial optimization problems. Simulated annealing has been influenced by a technique used in metallurgy, where controlled cooling of metals is used to improve their properties. The performance of the simulated annealing algorithm is greatly affected by the so-called cooling schedule. The effects of choosing the appropriate cooling schedule is examined in the benchmark portion of this work.

Research problem: Solving the machine configuration and line balancing problem using simulated annealing.

Keywords: Modular placement machines, optimization, simulated annealing

# SISÄLLISLUETTELO

1 JOHDANTO .....	1
2 MODULAARINEN PIIRILEVYLADONTAKONE.....	7
2.1 Koneen rakenne.....	7
2.2 Koneen toiminta .....	10
2.2.1 Piirilevyn syöttäminen ladontakonelinjastolle.....	12
2.2.2 Piirilevyn kohdistusmerkkien tunnistaminen.....	12
2.2.3 Juotospastan paino.....	12
2.2.4 Suuttimien vaihtaminen suutinpankista .....	13
2.2.5 Komponenttien noukkiminen.....	13
2.2.6 Komponenttien tarkastaminen .....	14
2.2.7 Komponenttien asettaminen piirilevyllä .....	14
2.2.8 Juottaminen .....	15
2.2.9 Laadunvalvonta .....	15
2.2.10 Valmiiden piirilevyjen kerääminen ladontalinjastolta .....	16
3 MODULAARISEN LADONTAKONEEN REKONFIGUROINTI .....	17
3.1 Päiden uudelleenkonfigurointi .....	17
3.2 Liukuhihnatyyppin rekonfigurointi.....	18
3.3 Syöttimien vaihto .....	19
3.4 Gantryjen siirto ja vaihto.....	20
4 LADONTAKONELINJASTON KÄSITTELYMODUULIT .....	21
4.1 Ladontakonelinjaston alku- ja loppupää .....	22
4.2 Liukuhihnamoduulit.....	22
4.3 Hylättyjen piirilevyjen tarkastusasema .....	22
4.4 Puskurimoduulit keskellä linjastoa .....	22
4.5 Ladontalinjaston aukkomoduulit.....	22
4.6 Sivuttaissiirtomoduulit .....	23
4.7 Risteysmoduulit .....	24
4.8 Kääntömoduuli.....	24

5 PIIRILEVYLADONTAKONEIDEN OPTIMOINTIONGELMAT .....	25
5.1 Valmistuksen osaongelmat .....	25
5.2 Modulaarisiin ladontakonelinjastoihin liittyviä optimointiongelmia.....	26
5.3 Yksittäisiin ladontakoneisiin liittyviä optimointiongelmia.....	28
5.4 Tuotannon suunnitteluun liittyvät optimointiongelmät.....	30
6 LADONTAKONEKONFIGURAATION JA LINJASTON TASAPAINOTUKSEN ONGELMA.....	32
6.1 Ongelman yleiskuvaus .....	32
6.2 Ongelman yksityiskohdat ja ratkaisumalli .....	32
6.3 Ongelman kolmivaiheinen ratkaisuperiaate.....	37
6.3.1 Ladontapäiden määrääminen moduuleille .....	37
6.3.2 Ladottavien komponenttien määrääminen ladontapäille .....	39
6.3.3 Suuttimien määrääminen ladontapäille.....	40
7 LADONTAKONEKONFIGURAATION JA LINJASTON TASAPAINOTUKSEN ONGELMAN RATKAISEMINEN KÄYTTÄEN SIMULOITUA JÄÄHDYTYSTÄ .41	
7.1 Simuloitu jäähdytys.....	41
7.2 Algoritmi .....	42
7.3 Systemin jäähdytysaikataulu .....	44
8 SUORITUSKYKYMITTAUKSET .....	45
8.1 Alustava testaus parametrien määrittämiseksi .....	46
8.2 Simuloidun jäähdytyksen suorituskyky verrattuna muihin menetelmiin.....	52
8.3 Tehtävälölköjen aikakertoimen vaikutus ratkaisumenetelmien suorituskykyyn .56	
9 LOPUKSI.....	64
LÄHDELUETTELO.....	66
Liite 1: Englanninkieliset termit ja niistä käytetyt suomennökset .....	71
Liite 2: Simuloidun jäähdytyksen suorituskykymittaukset erilaisilla aloituslämpötiloilla ja jäähdytysaikatauluilla.....	73
Liite 3: MATLAB-ohjelmalistaukset.....	74

# 1 JOHDANTO

Tässä tutkielmassa perehdytään modulaaristen piirilevyldontakoneiden (engl. modular placement machines) rakenteeseen ja toimintaan, sekä niiden käyttöön liittyviin optimointiongelmiin.

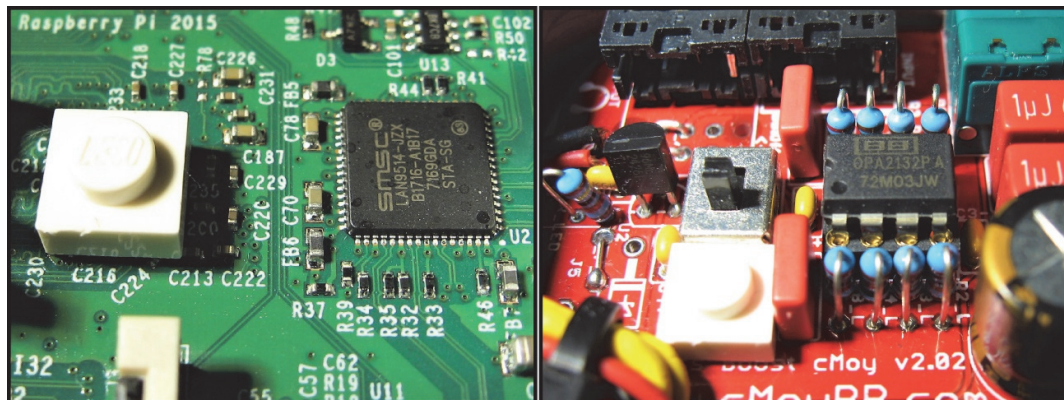
Elektroniikkateollisuus on globaalisti merkittävä teollisuudenala, johon kuuluu elektronisten laitteiden suunnittelu ja valmistus. Sähkö- ja elektroniikkalaitteita valmistavien yritysten yhteenlasketun vuotuisen liikevaihdon arvioidaan olevan maailmanlaajuisesti noin 4,5 biljoonaa Yhdysvaltain dollaria [1]. Vuoden 2017 liikevaihdon mitattuna kolme suurinta elektroniikka-alan yritystä maailmassa ovat yhdysvaltalainen Apple (229 miljardia dollaria), eteläkorealainen Samsung Electronics (212 miljardia dollaria) sekä taiwanilainen Foxconn (155 miljardia dollaria) [2]. Kiinassa valmistetaan sähkö- ja elektroniikkalaitteita vuosittain lähes 2400 miljardin dollarin arvosta ja se onkin tärkein valmistusmaa 51 % osuudella maailman markkinoista. Muita suuria valmistusmaita ovat Yhdysvallat (9 %), Japani (7 %), Korea (6 %), Saksa (3 %), Taiwan (2 %) ja Malesia (1 %). Elektroniikkateollisuuden tärkeimpiä alatoimialoja ovat muun muassa elektroniikan komponentit, automaatio, viihde-elektroniikka, tietoliikenne ja tietokoneet. Maailmanlaajuisesti sähkö- ja elektroniikkateollisuudessa työskentelee 27 biljoonaa työntekijää. [3]

Elektronisten laitteiden elektroniikkaosien massatuotanto perustuu nykypäivänä automatisoitujen piirilevyldontakoneiden käyttöön, jolloin elektronisten komponenttien asennus eli kalustus tehdään piirilevyille koneellisesti [4, s. 1311]. Ennen automatisoitujen piirilevyldontakoneiden hyödyntämistä, komponenttien asennus piirilevyille oli huomattavasti työintensiivisempää [5, s. 553].

Elektroniikkaa sisältävien laitteiden piirilevyt valmistetaan nykyisin suurelta osin nk. *pintaliitostekniikkaa* käyttäen. Pintaliitostekniikka (engl. surface-mount technology, SMT) on valtaosin korvannut aiemmin käytössä olleen piirilevyjen valmistusmenetelmän, jossa komponentit on asennettu piirilevyn läpi, jolloin käytetään

niin sanottuja läpiladottavia komponentteja (engl. through-hole technology). *Läpiladontatekniikkaa* käytetään kuitenkin vielä sovelluksissa, joissa ei voida tai haluta jostain syystä käyttää pintaliitoskomponentteja. [5, s. 553] Esimerkiksi paljon virtaa vievissä laitteissa, kuten televisioissa tai vahvistimissa joudutaan käyttämään läpiliitoskomponentteja. Läpiliitoskomponentteja käytetään myös silloin, kun halutaan valmistaa tavallista vankkarakenteisempia laitteita esimerkiksi olosuhteisiin, joissa laitteet saattavat altistua tavanomaista suuremmalle ulkoiselle rasitukselle [5, s. 553], [6, s. 1488].

Pintaliitostekniikka mahdollistaa useita etuja piirilevyjen valmistuksessa verrattaessa sitä läpiladontatekniikkaan. Näitä etuja ovat muun muassa valmiin piirilevyn suurempi komponenttitiheys sekä pienempi paino, tilavuus ja valmistuskustannukset. Nämä edut saavutetaan pintaliitostekniikassa käytettävien pintaliitoskomponenttien ansiosta, jotka ovat vastaavia läpiliitoskomponentteja huomattavasti kevyempiä ja pienempiä [7, s. 1]. Pintaliitoskomponenttien koneellinen käsittely on helpompaa kuin läpiladottavien komponenttien [8, s. 284]. Kuvassa 1 verrataan pinta- ja läpiliitostekniikoiden yleistä ulkonäköä.



Kuva 1: Pinta- ja läpiliitostekniikoita vertaava kuva. Vasemmanpuoleiselle piirilevyllä koneellisesti ladotut pintaliitoskomponentit ovat erittäin pieniä verrattuna oikealla oleviin käsin juotettuihin läpiliitoskomponentteihin. Komponenttien kokoa on pyritty havainnollistamaan asettamalla piirilevyllä standardi 1 × 1 -kokoinen legolevy. Neliönmuotoisen legolevyn sivujen pituudet ovat n. 7,85 mm.

Piirilevyldontakoneiden toimintaa optimoimalla, eli suoritustehoa (engl. throughput) parantamalla voidaan usein nopeuttaa koko tuotantoprosessin toimintaa, sillä itse ladontaprosessi muodostaa useasti rakennettavana olevan laitteen kokoonpanolinjastoon



eräänlaisen pullonkaulan, joka rajoittaa vastaavasti koko linjaston suoritustehoa [4, s. 1311], [9, s. 893], [10, s. 349].

Piirilevyloadontakoneiden suoritustehon parantamista voidaan tarkastella kahdesta eri näkökulmasta riippuen siitä, millaisessa käytössä loadontakoneet ovat. Mikäli piirilevyloadontakonetta käytetään suurten sarjojen valmistukseen, eli loadottava piirilevy ja siihen kiinnitettävät komponentit pysyvät pitkiä aikoja muuttumattomina, on järkevää yrittää optimoida itse loadontaprosessin yksityiskohtia siten, että yksittäisen piirilevyn valmistamiseen kuluu mahdollisimman lyhyt aika. Jos taas samaa piirilevyloadontakonetta käytetään vain pienten sarjojen valmistamiseen, eli loadottavat piirilevyt ja niihin asennettavat komponentit vaihtuvat lyhyin aikaväleihin, on hyödyllisempää ryhmitellä ja järjestää loadontatöitä siten, että töiden välillä siirtyminen tapahtuu mahdollisimman nopeasti. [4, s. 1311]

Valmistettujen elektronisten laitteiden, kuten kuluttajille suunnatun viihde-elektroniikan kehitys on nykyisin sangen nopeaa. Tämä aiheuttaa sen, että tuotteiden elinkaari on usein lyhyt ja yhtä tuotetta valmistetaan vain suhteellisen lyhyen aikaa. Tämän lisäksi samantyyppisten, mutta hieman erilaisten tuotteiden kirjo saattaa olla yhdellä valmistajalla suuri [11, s. 1189]. Hyvänä esimerkkinä ovat suuret matkapuhelinvalmistajat, joilla on tyypillisesti myynnissä samanaikaisesti lukuisia erihintaisia, vain hieman toisistaan rakenteellisesti eroavia malleja. Tämän lisäksi ainakin osa markkinoilla olevista puhelimista korvautuu käytännössä uusilla puhelinmalleilla vähintään kerran vuodessa. Tuotettavien laitteiden lyhyet elinkaaret ja sangen laajat tuotevalikoimat aiheuttavat sen, että tuotantolinjastoja joudutaan uudelleenkonfiguroimaan usein. Toisaalta suoritustehokkuuden eli käytännössä tuotteiden valmistusnopeuden pitäisi pysyä korkeana. Valmistajien kannalta tuotantolinjastojen ja erityisesti piirilevyloadontakoneiden pitäisi siis olla helposti ja edullisesti uudelleenkonfiguroitavia. Tuotantolinjaston joustavuudella ja suoritusteholla on huomattava vaikutus tuotteen valmistuskustannuksiin, joka vaikuttaa luonnollisesti myös tuotteiden hintoihin ja tätä kautta osaltaan myös yritysten kannattavuuteen. [6, s. 1488] Tässä tutkielmassa tarkasteltuja modulaarisia loadontakoneita markkinoidaan erityisesti tuomaan joustavuutta piirilevyjen loadontaprosessiin [12], [13], [14].

Piirilevyladontakoneet voidaan jakaa eri kategorioihin niiden rakenteen ja toiminnan mukaisesti. Ladontakoneiden toiminnan optimointiin vaikuttavat kunkin ladontakonetyypin erityispiirteet. Itse ladontakoneita ei ole kuitenkaan millään tavalla standardoitu [9, s. 894]. Ayob ja Kendall jakavat pintaliitoskomponenttien ladontaan käytettävät ladontakoneet viiteen eri kategoriaan: *kaksoissyöttö-* (engl. dual-delivery), *moniasema-* (engl. multi-station), *revolver-* (engl. turret-type), *monipää-* (engl. multi-head), sekä *peräkkäistoimisiin nouki-ja-asetatyyppisiin* (engl. sequential pick-and-place) ladontakoneisiin. [9, sivut 897-900] Tässä tutkielmassa tarkasteltavat *modulaariset piirilevyladontakoneet* kuuluvat Ayobin ja Kendallin määritelmän mukaan moniasema-tyyppisiin ladontakoneisiin.

Tutkielmassa tarkasteltava modulaarinen ladontakone eroaa muista ladontakonetyypeistä etenkin joustavuudellaan. Modulaarisista ladontakoneista rakennetaan linjasto, jossa jokainen moduuli vastaa tiettyjen komponenttien asentamisesta ladottavana olevalle piirilevylle. Joustavuus syntyy *ladontamoduulien* helposta muokattavuudesta – vaihtamalla ladontamoduuleihin esimerkiksi erilaisia ladontapäitä, suuttimia ja komponenttisyöttimiä, voidaan ladontalinjastoa muokata siten, että sillä voidaan tehokkaasti latoa kulloinkin ladottavana olevaa piirilevyerää. Muokattava, joustava ladontalinjasto vaikuttaa valmistajan näkökulmasta hyvältä ratkaisulta, mutta optimaalisen suorituskyvyn saavuttaminen tämällytyypisessä järjestelmässä ei ole triviaalia. Tämä johtuu muokattavuudesta, joka tuo optimointitehtävään useita eri muuttujia.

Modulaarisen piirilevyladontakoneen rakennetta ja toimintaa tarkastellaan luvussa 2. Koneen rakennetta käsitellään yksityiskohtaisesti siten, että lukija pystyy muodostamaan käsityksen ladontakoneen eri osista ja ladontaprosessiin kuuluvista työvaiheista. Ladontaprosessi pyritään kuvaamaan mahdollisimman kokonaisvaltaisesti – tyhjän piirilevyn syöttämisestä linjastolle aina valmiiksi kalustetun piirilevyn keräämiseen linjaston loppupäästä. Pääluvussa 3 käsitellään modulaaristen ladontakoneiden uudelleenkonfigurointia, eli sitä kuinka ladontakonemoduulia voidaan muokata vaihtamalla siihen erilaisia osia aina kulloisenkin tarpeen mukaan. Markkinoilla olevien

ladontakoneiden muokattavuusaste vaihtelee melko suuresti, mutta tyypillistä on, että koneisiin pystyy vaihtamaan ainakin erilaisia ladontapäitä sekä näihin kiinnitettäviä suuttimia.

Luvussa 4 käsitellään ladontakonelinjastoilla olevia piirilevyjen käsittelylaitteita. Erilaisia liukuhihnamoduuleita käyttämällä pystytään esimerkiksi rakentamaan monimutkaisia linjastoja, joissa piirilevyjä voidaan reitittää kulkemaan eri ladontakoneiden lävitse kulloistenkin tarpeiden mukaisesti. Käsittelylaitteisiin kuuluu myös laitteet, joilla esimerkiksi kerätään valmiit piirilevyt ladontalinjastolta.

Pääluvussa 5 tehdään katsaus ladontakoneisiin liittyviin optimointiongelmien ja niiden keskinäiseen hierarkiaan. Tässä yhteydessä tarkastellaan lyhyesti myös eräitä modulaaristen ladontakoneiden optimointiin liittyviä teoksia. Tämän tutkielman kannalta tärkeäksi lähdeoteeksi muodostuu Tóthin, Knuutilan ja Nevalaisen artikkeli *Reconfiguring flexible machine modules of a PCB assembly* [15], jota tarkastellaan yksityiskohtaisemmin luvussa 6. Artikkelissa esitetään ratkaisu yhdistetylle ongelmalle, jossa halutaan saada selville sekä optimaalinen konekonfiguraatio yksittäisille ladontamoduuleille, että komponenttien allokointi eri ladontamoduuleille siten, että ladontatyö tapahtuu mahdollisimman nopeasti, kun linjastolla valmistetaan yhdentyyppistä piirilevyä (lyh. *MCLB-ongelma*). Tóthin, Knuutilan ja Nevalaisen artikkelissa [15] ongelman ratkaisemiseksi ehdotetaan geneettistä algoritmia, jonka lisäksi työssä on esitelty suorituskykyvertailun mahdollistamiseksi myös neljä muuta ratkaisumenetelmää.

Tässä tutkielmassa MCLB-ongelman ratkaisuksi esitetään metaheuristista menetelmää, jota käsitellään pääluvussa 7. Ehdotettu menetelmä perustuu *simuloituun jäädytykseen*. Ongelmanratkaisussa hyödynnetään mahdollisuuksien mukaan Tóthin ym. [15] artikkelissa esiteltyjä käsitteitä, esimerkkiongelmia ja algoritmeja. Luvussa 7 tehdään aluksi katsaus simuloitun jäädytyksen toimintaan ja käyttökohteisiin. Työssä ehdotettu simuloitua jäädytystä hyödyntävä ratkaisumenetelmä kuvataan yksityiskohtaisesti. Ratkaisumenetelmään kuuluu olennaisena osana systeemin ”jäädyttäminen”, joka vastaa metallin karkaisussa tapahtuvaa hallittua metallin jäädyttämistä. Systeemin

jäähdyttäminen tapahtuu ns. jäähdytysaikataulun avulla, jota tarkastellaan omassa aliluvussa.

Tutkielman viimeisessä osassa testataan simuloitun jäähdytyksen suorituskykyä MCLB-ongelman ratkaisemisessa. MCLB-ongelman ratkaiseva ohjelmakoodi on ohjelmoitu MATLAB-ohjelmointikielellä. Testaamiseen ensimmäisessä osassa tutkitaan jäähdytysaikataulun vaikutusta tarkasteltavana olevan algoritmin suorituskykyyn ottaen huomioon sekä saatavien ratkaisujen laatu, että ohjelman suoritus aika. Ohjelmalla tehtävän alustavan testauksen perusteella valitaan jäähdytysaikataululle sopivat parametrit. Testausta jatketaan vertaamalla simuloituun jäähdytykseen perustuvaan ratkaisumenetelmää artikkelissa [15] esiteltyyn geneettiseen algoritmiin perustuvaan ratkaisumenetelmään. Testejä tehdään kattavasti käyttäen syötteenä vaikeusasteeltaan erilaisia esimerkkiongelmia. Saatuja tuloksia analysoidaan seikkaperäisesti.

## 2 MODULAARINEN PIIRILEVYLADONTAKONE

Modulaariset piirilevyladontakoneet ovat Csaszarin, Nelsonin, Rajbhandarin ja Tirpakin määritelmän mukaan modulaarisia, nopeatoimisia, sisältävät useamman ladonta-aseman (engl. multistation), sekä omaavat askelpalkkityyppisen liukuhihnan (engl. walking-beam conveyor), jolla ladottavia piirilevyjä kuljetetaan ladontapisteeltä seuraavalle [16, s. 408]. Kuten tutkielmassa myöhemmin huomataan, on tämän määritelmän mukaan kuvailtu ladontakone rakenteeltaan varsin yksinkertainen, eikä siis vastaa välttämättä markkinoilla olevien ladontakoneiden rakennetta. Modulaarisen ladontakoneen ehkä tärkeimpään ominaisuuteen, eli sen rekonfiguroinnin tuomiin erilaisiin mahdollisuuksiin paneudutaan tarkemmin seuraavassa pääluvussa, jossa tarkastellaan eri ladontakonevalmistajien ratkaisuja.

Tutkielman johdannossa mainittiin, ettei erilaisia ladontakonetyyppejä ei ole mitenkään standardoitu [9, s. 894], joten on oletettavaa, että eri valmistajien rakentamat modulaariset ladontakoneet eroavat esimerkiksi rakenteellisesti keskenään toisistaan. Tutkielmaa varten on tutkittu eri ladontakonevalmistajien ratkaisuja. Modulaarisia piirilevyladontakoneita löytyy useilta eri valmistajilta. Tutkimalla valmistajien internet-sivuilla saatavilla olevaa markkinointimateriaalia ja konekohtaisia spesifikaatioita, voidaan tarkastella millä eri tavoin modulaarisia ladontakoneita on toteutettu.

### 2.1 Koneen rakenne

Modulaarisista, uudelleenkonfiguroitavista ladontakoneista voidaan muodostaa kokonainen tuotantolinjasto, jolla pystytään kalustamaan tietynlaisia piirilevyjä. Peräkkäin linjastolla olevat moduulit ovat liitettyjä toisiinsa liukuhihnan välityksellä. [10, s. 349], [15, s. 85] Itse ladontamoduulien lisäksi kokoonpanolinjastolla on myös muuntyyppisiä, piirilevyladonnan eri vaiheissa tarvittavia koneita. Valmistettavana oleva, aluksi tyhjä piirilevy syötetään linjaston alkupäähän, josta sitä kuljetetaan liukuhihnan välityksellä jokaiseen linjastolla olevaan koneeseen. Piirilevyjen kokoonpanolinjastolla olevien koneiden avulla suoritetaan komponenttiladonnan viisi päätyövaihetta: liiman tai juotospastan levittäminen piirilevylle, komponenttien asettaminen omille paikoilleen, piirilevyn ja komponenttien lämmittäminen (reflow-

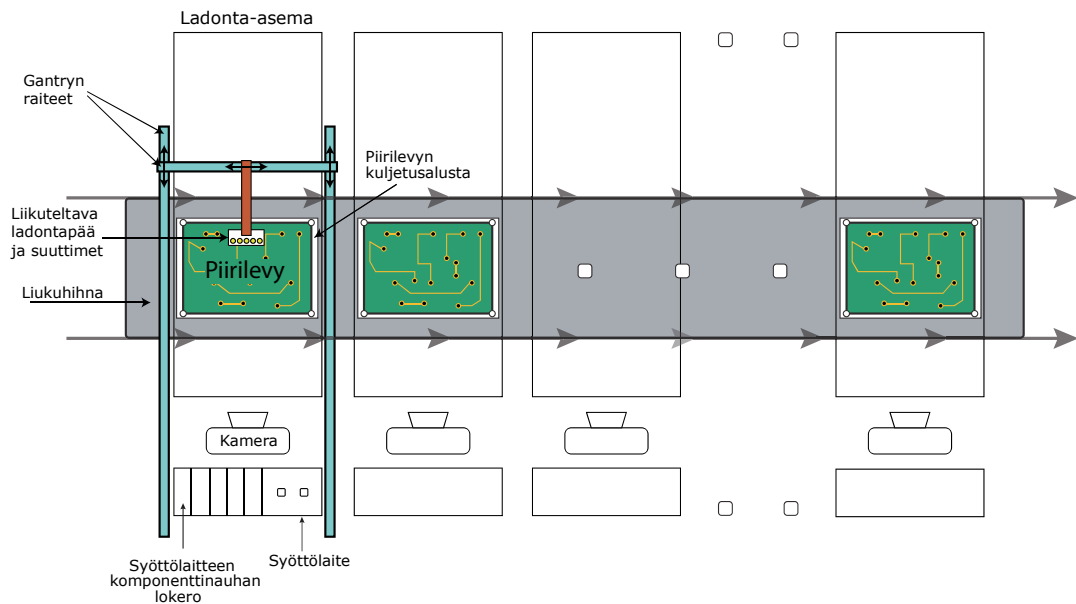
juotos), valmiin piirilevyn puhdistus, sekä lopuksi piirilevyn testaaminen ja tarkastus. Valmis piirilevy poistuu linjaston loppupäästä. [10, s. 349] Tóthin, ym. mukaan ladontamoduulit ovat keskenään identtisiä, mutta tähän löytyy myös poikkeuksia. Esimerkiksi Fujin NXT III-sarjan, sekä Panasonicin NPM-sarjan ladontakoneista on mahdollista rakentaa tuotantolinjasto, jossa on liitetty toisiinsa yhteensopivia, mutta erilaisia moduuleita. [15, s. 85], [14, s. 2], [12, s. 10]

Modulaarinen piirilevyladontakone toimii siten, että liukuhihnan alkupäähän syötetään komponenttien ladontaa odottavat piirilevyt. Liukuhihnalla olevien piirilevyjen kuljetusalustojen (engl. pallet) tehtävänä on pitää piirilevyt tukevasti kiinnitettynä liikkuvalla hihnalla. Liukuhihnaa liikutetaan askel askeleelta eteenpäin siten, että jokainen koneeseen syötetty piirilevy on lopulta kulkenut jokaisen koneessa olevan ladonta-aseman ohitse. Kun piirilevy saapuu liukuhihnan loppuun se voidaan poistaa koneesta, jolloin liukuhihnan alkuun on vapautunut riittävästi tilaa uuden tyhjän piirilevyn syöttämiseen. [16, s. 408]

Ladonta-asemat, joita kutsutaan myös ladontamoduuleiksi (engl. placement module), vastaavat elektronisten komponenttien kiinnittämisestä piirilevylle. Ladonta-asemat voivat toimia samanaikaisesti. Yksinkertaisimillaan ladonta-asema koostuu x- ja y-suunnissa liikkuvasta robottikäsivarresta (engl. robotic arm), jonka päässä on *ladontapä* (engl. placement head) ja yksi tai useampi *suutin* (engl. nozzle). [16, s. 408] Yleisemmin modulaarisissa ladontakoneissa kuitenkin käytetään komponenttien käsittelyyn robottikäsivarren sijasta niinkutsuttua liikkuvaa ristikkopalkkia (engl. *gantry*) [15, s. 85]. Tämän lisäksi jokaisella ladonta-asemalla on omat kiinteästi asennetut elektronisten komponenttien jakelusta vastaavat *syöttölaitteensa* (engl. feeder – syöttölaite, syötinlaite), sekä kamera, jota käytetään komponenttien asennon tunnistamiseen [16, s. 408]. Edellä kuvattu järjestelmä toimii ns. 'Pick-and-Place' -periaatteen mukaisesti, eli siinä komponentit noukitaan syöttölaitteesta ja asennetaan piirilevylle yksi kerrallaan. Mikäli ladonta-asemilla käytetyt ladontapäät ovat monipäisiä ja -suuttimisia, voidaan komponenttien asennus piirilevyille tehdä myös ns. 'Collect-and-Place' -periaatteen mukaisesti. Tällöin syöttölaitteesta voidaan noukkia ladontapään kyytiin useita komponentteja, jotka sitten asennetaan piirilevylle peräjälkeen. [4, s. 1312]

Yksittäisen ladontamoduulin voidaan katsoa kuuluvan Ayobin ja Kendallin luokittelun [9] mukaisesti monipääladontakoneisiin.

Ladontakoneen ohjausyksikkö vastaa ladontakoneen toiminnasta, eli liukuhihnan liikuttamisesta, sekä jokaisen yksittäisen ladonta-aseman toiminnasta. Itse koneen toiminta jakautuu sykleihin, joissa yksittäinen sykli koostuu toiminnoista, jotka tapahtuvat kahden peräkkäisen liukuhihna-askeleen välissä. Syklin alussa koneen ohjausyksikkö jakaa käskyt jokaiselle itsenäisesti toimivalle ladonta-asemalle. Ladonta-asemien käskyt koostuvat komponenttien noukkimisesta syöttölaitteesta, sekä niiden asentamisesta piirilevyille. Kun kaikki ladonta-asemat ovat suorittaneet niille annetut noukkimis- sekä asentamistehtävänsä, voidaan liukuhihnaa siirtää askel eteenpäin, jolloin uusi sykli voidaan aloittaa. Näin ollen piirilevyjen siirto koneelta toiselle tapahtuu synkronisesti. [16, s. 409] Kuvassa 2 on esitetty modulaarisen piirilevyladontakoneen perusrakenne.



Kuva 2: Erään modulaarisen piirilevyladontakoneen rakenne. Kuvan koneessa ladontapäätä liikutetaan (x,y)-suunnissa liikkuvan gantryn avulla. Ladontapäät sisältää useamman suuttimen.

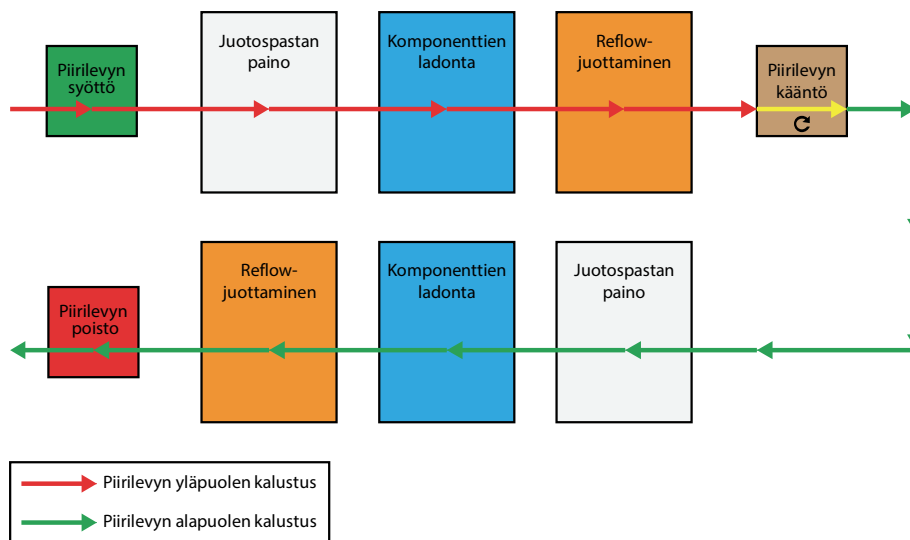
## 2.2 Koneen toiminta

Csaszarin ym. [16] mukaan ladonta-asemalla tapahtuva osien asentaminen piirilevyille koostuu kolmesta askeleesta. Ladontapään suutin poimii syöttölaitteesta komponentin, siirtää sen kameran eteen tarkistusta varten, sekä lopulta siirtää sen piirilevyille ohjausyksikön määräämään kohtaan.

Kallion ym. artikkelissa *Estimating the operation time of flexible surface mount placement machines* [17] tutkitaan eräänntyypin modulaarisen ladontakoneen toiminnan simuloimista, jolloin pystytään tarkasti ennustamaan komponenttien latomiseen kuluva aika. Tarkan simulaation saavuttamiseksi on artikkelissa analysoitu kaikki koneen toimintaan kuuluvat työvaiheet, joilla on vaikutusta ladonnan kokonaisaikaan. Työvaiheet on jaoteltu edellämainitussa artikkelissa kuuteen päävaiheeseen [17, s. 322], joita voimme käyttää apuna esitellessämme koneen toimintaa tarkemmin seuraavissa aliluvuissa. Näiden työvaiheiden lisäksi piirilevyjen ladontalinjastolla on joitain muitakin työvaiheita, joita Kallion ym. artikkelissa ei ole käsitelty. Varsinainen komponenttien ladontatyö koostuu kolmesta työvaiheesta, joihin kuuluvat komponenttien noukkiminen, tunnistaminen sekä asettaminen oikeaan paikkaan piirilevyille.

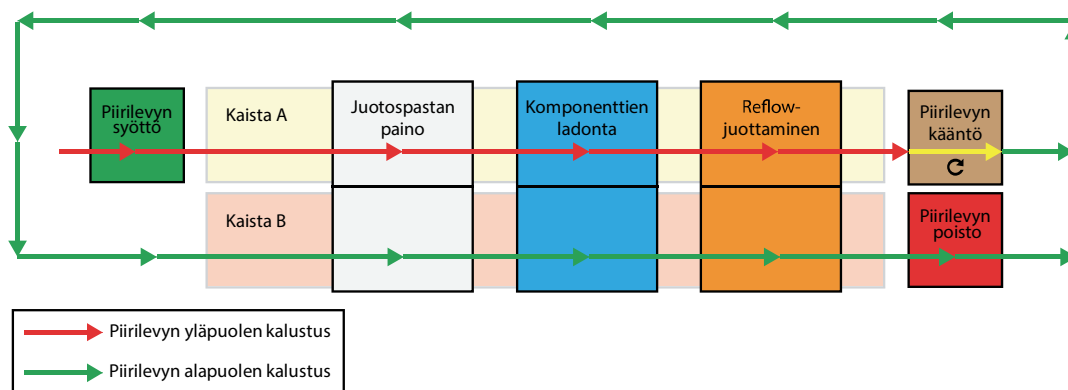
Kaksipuolisten piirilevyjen kalustaminen tapahtuu käytännössä siten, että piirilevyille tehtävät toimenpiteet tehdään kahteen kertaan. Ensin kalustetaan piirilevyn toinen puoli, jonka jälkeen piirilevy käännetään ympäri. Tämän jälkeen voidaan kalustaa toinen puoli piirilevystä. [18] Mikäli kaksipuolisen piirilevyn kalustaminen tapahtuu alusta loppuun yksikaistaisella linjastolla, tarvitaan siis linjastolle kaikkia kalustamiseen liittyviä laitteita vähintään kaksin kappalein. Kuvassa 3 on esimerkki linjastosta, jolla pystytään valmistamaan kaksipuolisia piirilevyjä.





Kuva 3: Kaksipuolisten piirilevyjen kalustaminen yksikaistaisella ladontakonelinjastolla.

Käyttämällä kaksikaistaisia koneita pastanpainoon, ladontaan ja reflow-juottamiseen, voidaan kaksipuoleiset piirilevyt valmistaa myös pienemmällä määrällä koneita (kts. kuva 4). Tämä tapahtuu kuljettamalla ladottavat piirilevyt kääntämisen jälkeen saman ladontalinjaston alkupäähän, mutta toiselle kaistalle. Tällainen ladontalinjasto säästää yksikaistaiseen linjastoon verrattuna sekä sähköä, että tehtaan lattiapinta-alaa. Tämän lisäksi yllä kuvatulla järjestelyllä saavutetaan yksikaistaiseen ladontajärjestelyyn verrattuna täydellinen työmäärän tasapainotus piirilevyn ylä- ja alapuolen ladonnan välillä. Tämä johtuu siitä, että ladontakone käsittelee kahdella kaistalla samanaikaisesti kulkevia piirilevyjä yhtenä virtuaalisena piirilevynä. [19, s. 92].



Kuva 4: Kaksipuolisten piirilevyjen kalustaminen kaksikaistaisella linjastolla. Ladontatyö voidaan tehdä pienemmällä määrällä koneita.

### 2.2.1 Piirilevyn syöttäminen ladontakonelinjastolle

Tehdasolosuhteissa piirilevyjen syöttäminen ladontalinjastolle voidaan tehdä automaattisesti. Tämä tapahtuu käyttämällä piirilevyjen syöttölaitetta. Syöttölaite voi olla esimerkiksi makasiinityyppinen. Tällainen syöttölaite sisältää makasiineja, joista kukin sisältää tietyn määrän latomattomia piirilevyjä. Mikäli syöttölaitteeseen mahtuu esimerkiksi 5 makasiinia ja kuhunkin makasiiniin 50 tyhjää piirilevyä, on syöttölaitteen kokonaiskapasiteetti 250 tyhjää piirilevyä. Makasiinien tyhjentymässä vaihdetaan syöttölaitteeseen uudet, täydet makasiinit, jolloin ladontalinjasto pystyy periaatteessa toimimaan keskeytymättä. Syöttölaite toimii tällöin eräänlaisena puskurina piirilevyjen lisäämisen suhteen. Tarkasteltaessa syöttölaitteen suorituskykyä voidaan tarkastella esimerkiksi piirilevyjen käsittelyyn erikoistuneen Flexlinkin Genius 1-ML -mallista syöttölaitetta, joka pystyy syöttämään uuden piirilevyn ladontalinjastolle 7 sekunnin välein. Saman laitteen makasiininvaihtoaika on n. 20 sekuntia. [20, sivut 7 - 8]

### 2.2.2 Piirilevyn kohdistusmerkkien tunnistaminen

Ladontalinjastolle syötetystä tyhjästä piirilevystä tarkastetaan ensimmäiseksi *kohdistusmerkit* (engl. fiducial marks). Kohdistusmerkkien avulla ladontakone saa tiedon piirilevyn tarkasta sijainnista liukuhihnalla. Piirilevyt eivät ole keskenään välttämättä samanlaisia vaan niiden mittasuhteissa saattaa ilmetä tuotannossa syntyviä poikkeavuuksia. Kohdistusmerkkien avulla ladontakone pystyy kuitenkin kompensoimaan poikkeavuuksista johtuvia epätarkkuuksia hienosäätämällä ladontapään sijaintia komponenttien asennusvaiheessa. Tyypillisesti piirilevyllä on kaksi tai kolme kohdistusmerkkiä. [17, s. 323] Ayobin mukaan kohdistusmerkkejä on tyypillisesti 2 – 4 [9, s. 894]. Piirilevyillä olevat kohdistusmerkit voidaan tehdä siten, että ne ovat piirilevyllä epäsymmetrisesti, jolloin voidaan samalla varmistua, että kone havaitsee ladontalinjastolle väärinpäin syötetyt piirilevyt [18].

### 2.2.3 Juotospasta paino

Piirilevyille kiinnitettävät komponentit juotetaan piirilevyille kiinni juotospasta (engl. solder paste) avulla. Juotospasta sisältää tinaa, seosaineita ja juoksutetta. Juotospasta painetaan piirilevyille metallisen seulan eli stensiilin avulla, jolloin pasta saadaan

annosteltua tarkasti juotostäplien kohdalle. [8, s. 286] Juotospastan koostumus on sellainen, että siihen ladontavaiheessa painetut komponentit pysyvät paikoillaan ennen juotosvaiheessa tapahtuvaa lopullista kiinnitystä [18].

#### 2.2.4 Suuttimien vaihtaminen suutinpankista

Erilaiset piirilevyille aseteltavat komponenttityypit vaativat erilaisten suuttimien käyttöä [11, s. 1189]. Koska suuttimien määrä ladontapäässä on rajallinen, on mahdollista, että ladontapään suuttimia joudutaan vaihtamaan kesken ladontatyön. Ladontakone pystyy vaihtamaan käytössä olevia suuttimia automaattisesti *suutinpankista* (engl. nozzle bank, nozzle station) kesken ladontatyön ilman, että ladontalinjastoa tarvitsee pysäyttää. Tällaisessa tapauksessa ladontakoneet jatkavat toimintaansa keskeytymättä, eli kyseessä on nk. on-line -tyyppinen muutostoimenpide. Käytännössä suuttimenvaihto tapahtuu siten, että ladontapää siirtyy suutinpankin luo, jossa tarpeeton suutin irrotetaan ja asetetaan omalle paikalleen suutinpankkiin. Tämän jälkeen noukitaan tilalle suutinpankista löytyvä tarvittava suutin. Suuttimia voidaan monisuuttimisessa ladontapäässä vaihtaa myös useampia samanaikaisesti. [17, sivut 322, 324] Suuttimien vaihtaminen suutinpankista on ajallisesti kallis operaatio verrattuna muihin ladontaprosessissa tapahtuviin työvaiheisiin [11, s. 1189].

#### 2.2.5 Komponenttien noukkiminen

Ladontapää suorittaa komponenttien asentamisen piirilevyille. Komponenttien noukkiminen asentamista varten tapahtuu siten, että ladontapää siirtyy syötinlaitteelle ja noukkii suuttimen avulla jostain syötinlaitteen komponenttilokerosta ensimmäisen tarvitsemansa komponentin. Tämän jälkeen ladontapää tarvittaessa siirtyy edelleen eri komponentteja sisältävien komponenttilokeroiden välillä ja noukkii niistä muut ladontasyklin aikana mahdollisesti tarvittavat komponentit. [17, s. 323] Tyypillisesti suuttimien kyky noukkia ja kuljettaa komponentteja perustuu alipaineen muodostamaan imuun, joka kohdistetaan suoraan komponenttiin. Alipainetta monitoroimalla voidaan varmistua komponenttien onnistuneesta noukkimisesta ja siitä, etteivät komponentit ole tippuneet ennen niiden ladontaa. [21]

### 2.2.6 Komponenttien tarkastaminen

Kun ladontapää on edellisessä työvaiheessa noukkinut kaikki tarvittavat komponentit, ne pitää vielä tarkastaa kameran avulla ennen asentamista piirilevylle [17, s. 323]. Kameralla tarkastetaan noukitun komponentin tarkka asento ja sijainti suhteessa ladontapähän, jolloin komponentin asennus piirilevylle seuraavassa työvaiheessa on mahdollisimman tarkka. Komponenttien optisessa tarkastuksessa hylätään myös ulkonäön perusteella viallisiksi todetut tai väärinpäin noukitut komponentit. [21] Komponenttien tarkistamiseen voidaan käyttää joko kiinteää tai ladontapään mukana liikkuvaa kameraa. Ladonta-asemaan kiinteästi asennettu kamera sopii lähinnä suurempien komponenttien tarkastamiseen. On huomattava, että kameratyyppi vaikuttaa ladontakoneen nopeuteen, sillä käytettäessä kiinteästi asennettua kameraa tarkastukseen, joutuu ladontapää liikuttamaan komponentit tarkistusta varten kameran luo, jolloin ladontaoperaatio kestää kauemmin. Ladontapähän sijoitettu kamera ei sen sijaan lisää Kallion ym. mukaan ladontaoperaatioon kuluva aikaa. [17, s. 323]

### 2.2.7 Komponenttien asettaminen piirilevylle

Komponenttien asettaminen piirilevylle on varsinaisen ladontaoperaation viimeinen vaihe. Siinä ladontapää liikkuu määrätulle kohdalle piirilevyä ja asettaa komponentin paikalleen [17, s. 324]. Komponentin asennuksessa on oleellista, että komponentin jalat osuvat piirilevyllä oleviin juotospisteisiin. Tällainen tarkkuus saavutetaan laskemalla jokaiselle ladontapäässä olevalle komponentille kameralla tapahtuvan tarkastuksen yhteydessä korjausarvot [21]. Ensimmäisen piirilevylle asetetun komponentin jälkeen ladontapää asettaa kaikki muutkin syöttölaitteesta noukkimansa komponentit omille paikoilleen piirilevyllä. Komponenttien tyypit ja ladontapäässä käytetyt suuttimet vaikuttavat ladontapään liikkumisnopeuteen ja käytettävään asennusvoimaan ladontaoperaation aikana. Kun kaikki yhden ladontaoperaation aikana asennettavat komponentit on sijoitettu piirilevylle, on ladontapää valmis uuteen ladontaoperaatioon, joka alkaa komponenttien noukkimisella syöttölaitteesta (kts. kohta 2.2.5). [17, s. 324]

### 2.2.8 Juottaminen

Piirilevyille ladottujen komponenttien juottaminen piirilevyille tapahtuu ladontalinjastolla olevassa sulatusuunissa (engl. reflow oven). Prosessista voidaan käyttää termiä reflow-juottaminen tai pastajuottaminen. Uunissa lämmitettävänä olevan piirilevyn kuumentuessa juotospasta sulaa muodostaen komponenttien ja juotospisteiden väliset juotokset. Juotoksen jälkeen piirilevy jäädytetään. [21]

Kaksipuoleisen piirilevyn reflow-juottaminen aiheuttaa myös aiemmin juotettujen komponenttien juotosten uudelleensulamisen. Koska aiemmin juotettu puoli piirilevystä on ylösalaisin, on piirilevyn suunnittelussa ja valmistuksessa otettava huomioon, etteivät komponentit pääse irtoamaan juotosten sulamisen yhteydessä. Pienemmät piirilevyillä olevat komponentit pysyvät paikoillaan pelkän tinan pintajännityksen ansiosta. Mikäli piirilevyn alapuolella on painavampia komponentteja, joudutaan niiden kiinnitys piirilevyille varmistamaan esimerkiksi liimalla. [18]

### 2.2.9 Laadunvalvonta

Laadunvalvonta on oleellinen osa piirilevyn valmistusprosessia. Esimerkiksi puuttuvat, sekä vinoon tai väärinpäin asennetut komponentit ovat kriittisiä valmistusvirheitä, jotka pitää tunnistaa ennen kuin piirilevy päästetään eteenpäin piirilevyn ladontalinjastolta. Pintaliitosladontalinjaston päätehtävät, eli pastanpaino, ladonta ja juottaminen määrittelevät myös tärkeimmät tarkastusten kohteet. [22, s. 535] Piirilevyille annosteltava juotospasta tarkastetaan heti pastanpainon jälkeen [23]. Komponenttien ladonta tarkistetaan varmuuden vuoksi kahteen kertaan. Ladonnan tarkastus tehdään yleensä ensimmäisen kerran välittömästi ladontavaiheen jälkeen ja tämän jälkeen uudelleen vielä piirilevyn reflow-juottamisen jälkeen. [22, sivut 535 - 536]

Ladontalinjastoilla käytettävät tarkastuslaitteet toimivat automaattisesti ja niillä pystytään havaitsemaan piirilevyjen ladonnassa yleisimmin ilmeneviä ongelmia. Ladontalinjaston tarkastuslaitteet pystyvät tarkastamaan reaaliaikaisesti juotostahnan oikean annostelun (SPI – Solder Paste Inspection) ja oikein asennetut komponentit (AOI – Automated Optical Inspection). [24] SPI- ja AOI- tarkastus tapahtuu optisesti, eli tarkasteltavana olevista piirilevyistä otetaan kameroiden avulla kuvia, joita analysoimalla pystytään

havaitsemaan piirilevyillä olevat viat. Usean kameran avulla luoduista kuvista pystytään luomaan kolmiulotteisia malleja, jolloin pystytään arvioimaan tarkasti muun muassa annostellun juotospastan määrää. [23], [25] Röntgensäteiden avulla toimivalla AXI-tarkastuslaitteella (Automated X-ray Inspection) tehdään piirilevyn läpivalaisu, jolloin voidaan havaita myös vikoja, joihin AOI-laitteet eivät pysty. AXI-laitteella voidaan esimerkiksi tarkistaa, että komponentin alla piilossa olevat juotospinnat ovat oikein juotettu. [26]

#### 2.2.10 Valmiiden piirilevyjen kerääminen ladontalinjastolta

Tehdasolosuhteissa valmiiden piirilevyjen kerääminen ladontalinjastolta ei tapahdu käsin, vaan käyttämällä automaattista keräilylaitetta. Keräilylaite voi perustua vaihdettaviin makasiineihin samaan tapaan kuin piirilevyjen syöttölaite (kts. 2.2.1). Tällöin linjastolta saapuvat piirilevyt kerätään laitteen sisällä olevaan makasiiniin. Mikäli keräilylaite sisältää useita makasiineja, voidaan täysiä makasiineja vaihtaa tyhjiin, jolloin ladontalinjaston toimintaan ei kohdistu valmiiden piirilevyjen keräilystä johtuvaa viivettä. Useita makasiineja sisältävä keräilylaite mahdollistaa toisin sanoen puskurointiominaisuuden ladontalinjaston loppupäähän. FlexLinkin useampimakasiiniset keräilylaitteet pystyvät keräämään myös viallisia piirilevyjä omaan makasiiniinsa, mikäli viallisten piirilevyjen keräys sijoittuu linjaston loppupäähän. Tällöin kerääjä on saanut tiedon poiskerättävistä piirilevyistä SPI- ja AOI- tarkastuksia tekeviltä laitteilta. [20, sivut 47 - 48]

### **3 MODULAARISEN LADONTAKONEEN REKONFIGUROINTI**

Tässä luvussa tarkastellaan erilaisia tapoja uudelleenkonfiguroida modulaarisia ladontakoneita. Erilaisia keinoja ladontakoneiden uudelleenkonfigurointiin ovat muun muassa päiden vaihtaminen, liukuhihnatyypin muutokset sekä syöttölaitteisiin tehtävät muutokset. Tietoa modulaaristen ladontakoneiden uudelleenkonfiguroimis- mahdollisuuksista on mahdollista etsiä esimerkiksi tutkimalla eri ladontakonevalmistajien julkaisemia markkinointimateriaaleja, käyttöohjeita tai teknisiä spesifikaatioita.

#### **3.1 Päiden uudelleenkonfigurointi**

Modulaarisessa ladontakoneessa olevia päitä voidaan vaihtaa tarpeen mukaisesti. Ladontakoneen päässä oleva suutin tai suuttimet määrittävät, minkälaisia komponentteja piirilevyille pystytään sen avulla asentamaan. Esimerkiksi Panasonicin NPM-sarjan modulaarisissa ladontakoneissa voidaan valita neljästä eri ladontapästä, joissa on joko 3, 8, 12 tai 16 suutinta. Ladontapäissä olevien suutinten määrän kasvaessa nousee myös komponenttien asennusnopeus. Tämä johtuu siitä, että yhdellä kertaa voidaan asentaa suurempi määrä komponentteja ilman, että ladontapään tarvitsee siirtyä noutamaan lisää komponentteja syöttölaitteesta. Tarkasteltavana olevassa Panasonicsissa ladontapään valinta vaikuttaa myös siihen, minkä kokoisia komponentteja sillä voidaan asettaa piirilevyille. Suurimmat ja eniten asennusvoimaa vaativat komponentit voidaan asentaa ladontapäällä, jossa on vähän isoja suuttimia, kun taas pienimpien ja eniten tarkkuutta vaativien komponenttien asennusta varten käytetään päätä, jossa on paljon pienempiä suuttimia. [12, s. 4]

Ladontapäiden uudelleenkonfigurointiin liittyvät myös oleellisesti suutinpankit, jotka sisältävät lajitelman kulloisessakin ladontatyössä tarvittavia suuttimia [12, s. 4]. Ladontatyön aikana tapahtuva suuttimien automaattinen vaihto tapahtuu siten, että ladontapää siirtyy suutinpankille, asettaa kyseisellä hetkellä tarpeettoman suuttimen suutinpankissa olevaan tyhjään paikkaan ja noukkii sen jälkeen työssä tarvittun suuttimen suutinpankista [17, sivut 322, 324]. Suutinpankki laajentaa ladontatyössä käytettävien suutinten valikoimaa, sillä ladontapähän samanaikaisesti mahtuvien suutinten määrä on

rajallinen. Esimerkiksi Panasonicin NPM-sarjan laitteissa ladontapään suutinkapasiteetti on 3 – 16 riippuen käytössä olevasta ladontapäätästä. Itse suutinpankki suuttimiseen on nopeasti vaihdettavissa ladontakoneen rekonfiguroinnin yhteydessä. [12, s. 4]

Ladontapää voidaan Panasonicin NPM-sarjan koneissa vaihtaa myös annostelupääksi (engl. dispensing head) tai tarkastuspääksi (engl. inspection head). Annostelupäätä voidaan käyttää joko liiman tai juotosaineen annosteluun piirilevyille. Piirilevyille asennettavat komponentit pitää juottamisen lisäksi kiinnittää liimalla esimerkiksi silloin, kun käytettävät komponentit ovat erityisen suuria, tai silloin kun käytetään kaksipuolisia piirilevyjä. [12, s. 4]

Ladontapään vaihtaminen tarkastuspääksi mahdollistaa komponenttien oikean ladonnan ja juotosten laadun tarkastamisen. Ladontaprosessin aikana asennettu komponentti saattaa olla esimerkiksi siirtynyt oikealta paikaltaan, olla väärin päin tai jopa puuttua kokonaan. Juotoksissa ongelmia saattaa olla esimerkiksi juotosaineen pursuaminen, huonot liitokset tai ns. tinasillat. [12, s. 5]

### **3.2 Liukuhihnatyypin rekonfigurointi**

Panasonicin NPM-sarjan ladontakoneissa voidaan vaihtaa tuotannon tarpeista riippuen käytössä olevaa liukuhihnatyyppejä. Fyysisesti koneessa on 2 liukuhihnaa, mutta ladonta voidaan suorittaa käyttäen joko yhtä tai kahta kaistaa (engl. single-lane board handling, dual-lane board handling). Mikäli ladonta suoritetaan käyttäen yhtä kaistaa, tarkoittaa se sitä, että koneen molemmat liukuhihnat kulkevat samaan tahtiin. Käytännössä kaksi samalla tavalla liikkuvaa liukuhihnaa voidaan siis ajatella yhdeksi suuremmaksi liukuhihnaksi. Tämä mahdollistaa suurempien piirilevyjen kalustamisen ladontakoneessa. Panasonicin NPM-sarjan ladontakoneet sisältävät kaksi ladontapäätä. [12, s. 8]

Kun ladontakonetta käytetään kaksikaistaisena, molemmilla liukuhihnoilla kuljetetaan erillisiä piirilevyjä. Panasonicin NPM-sarjan koneet mahdollistavat kaksikaistakäytössä erilaisia moodeja, joiden avulla piirilevyjen kalustamisesta vastaavien ladonta-asemien työtä voidaan jakaa mahdollisimman suuren tehokkuuden saavuttamiseksi. Jaetussa



tilassa (engl. shared-mode) ladontakone toimii siten, että molemmat ladontapäät latovat molempien kaistojen piirilevyjä. [12, s. 8] Kun piirilevyä liikutetaan liukuhihnalla syklien välissä (kts. luku 2.1), ei siihen voida latoa komponentteja [16, s. 409]. Jaetussa tilassa toimivassa ladontakoneessa voidaan piirilevyjen liikuttaminen järjestää siten, että toisen piirilevyn liikkua omalla kaistallaan, voidaan toista paikallaan olevaa piirilevyä latoa. Liukuhihnojen liikuttaminen ei siis käytännössä viivästyä ladontakoneen toimintaa. Syöttölaitteiden lokeroiden käyttöaste on jaetussa tilassa Panasonicin mukaan maksimoitu. [12, s. 8]

Itsenäisessä tilassa (engl. independent-mode) kumpikin ladontapäät lataa komponentteja vain lähimpänä olevan kaistan läpi kulkeviin piirilevyihin. Tässä tilassa toimiva ladontakone toimii siis käytännössä kuten kaksi erillistä, toisistaan riippumatonta ladontalinjastoa. Panasonicin mukaan tässä tilassa ladontakoneen suoritusteho on suurimmillaan. Koska kaistat toimivat toisistaan riippumatta, voidaan toiseen kaistaan tehdä muutoksia ilman, että koko ladontakonetta tarvitsee pysäyttää. Tämä tarkoittaa, että voidaan vaihtaa esimerkiksi ladontamoduulin komponenttisyöttimet sisältävä vaunu tai jopa uudelleenkonfiguroida koko kaista toisenlaista tuotetta varten niin sanotusti lennossa (engl. on-the-fly changeover). [12, s. 8]

Näiden kahden moodin lisäksi on mahdollista käyttää ladontakonetta ns. hybriditilassa (engl. hybrid-mode), jolloin ladontalinjastossa on sekä jaetussa, että itsenäisessä tilassa toimivia ladontamoduuleita. Ladontalinjaston käyttäminen tällaisessa moodissa maksimoi Panasonicin mukaan tehokkaasti sen käyttöasteen ja toisaalta minimoi tarvittavan pääoman ladontakoneiden hankinnassa. [12, s. 8]

### **3.3 Syöttimien vaihto**

Modulaarisessa ladontakoneessa voidaan vaihtaa myös komponenttien syöttölaitteita. Komponenttien pakkaustapa vaikuttaa siihen, millaisilla syöttölaitteilla niitä voidaan koneen ladottavaksi syöttää. Panasonicin NPM-sarjan ladontakoneissa on mahdollista käyttää komponentteja, jotka on pakattu keloilla oleviin nauhoihin (engl. tape and reel), tikkuihin (engl. stick) tai ns. paletille (engl. tray). [12, s. 6]

Panasonicin NPM-sarjan ladontakoneissa nauha- ja tikkusyöttimet asennetaan syötinvaunuihin ja paletit ns. tray-vaihtajaan (engl. tray tower). Nauhasyöttimiä on erilaisia riippuen käytettävän komponenttinauhan leveydestä. Syötinvaunut ja tray-vaihtajat ovat liikuteltavia vaunuja, jotka voidaan vaihtaa kokonaan uuteen vaunuun, jossa on valmiina tarvittavat komponentit ja niiden syöttimet. Mahdollisuus vaihtaa tällä tavoin koko syötinvaunu kerrallaan nopeuttaa ladontalinjaston uudelleenkonfigurointia, kun ryhdytään valmistamaan toisenlaista tuotetta. Syötinvaunu voidaan myös vaihtaa silloin, mikäli kesken tuotannon tulee tarve lisätä komponentteja ladontakoneeseen. Komponenttien lisäys voidaan tehdä myös hot-swapping -tyylisesti vaihtamalla pelkästään syötinvaunussa olevia yksittäisiä syöttimiä. Tällöin ladontalinjaa ei tarvitse välttämättä pysäyttää. [12, s. 6]

### **3.4 Gantryjen siirto ja vaihto**

Siplacen SX -sarjan ladontakoneissa on gantryista tehty modulaarisia, eli ne ovat tarvittaessa irrotettavia ja toisaalta helposti uudelleen asennettavia. [13, s. 4] Ladontakonevalmistajan mukaan modulaaristen gantryjen avulla voidaan esimerkiksi siirtää tuotantokapasiteettia helposti kahden eri ladontalinjaston välillä. Gantryjen modulaarisuus mahdollistaa myös sen, että ladontakonevalmistajalta voi ostaa ladontasemamoduuleja ilman gantryja. Tällöin asiakas voi pienemmällä investoinnilla hankkia ladontalinjastolle uuden gantryttoman perusladontamoduulin, johon gantryn voi lisätä esimerkiksi toiselta linjastolta, josta on varaa pienentää tuotantokapasiteettia. [13, s. 5] Gantryjen irrottaminen ladontamoduulista sekä asentaminen ladontamoduuliin kestävät valmistajan mukaan alle 30 minuuttia. Gantryja on myös mahdollista vuokrata, jos valmistuskapasiteettia tarvitsee esimerkiksi kasvattaa vain tilapäisesti [13, s. 4]

## 4 LADONTAKONELINJASTON KÄSITTELYMODUULIT

Piirilevyjen käsittelyyn erikoistuneilla laitevalmistajilla on monipuolinen valikoima erilaisia laitteita, joilla piirilevyjä kuljetetaan ja käsitellään ladontakonelinjastolla. Kuljettamiseen tarkoitettavat laitemoduulit toimivat liukuhihnojen avulla. Tällaisten, liukuhihnan sisältävien laitteiden tarkoitus on kuljettaa ladottavia piirilevyjä varsinaisten ladontatyötä tekevien moduulien välillä. Kuljettamisen lisäksi piirilevyjen käsittelylaitteisiin kuuluvat myös esimerkiksi moduulit, joilla piirilevyjä voidaan puskuroida linjastoilla.

Piirilevyjen käsittelylaitteilla on mahdollista rakentaa monimutkaisia ladontakonelinjastoja, sillä osa niistä mahdollistaa piirilevyjen kuljettamisen valmistusvaiheiden välillä usean eri reittivaihtoehdon välillä. Tässä luvussa tarkastellaan erilaisia käsittelylaitteita ja pohditaan, minkälaisia ladontakonelinjastokokonaisuuksia niillä on teoriassa mahdollista rakentaa. Esimerkiksi aliluvussa 5.6 esitellyn sivuttaissiirtomodulin tai aliluvussa 5.7 esitellyn risteysmodulin avulla yhdellä linjastolla kulkevan piirilevyn voi reitittää toiselle linjastolle. Pelkästään tämäntyyppisiä moduuleita yhdistelemällä linjastoista voidaan ainakin teoriassa rakentaa todella monimutkaisia ladontakonelinjastoja. Teoksessa *Production planning problems in printed circuit board assembly*, Crama, van de Klundert ja Spieksma [27] kutsuvat reitityksen mahdollistavaa ladontakonelinjastoa joustavaksi soluksi (engl. flexible cell). Vain yhden reitin omaavaa linjastoa kutsutaan yksinkertaisesti ladontalinjaksi (engl. line).

FlexLink on eräs piirilevyjen käsittelylaitteistojen valmistaja. Luvussa tarkasteltavat käsittelylaitteistotyypit perustuvat valmistajan tuote-esitteeseen [20]. Tuote-esitteessä kuvatut moduulit ovat saatavana joko yksi- tai kaksikaistaisia. Moduulit toimivat itsenäisesti toisistaan riippumatta. Käyttämällä erillistä korkeamman tason ohjausjärjestelmää, moduulien muodostamaa liukuhihnastokokonaisuutta pystytään valvomaan ja kontrolloimaan. Ohjausjärjestelmän käyttäminen mahdollistaa esimerkiksi piirilevyjen reitittämisen risteyskohtia sisältävien liukuhihnastojen lävitse [20, s. 2].

#### **4.1 Ladontakonelinjaston alku- ja loppupää**

Kuten jo aiemmin on käsitelty aliluvussa 3.2, on piirilevyjen automaattiseen syöttämiseen ja poistamiseen ladontalinjastolta omat piirilevyjä käsittelevät laitteensa. Makasiinityyppisten syöttö- ja keräilylaitteiden lisäksi on käytössä myös laitteita, joissa piirilevyt syötetään linjastolle ja kerätään linjastoilta pinoissa, ilman varastoimista makasiineihin. [20, sivut 3 - 12, 41-50]

#### **4.2 Liukuhihnamoduulit**

Liukuhihnamoduulien päätehtävänä on kuljettaa ladottavana olevia piirilevyjä linjaston eri moduulien välillä. Moduulien avulla liukuhihnaa voidaan myös pidentää, mikäli halutaan yksinkertainen, mutta tilaa vievä keino puskuroida linjastolla kuljetettavia piirilevyjä. [20, s. 13]

#### **4.3 Hylättyjen piirilevyjen tarkastusasema**

AOI-, SPI- tai AXI-tarkastuksessa vialliseksi liputetut piirilevyt voidaan asettaa syrjään erityisellä tarkastusasemalla. Tällainen tarkastusasema (esimerkiksi FlexLink GENIUS 1-RJ) mahdollistaa vialliseksi ilmoitettujen piirilevyjen silmämääräisen tarkastamisen ilman, että koko linjastoa tarvitsee pysäyttää. [20, sivut 17 - 18]

#### **4.4 Puskurimoduulit keskellä linjastoa**

LIFO tai FIFO -periaatteella sekä makasiineja hyödyntävät puskurimoduulit mahdollistavat nimensä mukaisesti keskeneräisten piirilevyjen väliaikaisen varastoinnin keskellä ladontalinjastoa. Puskurimoduuliin varastoidut piirilevyt tekevät tilaa ruuhkautuneelle ladontalinjastolle. Täten kyseisen linjaston alkupää pystyy edelleen jatkamaan töitä uusien piirilevyjen kanssa, vaikka ladontalinjaston loppupää olisikin jostain syystä hidastunut. [20, sivut 19-22]

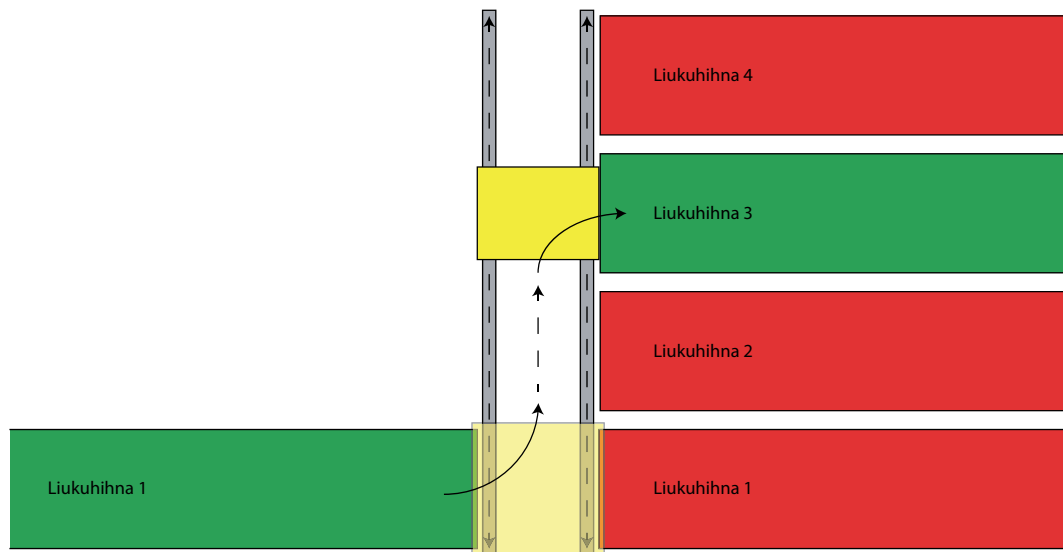
#### **4.5 Ladontalinjaston aukkomoduulit**

Tehdasympäristöön rakennetut monimutkaiset linjastokokonaisuudet saattavat vaikeuttaa tuotantoalueella työntekijöiden kulkua. Tämän vuoksi on kehitetty erityisiä aukkomoduuleita, joilla on mahdollista muodostaa tilapäinen käytävä piirilevyjä kuljettavan liukuhihnareitin poikki. Aukkomoduulit voivat toimia esimerkiksi sukkula-,

tai teleskooppiperiaatteella. Sukkulatyypisessä aukkomoduulissa piirilevy siirretään eteenpäin linjastolla siirtämällä moottorin avulla koko liukuhihnaa laitteen ulkopuolelle. Täten aukkomoduuli on vuorotellen auki ja kiinni. Teleskooppityyppisessä moduulissa kulkuväylä on oletusarvoisesti kiinni -asennossa, jolloin moduuli toimii normaalin liukuhinamoduulin tapaan. Kun käyttäjä haluaa kulkea linjaston läpi, painetaan nappia, joka pysäyttää liukuhinnan ja vetää teleskooppisen liukuhinnan suljettu -asentoon. [20, sivut 31-34]

#### 4.6 Sivuttaissiirtomoduulit

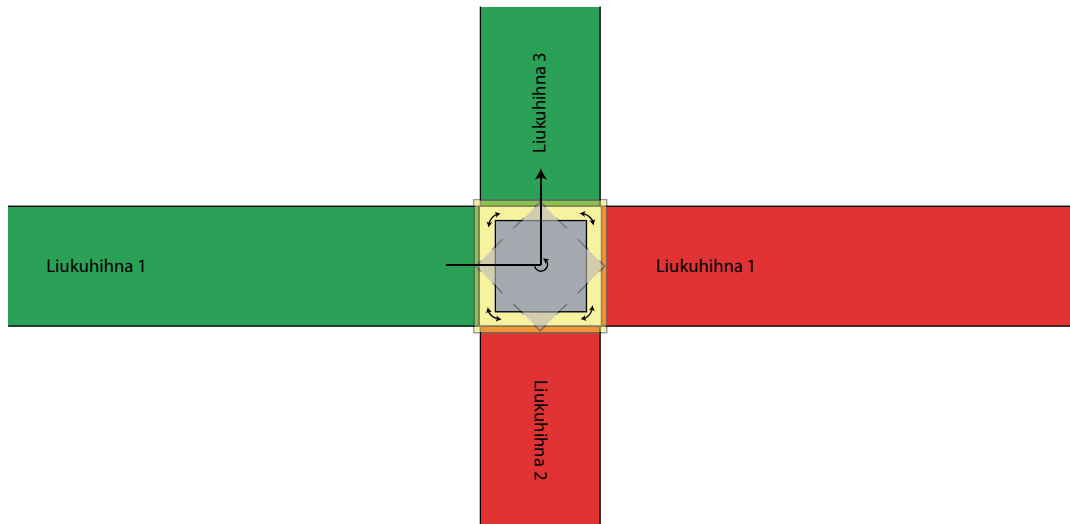
Sivuttaissiirtomoduuleilla (engl. side shuttle) piirilevyjä voidaan siirtää kulkusuuntaan nähden sivuttaissuunnassa (kts. kuva 5). Tällä tavoin kuljetettava piirilevy voidaan siirtää toiselle samaan suuntaan kulkevalle liukuhihnalle. Esimerkiksi FlexLinkin Genius 1-SH pystyy liikuttamaan liukuhihnaa sivusuunnassa jopa 4 metriä, joka mahdollistaa siirtämisen usean eri vaihtoehtoisen liukuhihnareitin välillä. [20, sivut 35 - 36]



Kuva 5: Sivuttaissiirtomoduulin toiminta yläpuolelta kuvattuna. Esimerkkikuvassa liukuhihnalta 1 saapuva piirilevy asetetaan kulkemaan liukuhihnalle 3.

#### 4.7 Risteysmoduulit

Risteysmoduulit (engl. turn unit) mahdollistavat risteysten muodostamisen ladontalinjastolle (kts. kuva 6). Risteysmoduuliin saapuva piirilevy voidaan ohjata kulkemaan menosuuntaan nähden joko vasemmalle, suoraan tai oikealle. Risteysmoduulia voidaan hyödyntää esimerkiksi siten, että vialliseksi liputetut piirilevyt ohjataan pois ladontalinjaston pääliukuhihnalta esimerkiksi poiskeräämistä varten. [20]



Kuva 6: Risteysmoduulin toiminta yläpuolelta kuvattuna. Esimerkkikuvassa liukuhihnalta 1 saapuva piirilevy käännetään kulkemaan liukuhihnalle 3.

#### 4.8 Kääntömoduuli

Piirilevyn kääntäminen tapahtuu kääntömoduulin (engl. inverter) avulla. Kääntämistä tarvitaan linjastolla esimerkiksi silloin, kun kaksipuoleisen piirilevyn kalustamisessa siirrytään yläpuolen kalustamisesta alapuolen kalustamiseen. [20, sivut 39 - 40]

## 5 PIIRILEVYLADONTAKONEIDEN OPTIMOINTIONGELMAT

Ladontakoneiden optimointi on aiheena laaja kokonaisuus, jossa optimointimenetelmät määräytyvät pääasiassa käytössä olevan ladontakonelaitteiston mukaan. Eri tyyppisten ladontakoneiden ominaispiirteitä ja niiden optimointiin liittyviä erityispiirteitä on käsitelty muun muassa Ayobin ja Kendallin teoksessa *A survey of surface mount device placement machine optimisation: Machine classification* [9].

### 5.1 Valmistuksen osaongelmat

Automaattisten piirilevyladontakoneiden optimointi ja piirilevyjen valmistuksen suunnittelu voidaan jakaa tuotantoympäristön erityispiirteet huomioiden pienempiin osaongelmiin. Crama ym. ovat listanneet teoksessa [27] 8 eri osaongelmaa, joihin kuuluvat:

1. ladottavien piirilevytyyppien määrääminen tuoteperheisiin ja tuoteperheiden määrääminen ladontakoneryhmille (ladontalinjoille tai joustaville soluille),
2. syöttimien ja komponenttien allokointi ladontakoneille (moduuleille),
3. piirilevytyyppikohtainen ladontojen jaottelu kullekin ladontakoneelle (moduulille),
4. töidenjärjestely kullekin ladontakoneryhmälle, eli ladottavien piirilevyjen ladontajärjestys,
5. syöttimien järjestys syötinyksiköissä,
6. komponenttien ladontajärjestyksen määrittäminen kullekin piirilevytyyppi – ladontakone -parille,
7. komponenttihakusuunnitelman määrittäminen kullekin piirilevytyyppi – ladontakone -parille, eli tiedot siitä, mistä syöttimestä kukin komponentti noudetaan
8. liikehallinta kullekin piirilevytyyppi – ladontakone -parille, eli tiedot siitä, mistä komponentit noukitaan ja mihin ne ladotaan

Yllä olevat osaongelmat jakautuvat selvästi eri hierarkiatasoisille. Korkeimmalle hierarkiatasolle kuuluu osaongelma 1, jossa käsitellään optimointia koko

tuotantolaitoksen tasolla. Siinä tulee ottaa huomioon kaikki valmistettavana olevat tuotteet ja käytössä olevat ladontalinjastot. Osaongelmat 2 – 4 sen sijaan käsittelevät optimointia tuoteperhekohtaisesti yksittäisen ladontakonelinjaston tasolla ja osaongelmat 5 – 8 yksittäisen ladontakoneen suhteen. Ladontakonelinjaston tasolla käsiteltävät osaongelmat ovat tämän tutkielman kannalta oleellisimpia.

Edellä kuvatut osaongelmat kietoutuvat toisiinsa voimakkaasti. Osaongelmien ratkaisujärjestys on siis jo itsessään mielenkiintoinen ongelma, johon tutkimus ei ole löytänyt yksiselitteistä ratkaisua. Asiaa käsittelevässä kirjallisuudessa ongelmakokonaisuuteen on esitetty esimerkiksi erilaisia iteratiivisia tai hierarkkisia ratkaisumenetelmiä. Monissa ratkaisumenetelmissä optimointiongelmaa on myös yksinkertaistettu siten, että ongelmakokonaisuudesta ratkaistaan vain osa, esimerkiksi yksi tai muutama osaongelma. Yksinkertaistetuissa ratkaisumenetelmissä optimoinnin ulkopuolelle jätetyille osaongelmille oletetaan löytyvän jo ennalta optimaaliset ratkaisut, joihin varsinaisen optimointiongelman ratkaisu täten mukautuu. [5, s. 556]

## **5.2 Modulaarisiin ladontakonelinjastoihin liittyviä optimointiongelmia**

Yksittäisen ladontakonelinjaston tasolla (Crama ym. [27] hierarkian tasot 2 - 4) tapahtuvaa modulaaristen piirilevy-ladontakoneiden optimointia on käsitelty lukuisissa artikkeleissa. Tóthin, Knuutilan ja Nevalaisen artikkelissa *Reconfiguring flexible machine modules of a PCB assembly* [15] esitetään ratkaisu yhdistetylle ongelmalle, jossa halutaan saada selville sekä optimaalinen konekonfiguraatio yksittäisille ladontamoduuleille, että komponenttien allokointi eri ladontamoduuleille siten, että ladontatyö voidaan tehdä mahdollisimman tehokkaasti, kun linjastolla valmistetaan yhdentyyppistä piirilevyä (lyh. MCLB-ongelma). Kun valmistuslinjastolla valmistetaan vain yhden tyyppistä tuotetta, puhutaan tällöin ns. yhden mallin (engl. single-model) linjastosta. Artikkelin ongelma-asettelussa jokainen moduuli sisältää yhden ladontapään ja kukin ladontapää yhden tai useamman suuttimen. Kyseisessä artikkelissa ongelman ratkaisemiseksi ehdotetaan geneettistä algoritmia. Artikkelia käsitellään yksityiskohtaisesti luvussa 6, jossa käsitellään tarkemmin myös MCLB-ongelmaa.



Csaszar ym. ovat tutkineet modulaarisen piirilevyladontakoneen optimointia julkaisussa *Optimization of Automated High-Speed Modular Placement Machines Using Knowledge-Based Systems* [16]. Kyseinen tutkimus edeltää ajallisesti Tóthin ym. artikkelia [15]. Kyseisessä artikkelissa ladontamoduulit ovat rakenteeltaan Tóthin artikkelissa esitetyjä moduuleita yksinkertaisempia, sillä jokainen moduuli sisältää vain yhden yksisuuttimisen ladontapään. Tutkimus esittää MCLB-ongelmalle ratkaisun, joka perustuu tietämuskannan käyttöön (engl. knowledge-based system). Artikkelissa esitetty tietämuskantaan perustuva ratkaisumenetelmä on pohjimmiltaan asiantuntijajärjestelmä. Se siis pyrkii emuloimaan ladontakoneen manuaaliseen optimointiin erikoistuneen asiantuntijan toimia.

Artikkelissa *An integrated allocation method for the PCB assembly line balancing with nozzle changes* [28], Guo, Takahashi, Morikawa ja Jin laajentavat Tóthin ym. artikkelissa [15] esitettyä MCLB-ongelmaa sallimalla suutinvaihdot. Artikkeleissa esitetyt ladontakoneet ovat rakenteeltaan muuten samanlaisia, mutta Guon ym. artikkelissa kuvailtu kone sisältää vaihtosuuttimet sisältävän suutinpainin. Artikkelissa kuvailtu ladontakone pystyy vaihtamaan ladontapäässä olevia suuttimia lennossa sen mukaan, millaisia komponentteja piirilevylle ladotaan. Ratkaisumenetelmät perustuvat Tóthin ym. artikkelissa esitettyyn geneettiseen algoritmiin. Guon ym. artikkelissa tehdyn testauksen perusteella suutinvaihtojen salliminen osoittautuu tehokkaaksi menetelmäksi linjaston tasapainottamisessa.

Tässä työssä ja esimerkiksi artikkelissa [15] tarkasteltu MCLB-ongelma käsittelee yhden mallin linjastoa. Tóthin, Knuutilan ja Nevalaisen artikkelissa *Machine configuration and workload balancing of modular placement machines in multi-product PCB assembly* [29] MCLB-ongelmaa laajennetaan siten, että ladontalinjastolla pystytään latomaan useamman tyyppistä piirilevyä ilman, että konekonfiguraatiota joudutaan välillä muuttamaan. Tällöin puhutaan ns. sekamallilinjastosta (engl. mixed-model). Valmistuslinjaston pysäyttämistä uudelleenkonfiguroimista varten on yleensä järkevää välttää varsinkin pienempien valmistuserien kohdalla, sillä keskeytykset tuotteiden valmistuksessa heikentävät tuottavuutta. Artikkelissa käsitellystä optimointiongelmaasta käytetään termiä MCLB-M -ongelma.

MCLB-ongelma on käytännössä sen verran haastava, että sen ratkaisumenetelmät ovat perustuneet yleensä heuristisiin menetelmiin. Rongin, Tóthin, Nevalaisen, Knuutilan ja Lahdelman työssä *Modeling the machine configuration and line-balancing problem of a PCB assembly line with modular placement machines* [10] esitetään malli, jolla MCLB-ongelma voidaan ratkaista matemaattisesti. Matemaattisella ratkaisumenetelmällä MCLB-ongelmaan saadaan optimaalinen ratkaisu. Täten kyseistä menetelmää voidaan hyödyntää esimerkiksi heuristisilla menetelmillä saatujen ratkaisujen hyvyden arviointiin. Matemaattinen ratkaisumenetelmä toimii parhaiten ratkaistaessa pienikokoisia ongelmia. Hieman vaativampien ongelmien kanssa artikkelissa esitettyä ratkaisumallia joudutaan käytännössä yksinkertaistamaan. Vaikeiden ongelmien ratkaisemiseen menetelmä ei sovellu, jolloin joudutaan käyttämään heuristisia menetelmiä.

MCLB-ongelman ratkaisuun oleellisesti kuuluvassa linjaston tasapainotusongelmassa pyritään pienentämään pullonkaulamoduulin suoritusaikaa. Suoritusajan arvioimiseksi on kehitetty erilaisia menetelmiä. Täten edellä mainittujen MCLB-ongelmien eri varianttien lisäksi on syytä mainita kaksi lähdeostoa, joissa on tutkittu yksittäisten moduulien suoritusajan mittaamista. Kallion, Johnssonin ja Nevalaisen artikkelissa *Estimating the operation time of flexible surface mount placement machines* [17] sekä Vainion, Pahikkalan, Johnssonin, Nevalaisen ja Knuutilan artikkelissa *Estimating the production time of a PCB assembly job without solving the optimised machine control* [30] esitetään kussakin menetelmät, joilla voidaan suhteellisen tarkasti arvioida yksittäisen ladontamoduulin ladontatyöhön kuluva aikaa.

### **5.3 Yksittäisiin ladontakoneisiin liittyviä optimointiongelmia**

Yksittäisen ladontakoneen optimaaliseen toimintaan liittyvät ohjausongelmat ovat hierarkialtaan matalimman tason (Crama ym. [27] hierarkian tasot 5 - 8) osaongelmia. Näiden osatehtävien optimoinnilla saattaa olla merkittävä vaikutus ladontakoneen kokonaissuorituskykyyn, sillä piirilevyille saatetaan latoa satoja tai jopa tuhansia komponentteja [9, s. 894]. Jos esimerkiksi ladontapään kulkemaa matkaa pystytään

lyhentämään edes vähän yksittäisen komponentin ladonnassa, saadaan tästä merkittävää säästöä ladontapään kulkemaan kokonaismatkaan.

Ayobin ja Kendallin artikkelin *The optimisation of the single surface mount device placement machine in printed circuit board assembly: a survey* mukaan yksittäisen ladontakoneen optimointiongelma koostuu viidestä osaongelmasta. Artikkelissa listatut osaongelmat ovat: liikkeenhallinta (engl. motion control), syötinkokoonpano, komponenttien noukinta- ja ladontaoperaatioiden järjestys, komponenttien haku sekä suutinjärjestely, joista jälkimmäisin on lisäksi Craman ym. [27] esittämiin osaongelmiin. [5, sivut 557 - 565]

Yllämainitut osaongelmat nivoutuvat käytännössä toisiinsa tiiviisti, sillä yhteen ladontakoneen osa-asetukseen tehty muutos vaikuttaa aina myös toisiin osa-asetuksiin. Esimerkiksi muutokset suutinjärjestelyssä (kuten missä järjestyksessä suuttimia vaihdetaan) vaikuttavat komponenttien ladontajärjestykseen ja päinvastoin. Yksittäisen ladontakoneen optimointiongelmia käsittelevässä kirjallisuudessa osaongelmia käsitellään sekä yhdistettyinä ongelmina että ongelmina, jossa tarkastellaan vain yksittäisen osaongelman optimointia. Yksittäisen osaongelman optimointi on siinä mielessä haastavaa, että ongelman muiden osien vaikutuksesta koko ladontakoneen suorituskykyyn voidaan tehdä ainoastaan oletuksia. [5, s. 557]

Artikkelissa *PCB assembly optimization in a single gantry high-speed rotary-head collect-and-place machine* [31] Li ja Yoon esittävät optimointimenetelmän yksittäiselle revolveri-tyyppisen ladontapään omaavalle ladontakoneelle. ANNTS-niminen ratkaisumenetelmä (adaptive nearest neighbor tabu search) perustuu tabuhakuun ja se kykenee löytämään optimaalisen ratkaisun yhdistetysti syötinmäärittelylle, ladontajärjestykselle sekä suutinasettelulle (suutinasettelu sisältää suuttimien valinnan ja niiden optimaalisen järjestyksen ladontapäässä). Optimointimenetelmän tarkoituksena on minimoida ladontapään kulkema matka, joka korreloi vahvasti ladontakoneen käyttämän ladonta-ajan kanssa.

Artikkelissa *The Component Retrieval Problem in Printed Circuit Board Assembly* [32] Crama ym. tutkivat mistä syöttimestä komponentteja kannattaa noukkia sellaisessa tapauksessa, jossa samantyyppisiä komponenttityyppejä on asetettu useampaan syöttimeen. Optimaalinen syötin (eli noukkimissijainti) vaihtelee tällaisessa skenaariossa riippuen siitä missä vaiheessa piirilevyn ladonta on, koska ladottavat komponentit sijaitsevat luonnollisesti eri kohdissa piirilevyä. Työssä esitetään kaksivaiheinen dynaamiseen ohjelmointiin perustuva ratkaisumenetelmä, joka kykenee määrittämään polynomisessa ajassa optimaalisen komponenttien noutopaikan jokaista ladontaoperaatiota kohden Fuji CP -tyyppisellä ladontakoneella. Yleisesti ottaen tämän tyyppiset optimointiongelmat pyrkivät löytämään lyhimmän polun ladontapään liikkeille.

Teoksessa *Organizing the nozzle magazine of a gantry-type PCB assembly machine* Knuutila ym. [11] tutkivat suutinpankissa olevien suuttimien järjestämistä mahdollisimman nopeiden suutinvaihtojen saavuttamiseksi (engl. nozzle magazine management problem, NMM). Ongelman ratkaisemiseksi esitetään kaksi erilaista heuristista menetelmää, joiden suorituskykyä arvioidaan vertailemalla niitä brute-force -tyyppiseen menetelmään sekä suutinpankin satunnaisesti järjestelevään menetelmään. Esitellyistä heuristisista menetelmistä toinen perustuu geneettiseen algoritmiin ja toinen parviälgoritmiin. Geneettistä algoritmia hyödyntävän menetelmän todetaan toimivan lähes optimaalisesti pienikokoisissa ongelmissa. Tämän lisäksi isommissa ongelmissa sama menetelmä päihittää selvästi muut artikkelissa esitellyt menetelmät. [11, sivut 1189, 1194] Knuutilan ym. mukaan suutinpankin järjestyksen optimoinnilla saavutetaan yleisesti 5 - 10 % ja satunnaisesti jopa 20 - 30 % parannus suutinvaihtojen nopeuteen verrattuna järjestä-mättömään suutinpankkiin. [11, s. 1200]

#### **5.4 Tuotannon suunnitteluun liittyvät optimointiongelmat**

Piirilevyjen ladontaan liittyvistä optimointiongelmista hierarkkisesti korkeimmalla tasolla ovat optimointiongelmat, jossa käsitellään piirilevyjen valmistusta koko tuotantolaitoksen näkökulmasta. Tehtävänä on tällöin jaotella valmistettavana olevia piirilevysarjoja käytettävissä oleville ladontakoneryhmille eli ladontalinjastoille siten, että piirilevytuotanto toimii kokonaisuuden kannalta mahdollisimman tehokkaasti. Kyse on siis tuotannon suunnittelusta. Täten ongelmana on yleensä piirilevytyyppien jaottelu

tuoteperheisiin ja edelleen tuoteperheiden allokointi käytettävissä oleville ladontalinjastoille.

Hon ja Jin artikkelissa *PCB assembly line assignment: a genetic algorithm approach* [32] käsitellään erästä korkean tason optimointiongelmaa. Työssä esitetään geneettiseen algoritmiin perustuva ratkaisumenetelmä piirilevytyyppien määräämiseksi käytettävissä oleville ladontalinjoille. Artikkelissa esitetyssä skenaariossa piirilevyjen valmistajalla on käytössään joukko ladontalinjoja, joilla pystytään valmistamaan suuria eriä erityyppisiä piirilevyjä. Ongelmassa oletetaan, että piirilevy pystytään kalustamaan valmiiksi yhdellä ladontalinjastolla ja että kaikilla ladontalinjastoilla pystytään valmistamaan kaikkia valmistettavia piirilevyjä. Hon ja Jin työssä esitetty optimointiongelma eroaa kiinnostavalla tavalla kaikista aiemmissä aliluvuissa esitellyistä optimointiongelmistä, sillä optimoinnin tavoitteena on minimoida ajankäytön sijaan tuotannosta syntyvät kokonaiskulut.

## 6 LADONTAKONEKONFIGURAATION JA LINJASTON TASAPAINOTUKSEN ONGELMA

Modulaaristen ladontakoneiden optimointi koostuu usein kahdesta tiukasti yhteen nivoutuvasta isommasta ongelmasta, itse ladontakonemoduulien konfiguraatiosta sekä näiden moduulien muodostaman linjaston tasapainotuksesta (engl. line balancing). Tästä ongelmasta käytetään myös nimitystä MCLB-ongelma (machine configuration and line balancing). MCLB-ongelmaa on käsitelty muun muassa Tóthin, Knuutilan ja Nevalaisen artikkelissa [15].

### 6.1 Ongelman yleiskuvaus

MCLB-ongelma on vaikea ongelma, sillä siihen liittyy useita muuttujia. Valmistettavana olevaan piirilevyyn ladotaan erityyppisiä komponentteja. Eri komponenttityypit taas vaativat tietyn tyyppisen ladontapään ja suuttimen. Eri ladontapäihin mahtuu eri määrä suuttimia, jonka lisäksi tulee huomioida suuttimien ja ladontapäiden sekä komponenttien ja suuttimien keskinäiset yhteensopivuudet. [15, s. 85]

Tóthin ym. artikkelissa [15] kuvatussa MCLB-ongelmassa ladontakonelinjastolla valmistetaan yhdentyypistä piirilevyä. Linjasto koostuu uudelleenkonfiguroitavista moduuleista. MCLB-ongelman ratkaisemiseksi tulee konfiguroida ladontamoduulit sekä allokoida ladottavat komponentit eri ladontamoduuleille siten, että piirilevyn kalustamiseen kuluva aika on mahdollisimman lyhyt. MCLB-ongelma on luonteeltaan kombinatorinen ja siinä on useita tiukkoja rajoitteita. Käytännössä optimointiongelmat ovat usein suuria, jolloin joudutaan tyytymään alioptimaalisiin (suboptimaalisiin) ratkaisuihin. [15, s. 86]

### 6.2 Ongelman yksityiskohdat ja ratkaisumalli

Tóthin ym. artikkelissa [15] käsitellään ongelmaa, jossa valmistettavana olevalle piirilevyille ladotaan  $N$  kappaletta komponentteja. Tavoitteena on valmistaa erä samanlaisia piirilevyjä mahdollisimman tehokkaasti. Oletetaan, että erilaisia komponenttityyppejä on  $t$  kappaletta ja että kutakin komponenttityyppiä  $i$  ladotaan  $c_i$  kappaletta. Tuotantolinjasto koostuu konfiguroitavista ladontamoduuleista, joita on  $m$

kappaletta. Ladontamoduuleiden lisäksi linjastoon kuuluu luonnollisesti muitakin laitteita (kts. luku 2), mutta koska esimerkki-ongelmassa nimenomaan komponenttien ladontatyö aiheuttaa linjaston pullonkaulan, on tarpeellista keskittyä minimoimaan itse ladontamoduulien kuluttamaa aikaa. [15, s. 86]

Ladontakonmoduuleihin voidaan asentaa esimerkkitapauksessa erilaisia ladontapäitä ja syöttimiä vastaamaan kulloinkin ladottavana olevalle piirilevyille asennettavia komponentteja. Moduulien uudelleenkonfigurointi tapahtuu off-line -tyyppisesti, eli silloin, kun ladottavana oleva piirilevytyyppi vaihtuu ja linjasto pysäytetään. Jokainen ladontamoduuli on perusrakenteeltaan luvussa 3 kuvatun kaltainen. Moduuli sisältää paikallaan olevan alustan piirilevyille, kiinteän syötinyksikön ja yhden (x,y)-suunnissa liikuteltavan gantry-tyyppisen ladontapään, joka sisältää useita suuttimia. Erilaisia ladontapäätyyppisiä (eri suutinkapasiteetein) on  $h$  kappaletta ja erilaisia suutintyyppisiä  $n$  kappaletta. Oletuksena on, että samaa tyyppiä olevien ladontapäiden tai suuttimien määrää ei ole rajoitettu. Suutinasettelu on annetussa esimerkkitapauksessa kiinteä. Se tarkoittaa sitä, että ladontapäähän asennettuja suuttimia ei voida vaihtaa lennossa kesken tuotantoerän valmistumisen. Yksittäisen komponentin asettamiseen piirilevyille kuluva aika ( $th_i$ ) ja komponentin noukkimiseen syöttimestä kuluva aika ( $tp_i$ ) ovat moduulista ( $i$ ) riippuvia vakioita. [15, s. 86]

MCLB-ongelman ratkaisemiseksi tarvitaan tiedot ladontapäiden ja suuttimien, suuttimien ja komponenttien sekä komponenttien ja ladontapäiden yhteensopivuuksista. Yhteensopivuudet ilmoitetaan  $HN$  (ladontapää – suutin -yhteensopivuus) ja  $CN$  (komponenttityyppi – suutintyyppi -yhteensopivuus) -matriiseina. Kyseisistä yhteensopivuusmatriiseista on esitetty esimerkki kuvassa 6.

Matriisien alkioiden arvot ovat täten:

- $HN_{i,j} = 1$ , mikäli ladontapäätyyppi  $i$  ja suutintyyppi  $j$  ovat yhteensopivia,
- $CN_{k,j} = 1$ , mikäli komponenttityyppi  $k$  ja suutintyyppi  $j$  ovat yhteensopivia ja
- $CH_{k,i} = 1$ , mikäli komponenttityyppi  $k$  ja ladontapäätyyppi  $i$  ovat yhteensopivia

Epäyhteensopivuus ilmoitetaan kaikissa matriiseissa alkion arvolla 0.

Yhteensopivuuksia tarkasteltaessa tulee huomata, että:

- yhdentyypinen ladontapää tukee joukkoa erilaisia suuttimia,
- yhdentyypisellä suuttimella voidaan käsitellä useaa erityyppistä komponenttia ja
- komponenttien ja ladontapäiden yhteensopivuus (CH-matriisi) seuraa aina transitiivisesti ladontapää – suutin ja komponentti – suutin -yhteensopivuuksista, joten sitä ei tarvitse erikseen ilmoittaa.

Yhteensopivuuksien lisäksi MCLB-ongelman ratkaisemiseksi tarvitaan seuraavat tiedot:

1. Komponenttityyppien syötinkelan leveys ( $w_i$ ) (mitataan syötinlaitteen lokeroiden lukumääränä)
2. Syötinlaitteen kapasiteetti (mitataan syötinlaitteen lokeroiden lukumääränä)
3. Ladonta-aika ( $th_j$ ) jokaiselle konetyypille  $j$  (ilmaistaan keskimääräisenä ladonta-aikana komponenttia kohti)
4. Noukkimisaika ( $tp_j$ ) jokaiselle konetyypille  $j$  (ilmaistaan keskimääräisenä ladonta-aikana komponenttia kohti)

Optimointitehtävän yksinkertaistamiseksi Tóthin ym. artikkelissa [15] on tehty oletus, että kullekin suutintyypille on olemassa vain yksi yhteensopiva ladontapäätyyppi. Tämän lisäksi jokaista komponenttityyppiä pystytään käsittelemään vain yhdentyypisellä suuttimella. [15, s. 86]

MCLB-ongelman ratkaisun tulee olla toteuttamiskelpoinen (engl. feasible), eli sallittu, joten sen tulee noudattaa seuraavia rajoitteita:

- C1. Jokainen yksittäinen komponenttiladonta on määrätty jollekin ladontamoduulille.
- C2. Jokaiseen ladontapähän on määrätty suuttimia enintään niin monta, kuin ladontapähän on suuttimia mahdollista asentaa.



- C3. Suutinta ei ole määrätty sellaiseen ladontapähän, minkä kanssa se ei ole yhteensopiva.
- C4. Jos komponentti on määrätty tietylle ladontakoneelle, ainakin yksi kyseisen koneen ladontapäissä olevista suuttimista on yhteensopiva edellä mainitun komponentin kanssa.
- C5. Tietylle ladontapäälle määrättyjen ladottavien komponenttien kokonaisleveys ei ylitä kyseiseen ladontapähän liittyvän ladontakoneen syötinyksikön kokonaiskapasiteettia.

Ratkaisu MCLB-ongelmaan koostuu kolmesta päävaiheesta; jokaista ladontalinjastolla olevaa ladontamoduulia kohden tulee selvittää:

1.  $mh_j$  = ladontapäätyyppi, joka on määrätty moduulille  $j$
2.  $x_{i,j}$  = moduulille  $j$  määrättyjen  $i$ -tyyppisten komponenttien lukumäärä
3.  $y_{i,j}$  = moduulin  $j$  ladontapähän määrättyjen  $i$ -tyyppisten suutinten lukumäärä

Ratkaisun löytäminen näille kolmelle määrittelylle ovat toisilleen yhteyksissä, eli niillä on vaikutusta toisiinsa. Parhaan ratkaisun löytämiseksi kaikkia kolmea määrittelyä pitäisikin tarkastella samanaikaisesti, mutta käytännön ongelmissa tehtävästä muodostuu liian monimutkainen. Määrittelyt tehdään tämän vuoksi peräkkäin ja iteratiivisesti. Ladontapään määrittely (valinta) tehdään ensin, koska tämän jälkeen mahdollisten komponenttien ja suutinten määrittely on pääosin kiinnitetty. Määrittelyssä oleva iterointi rajoittuu komponenttien ja suutinten määrittelyyn. MCLB-ongelman ratkaisun kolmea päävaihetta käsitellään tarkemmin seuraavassa aliluvussa. [15, s. 87]

MCLB-ongelmaa ratkaistaessa tavoitteena on saada aikaan kokoonpano, jossa piirilevyjen latominen tapahtuu mahdollisimman nopeasti. Kuten aiemmin on mainittu, piirilevyerän ladonnan nopeuttamiseksi tulisi lyhentää nimenomaan ladontalinjaston pullonkaulamoduulin eli hitaimman ladontakonemoduulin suoritusaikaa. Tóthin ym. artikkelissa [15] on esitelty menetelmä, jossa ladontakonemoduulien keskinäisiä suoritusajkoja voidaan vertailla, kun moduulien kokoonpanosta tiedetään ladottavien

komponenttien lukumäärä, ladontapään tarvitsema noukkimis- ja ladonta-aika sekä tehtävälohkojen lukumäärä. Tehtävälohkot koostuvat neljästä askeleesta:

1. komponenttien noukinta suuttimen avulla,
2. ladontapään siirtäminen kohtaan, johon ensimmäinen komponentti ladotaan,
3. komponenttien latominen oikeille paikoilleen piirilevylle ja
4. ladontapään siirtäminen takaisin syötinyksikön luo

Täten MCLB-ongelman ratkaiseva kohdefunktio voidaan kirjoittaa seuraavasti:

$$\min_{j \in \{1, 2, \dots, m\}} \max \left\{ \left( \sum_{i=1}^t x_{i,j} \times (th_j + tp_j) \right) + a \times b_j \right\},$$

missä  $b_j$  on koneen  $j$  tehtävälohkojen määrä ja  $a$  on tehtävälohkojen aikakerroin (sisältäen ajan, joka kuluu ladontapään liikkumiseen piirilevyn ja syöttimen välillä sekä ajan, joka kuluu komponentin tarkistamiseen kameran avulla). [15, s. 87]

Tehtävälohkojen määrä  $b_j$  voidaan laskea ladontapähän määritettyjen suutinten sekä moduulille määriteltyjen komponenttien lukumäärän perusteella. Jokaisella suutintyyppillä on tietty määrä tehtävälokoja. Tehtävälohkojen määrä saadaan laskemalla yhteen suutintyyppille mahdollisten ladontojen kokonaismäärä, joka jaetaan ladontapähän asennettujen tarkasteltavana olevan suutintyyppin suuttimien määrällä. Täten,  $b_j$  on maksimi tehtävälohkojen lukumäärästä kullekin suutintyyppille, joka ladontapähän on asennettu. Jos oletetaan, että jokaisessa moduulissa komponenttiladonnat on jaettu kaikkien yhteensopivien suuttimien kesken ja että suuttimet on täytetty ahnaasti, saamme:

$$b_j = \max_{\forall i \in N'_j \subseteq N} \left\{ \frac{\sum_{k=1}^t x_{k,j} \times CN_{k,i}}{y_{i,j}} \right\},$$

missä  $N'_j$  on koneen  $j$  ladontapähän määriteltyjen suutintyyppien joukko. [15, s. 88]

### 6.3 Ongelman kolmivaiheinen ratkaisuperiaate

Edellisessä aliluvussa esiteltiin lyhyesti Tóthin ym. artikkelissa oleva MCLB-ongelman ratkaisuperiaate, joka koostuu kolmesta päävaiheesta. Ensimmäisessä vaiheessa tapahtuu ladontapäiden allokointi ladontamoduuleille, toisessa vaiheessa määritellään millä ladontapäällä mikäkin komponentti ladotaan ja kolmannessa vaiheessa määrätään lopulta suuttimet eri ladontapäille. Tämän työn kannalta keskeisin osa MCLB-ongelman ratkaisua on keskimäinen vaihe, jossa etsitään komponentti – ladontapää -määrittely. Tähän ongelmaan esitetään vaihtoehtoinen ratkaisumenetelmä seuraavassa pääluvussa.

#### 6.3.1 Ladontapäiden määrääminen moduuleille

Ladontamoduulien tarvitsemat ladontapäät voidaan määrittellä, kun tiedetään ladontatyössä käytetyt komponenttityypit sekä kunkin komponenttityypin ladontamäärät piirilevyille. Tästä käytetään myöhemmin nimitystä ladontakone – ladontapää -määrittely (engl. *machine – head -configuration*). Ensimmäisessä vaiheessa saatu ladontakone – ladontapää -määrittely kertoo epäsuorasti myös, mitä suuttimia ladontapäille voidaan määrätä ongelmanratkaisun kolmannessa vaiheessa. [15, s. 88]

Ladontakone – ladontapää -määrittelyn etsiminen aloitetaan selvittämällä, kuinka monta ladontapäätyyppiä  $j$  vähintään tarvitaan. Käytännössä tämä tapahtuu ryhmittelemällä komponenttityypit sen mukaan, minkä ladontapäätyyppin kanssa ne ovat yhteensopivia. Ladontapäätyyppien  $j$  minimimäärä (ja täten myös ladontapäätypillä  $j$  varustettujen moduulien lukumäärä) on  $LB_j = \max\{LB_j^1, LB_j^2\}$ , joka sisältää:

#### Suutinvaatimuksen $LB_j^1$

Komponenttityypit, jotka on määrätty ladontapäätypille  $j$  määrittävät suuttimien tyypit  $S$ , jotka asennetaan kyseiseen ladontapähän. Ladontapäiden (tyyppiä  $j$ ) minimimäärä saadaan kaavasta  $\lceil |S|/ht_j \rceil$ .

### Syötinkapasiteetin $LB_j^2$

Koska jokaisessa moduulissa on yksi ladontapää, voidaan laskea tarvittavien moduulien lukumäärä kutakin ladontapäätyyppiä kohden käyttämällä hyväksi tietoa komponenttijoukosta. Tällöin pitää ottaa huomioon syötinlokeroiden määrä, jonka kuhunkin ladontapähän määrättyt komponentit vievät kyseisestä ladontamoduulista ja jakaa saatu tulos kyseisen moduulin syötinyksikön kokonaiskapasiteetilla ( $fc$ ). Tästä saatu tulos pyöristetään lopulta ylöspäin.

Olkoon  $L_j$  ladontapäätyyppi  $j$ :n kanssa yhteensopivien komponenttiladontojen kokonaismäärä jaettuna ladontapäättyypin kapasiteetilla.  $L_j$  siis ilmaisee ladontapäättyypin  $j$ :n tarpeellisuutta. Tämä lasketaan [15] mukaan seuraavasti:

$$L_j = \frac{\sum_{i=1}^t c_i \times CH_{i,j}}{ht_j}$$

Olkoon edelleen:

$$L = L_j - s$$

Moduulien  $m$  ladontapäät jaotellaan alustavasti eri ladontaoperaatioihin siten, että ladontapäät on jaoteltu tasaisesti yhteensopivien ladontaoperaatioiden lukumäärän suhteen. Ladontapäättyypillä  $j$  varustettujen moduulien lukumäärä on täten:

$$k_j = \left\lceil m \times \frac{L_j}{L} \right\rceil$$

Yllä olevassa kaavassa tapahtuva alaspäin pyöristäminen aiheuttaa sen, että jotkut ladontakonemoduuleista saattavat pysyä vapaina, ts. niille ei ole määritelty lainkaan ladontapäätä.

Ottaen huomioon aiemmin mainitut vaatimukset  $LB_j^1$  (suutinvaatimus) ja  $LB_j^2$  (syötinkapasiteetti), josta muodostuu vaatimus (alaraja)  $LB_j$  saadaan tästä kaksi eri tapausta:

1. Vaatimusten toteuttamiseksi moduuleita lisätään, jotta voitaisiin ottaa käyttöön ladontapäättyyppejä, joita ei ole vielä määrätty millekään moduulille. Tämän jälkeen vähennetään iteratiivisesti ja vähitellen ladontapäiden lukumäärää siten, että vaatimus  $LB_j$  edelleen toteutuu. Ladontapäät poistetaan  $k_j$ :n suhteen kasvavassa järjestyksessä, eli siten, että ensin poistetaan niitä ladontapäättyyppejä, joita moduuleille on vähiten määrätty. Jos tällaista ladontapäättyyppejä ei ole saatavilla, ei ongelmaan ole löydettävissä kelvollista ratkaisua.
2. Kaikki vaatimukset on täytetty ja osa moduuleista on vapaana. Tässä tapauksessa moduuleille määrätään iteratiivisesti vähän kerrallaan ladontapäitä  $k_j$ :n suhteen vähenevässä järjestyksessä.

Vaatimuksen  $LB_j$  toteutuminen takaa toteuttamiskelpoisen ratkaisun MCLB-ongelmaan. Suutinvaatimuksen  $LB_j^1$  tyydyttäminen tarkoittaa, että kaikille tarvittaville suutintyypeille on tilaa ladontapäissä; täten jokainen komponenttityyppi voidaan jyvittää ainakin yhdelle ladontapäälle, joka on varustettu yhteensopivalla suuttimella. Syötinkapasiteetin  $LB_j^2$  tyydyttäminen sen sijaan takaa, että syötinyksiköissä on tarpeeksi tilaa jokaista ladontapäättyyppeä varten, jotta komponentit voidaan varastoida konemoduulien syötinyksiköihin. [15, s. 88]

### 6.3.2 Ladottavien komponenttien määrääminen ladontapäille

Kun ladontakone – ladontapää -määrittely on selvitetty edellisessä työvaiheessa, voidaan seuraavaksi määrittellä millä ladontapäällä kukin komponenttiladonta suoritetaan. Koska komponentti – ladontapää -määrittely ja suutin – ladontapää -määrittely vaikuttavat toisiinsa rajoittavasti, täytyy molempia määrittelyitä arvioida yhdessä käyttäen MCLB-ongelman kohdefunktiota (kts. aliluku 6.2). Kun komponentteja jaotellaan ladontapäiden kesken, muodostuu hakuavaruudesta suuri. Tästä syystä ongelman ratkaisemiseen sopii erityisen hyvin jokin heuristinen ratkaisumenetelmä. [15, s. 88]

Tóthin, Knuutilan ja Nevalaisen artikkelissa esitetään eräs heuristinen ratkaisumenetelmä komponenttien määräämiseksi ladontapäille. Se on tyypiltään geneettinen algoritmi, joka perustuu luonnonvalintaan. Geneettinen algoritmi toimii yksinkertaistetusti siten, että parhaat ratkaisuyksilöt valitaan muodostamaan jälkeläisiä uudeksi sukupolveksi. Uusia sukupolvia tuotetaan, kunnes ratkaisuyksilöt eivät enää merkittävästi eroa vanhemmistaan. [15, s. 89]

### 6.3.3 Suuttimien määrääminen ladontapäille

Suuttimet voidaan määrätä ladontapäille, kun edellisten vaiheiden ladontapää - ladontakone -määrittely ja komponentti - ladontapää -määrittely on saatu selvitettyä. Suutin – ladontapää -määrittelyn alustava ratkaisu saadaan valitsemalla ladontapähän *j* yksi kutakin vaadittavaa suutintyyppiä toteuttamiskelpoisen ratkaisun varmistamiseksi. Tämän jälkeen vapaaksi jääneisiin suutinpitimiin jaetaan suuttimet komponenttiryhmiin koon mukaisesti. [15, s. 90]

## 7 LADONTAKONEKONFIGURAATION JA LINJASTON TASAPAINOTUKSEN ONGELMAN RATKAISEMINEN KÄYTTÄEN SIMULOITUA JÄÄHDYTYSTÄ

MCLB-ongelman komponentti - ladontapää -määrittelyn ratkaisuksi on annettu Tóthin, Knuutilan ja Nevalaisen työssä viisi eri ratkaisumenetelmää, jotka ovat brute force, satunnainen, ahnas, paikallinen haku sekä geneettinen algoritmi. Tässä työssä menetelmäkokoelmaa laajennetaan uudella metodilla, joka perustuu simuloituun jäähdytykseen. Työssä esitellyn metodin toteutus pohjautuu yllä mainitussa artikkelissa oleviin ratkaisumetodeihin. Käytännön ohjelmointityö tehtiin MATLAB-ohjelmistolla. MATLAB valittiin ohjelmointikieleksi pääasiassa kahdesta syystä. Ensimmäiseksi, Attila Tóthin tekemä ohjelmointityö edellä mainitussa artikkelissa toteutettuja ratkaisumetodeja varten oli toteutettu kyseisellä ohjelmointikielellä. Toiseksi, kaikki Tóthin, Knuutilan ja Nevalaisen artikkelissa esiteltyjen menetelmien lähdekoodi, sekä niiden testaukseen käytettävä aineisto saatiin hyödynnettäväksi tässä työssä kehitettävien menetelmien ohjelmointia ja testausta varten.

### 7.1 Simuloitu jäähdytys

Simuloitu jäähdytys<sup>1</sup> (engl. Simulated Annealing) on Kirkpatrickin, Gelattin ja Vecchin teoksessa *Optimization by Simulated Annealing* [33] sekä Černýn teoksessa *Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm* [34] esitelty metaheuristinen ja todennäköisyyspohjainen optimointimenetelmä. Menetelmästä käytetään myös muita suomenkielisiä nimityksiä, kuten simuloitu annelointi, simuloitu hehkutus, simuloitu mellotus [35] tai jäähdytysmenetelmä [36]. Menetelmä perustuu pohjimmiltaan paikalliseen hakuun. Menetelmää käytetään yleisesti vaikeiden kombinatoristen optimointiongelmien ratkaisuun. Tästä syystä se sopii erityisen hyvin myös työssä käsiteltävän MCLB-ongelman ratkaisemiseen. [37]

---

<sup>1</sup> Englanninkielinen termi annealing tarkoittaa suomeksi hehkutusta. Täten *simuloitu jäähdytys* ei ole kirjaimellisesti eksakti käännös alkuperäisestä englanninkielisestä termistä *simulated annealing*. Simuloitu jäähdytys on terminä kuitenkin laajasti käytössä, joten sen käyttö tässä tutkielmassa on perusteltua. Lisäksi se kuvaa hyvin algoritmin perusidea. Tutkielmassa käytettyä termiä käytetään muun muassa Turun yliopiston matematiikan ja tilastotieteen laitoksella optimointia käsittelevässä kurssimateriaalissa [33, p. 131].

Simuloitu jäähditys on saanut vaikutteita metallin karkaisusta, jossa metallin jäähtymistä pyritään hidastamaan sen valmistusvaiheessa. Nopea jäähtyminen vastaa käytännössä lokaalia hakua, eli kulkemista jatkuvasti kohti parempaa ratkaisua. Hitaampi jäähtyminen sen sijaan mahdollistaa myös huonempien ratkaisujen hyväksymisen tietyllä todennäköisyydellä, jolloin algoritmi ei juutu niin helposti lokaaliin minimiin tai maksimiin. [38, s. 53]

## 7.2 Algoritmi

Simuloituun jäähdytykseen perustuva MCLB-ongelman ratkaisumenetelmä on käytännössä muunnelma Tóthin, Knuutilan ja Nevalaisen työssä esitellystä lokaaliin hakuun perustuvasta ratkaisumenetelmästä [15, s. 93]. Simuloitua jäähdytystä käyttävän menetelmän pseudokoodi voidaan kirjoittaa täten seuraavasti:

---

SIMANL: Määrää ladottavat komponentit ladontakonemoduuleille käyttäen simuloitua jäähdytystä.

Syöte: Ongelmamäärittely<sup>2</sup>, Ladontakone – ladontapää -määrittely<sup>3</sup>

Tulos: Ratkaisuyksilö<sup>4</sup>

1. Asetetaan systeemin aloituslämpötila  $T$ .
2. Etsitään ongelmaan alustava ratkaisu käyttämällä satunnaismetodia.
3. Generoidaan nykyisestä ratkaisusta uusi ratkaisu seuraavasti:
  - a. Valitaan satunnainen ladontapää ( $j$ ).
  - b. Valitaan satunnainen komponenttityyppi ( $i$ ), joka on määrätty ladontapäälle ( $j$ ).
  - c. Generoidaan satunnaisluku  $r \in [0, x_{i,j}]$ , missä  $x_{i,j}$  on tyyppiä  $i$  ladontapäälle  $j$  määrättyjen komponenttiladontojen lukumäärä.

---

<sup>2</sup> Ongelmamäärittely sisältää tiedot ladottavana olevista komponenteista ja niiden määristä, sekä käytössä olevasta ladontakalustosta. Kuvassa 7 esitetään ongelmamäärittelyn sisältämät tiedot.

<sup>3</sup> Ladontakone – ladontapää -määrittely (MH), ratkaistaan MCLB-ongelman ratkaisun ensimmäisessä vaiheessa (kts. aliluku 6.3.1).

<sup>4</sup> Ratkaisuyksilö sisältää tiedon siitä, mille ladontamoduulille kukin komponenttiladonta on määrätty. Tämän lisäksi ratkaisuyksilö sisältää tiedon ladontapäille määrättyistä suuttimista. Ratkaisuyksilö on fitness-arvoltaan paras ohjelman suoritusaikana löydetty yksilö.



- d. Valitaan satunnaisesti toinen kelvollinen ladontapää ( $j'$ ), johon voidaan siirtää komponenttiladontoja. Ladontapää on kelvollinen, jos se ei riko rajoitteita C4 ja C5 (kts. aliluku 6.2).
  - e. Siirretään  $r$  määrä komponentteja ladontapäästä  $j$  ladontapäälle  $j'$ .
4. Hyväksymisfunktion avulla määritetään, valitaanko saatu ratkaisu uudeksi ratkaisuksi:
- a. Lasketaan todennäköisyys ( $P$ ) sille, että saatu ratkaisu hyväksytään uudeksi ratkaisuksi seuraavasti:
    - i. Jos uuden ratkaisun energia-arvo ( $e'$ ) (=fitness) on pienempi kuin nykyisen ratkaisun energia-arvo ( $e$ ), asetetaan  $P = 1$ .
    - ii. Muutoin,  $P = \exp\left(-\frac{e'-e}{T}\right)$ , missä  $T$  on systeemin lämpötila.
  - b. Generoidaan satunnaisluku  $r \in [0, 1]$ .
  - c. Jos  $P \geq r$ , hyväksytään saatu ratkaisu uudeksi ratkaisuksi.
5. Jäähdytä systeemin lämpötilaa  $T$  jäähdytysaikataulun mukaisesti.
6. Keskeytä algoritmin suoritus, mikäli lämpötila  $T < 1$ . Muutoin jatka algoritmin suoritusta kohdasta 3.
- 

Systeemin jäähdytys tapahtuu jäähdytysaikataulun mukaisesti. Lämpötilaa lasketaan algoritmin jokaisella iteraatiolla niin kauan, kunnes lämpötila laskee alle 1 asteen, jolloin algoritmin suoritus keskeytetään. Tässä työssä kehitetyssä algoritmissa käytetään eksponentiaalista jäähdytysaikataulua, jossa lämpötilaa lasketaan toistuvasti vakio kertoimen  $\alpha$  verran ( $0 < \alpha < 1$ ). Jäähdytysaikataulu voidaan esittää seuraavasti:

$$T(t) = T_0 \alpha^t$$

missä  $t$  (aika), kuvaa iteraatioaskeleita. Eksponentiaalinen jäähdytysaikataulu esiteltiin alun perin samaan aikaan simuloitun jäähdytyksen kanssa Kirkpatrickin, Gelattin ja Vecchin työssä [33, s. 675].

### 7.3 Systemin jäähdytysaikataulu

Simuloidussa jäähdytyksessä tapahtuva systeemin jäähdyttäminen voidaan suorittaa käyttäen erilaisia menetelmiä. Työssä esitetyssä ratkaisumenetelmässä käytetään eksponentiaalista jäähdytysaikataulua, jolle annetaan parametreina aloituslämpötila sekä jäähdytystahti.

Sekä eksponentiaalinen, että lineaarinen jäähdytysaikataulu ovat menetelmiä, jotka ovat olleet käytössä simuloitun jäähdytyksen esittelystä saakka. Tämän lisäksi simuloidussa jäähdytyksessä voidaan käyttää muun muassa logaritmista jäähdytysaikataulua [39, s. 8375].

## 8 SUORITUSKYKYMITTAUKSET

Edellisessä pääluvussa esitetty algoritmi SIMANL toteutettiin MATLAB-ohjelmistolla (kts. Liite 3). Sen toimintaa ja suorituskykyä (tulosten hyvyys ja algoritmin nopeus) arvioitiin antamalla ohjelmalle syötteeksi Tóthin, Knuutilan ja Nevalaisen artikkelissa *Reconfiguring flexible machine modules of a PCB assembly* esiteltyjä esimerkkiongelmia 1 – 5 (kts. taulukko 1). Nämä ongelmat simuloivat realistisen kaltaisia erikokoisia optimointiongelmia. Koska piirilevyjen rakenne saattaa vaihdella suuresti, vaihtelevat myös testiongelmien parametrit. Ongelmat on järjestetty vaikeuden mukaan siten, että ongelma 1 on helpoin, kun taas ongelma 5 on luonteeltaan vaativin. [15, sivut 90 - 91] Nyrkkisääntönä on, että MCLB-ongelmat vaikeutuvat sitä mukaa, kun ladottavien komponenttityyppien, ladontamoduulien, ladontapäättyyppien, suutintyyppien ja ladontojen kokonaismäärät kasvavat.

Ongelma	Komponentti-tyypit	Ladontakoneet	Ladontapäättyypit	Suutintyyppit	Ladonnat	Syötinkapasiteetti
1	8	5	2	5	55	10
2	8	4	2	5	74	12
3	10	5	3	8	100	20
4	20	8	4	10	320	30
5	40	8	4	15	525	30

Taulukko 1: Työssä käytetyt testiongelmia 1 - 5 [15, s. 90].

Kaikki suorituskykymittaukset on suoritettu samalla tietokoneella<sup>5</sup>, joten saadut tulokset ovat keskenään vertailukelpoisia. Suorituskykymittaukset tehtiin kullekin ongelmalle toistamalla ratkaisu 20 kertaa ja ottamalla saaduista mittaustuloksista keskiarvo. Suuremmalla otannalla pyritään saamaan yksittäisten algoritmien suorituskyvystä parempi yleiskuva ja parantamaan testauksen tilastollista tarkkuutta. Esimerkiksi simuloitua jäädytystä hyödyntävä algoritmi saattaa tuottaa fitness-arvoltaan reilusti

---

<sup>5</sup> Testijärjestelmän kokoonpano: 64-bittinen Windows 10 Pro -käyttöjärjestelmä (versio 1803), Intel Core i5-6600k suoritin (4 ydintä, 3.5 GHz), 16 GB RAM-muistia. Testauksessa käytetty MATLAB-ohjelmiston versio on 9.3.0.713579 (R2017b).

toisistaan eroavia ratkaisuja peräkkäisillä suorituserroilla, mikäli algoritmin jäähdytysaikataulu on valittu liian nopeaksi.

### **8.1 Alustava testaus parametrien määrittämiseksi**

Simuloidun jäähdytyksen toiminta riippuu suuresti valitusta aloituslämpötilasta sekä jäähdytysaikataulusta. Näiden seikkojen vaikutusta algoritmin toimintaan voidaan testata käytännössä vain testaamalla. Tämän vuoksi työssä tehdyt suorituskykymittaukset voidaan jakaa karkeasti kahteen osuuteen. Ensimmäiseksi, tehtävänä on selvittää, minkälaisella jäähdytysaikataululla ja aloituslämpötilalla algoritmia kannattaa suorittaa. Tämän jälkeen, voidaan testauksen toisessa osuudessa verrata jäähdytysaikataulultaan optimoidun simuloidun jäähdytyksen suorituskykyä muihin ratkaisumenetelmiin.

Tässä työssä alustava testaus tehdään käyttämällä syötteenä esimerkkipongelmaa 4 (kts. kuva 7). Esimerkkipongelma 4 on vaikeudeltaan sen verran haastava, ettei sitä voida esimerkiksi brute force -menetelmällä ratkaista kohtuullisessa ajassa, mutta muiden työssä esiteltyjen menetelmien testaukseen se sopii hyvin. Alustavassa testauksessa tehtävälohkojen (kts. aliluku 6.2) aikakertoimeksi on määritelty  $a = 10$ . Aikakerroin  $a$ :n arvo kuvaa sitä, kuinka suuri painoarvo MCLB-ongelman kohdefunktiossa annetaan ladontapään liikkeille komponenttisyöttimen ja piirilevyn välillä. Kyseinen parametri on ladontakonekohtainen ja lisäksi riippuvainen ladottavana olevan piirilevyn mitoista. Aikakerroin  $a = 10$  kuvaa tilannetta, jossa ladontapään liikkeille annetaan suuri painoarvo. Vastaavasti, jos  $a$ :n arvoksi annettaisiin 0, se kuvaisi tilannetta, jossa kohdefunktiossa jätettäisiin tehtävälohkojen määrä kokonaan huomioimatta. [15, sivut 90 - 91]

Testauksen ensimmäinen osuus tapahtuu suorittamalla simuloitua jäähdytystä hyödyntävä algoritmi useaan kertaan käyttäen erilaisia aloituslämpötiloja ja jäähdytysaikatauluja. Suorituskykyä mitatessa kiinnitetään huomiota saatujen tulosten hyvyyteen (= fitness-arvo) sekä algoritmin suoritusnopeuteen (kts. liite 2). Suorituskykymittausten pääasiallisena tarkoituksena on löytää kokeilemalla sellainen aloituslämpötila ja jäähdytysaikataulu, joilla algoritmi tuottaisi fitness-arvoltaan hyvän tuloksen kohtuullisessa ajassa. Alustavan testauksen tehtävänä on toisin sanoen

optimoida simuloidussa jäähdytyksessä käytettävät asetukset. Aloituslämpötilojen ja jäähdytysaikataulujen valinta tehdään kokeilemalla ja haarukoimalla. Testauksessa saadut tulokset viittaavat siihen, että algoritmi antaa parhaat tulokset silloin, kun aloituslämpötila on erittäin matala (lämpötila  $T \leq 4$ ) ja jäähdytysaikataulu erittäin hidas (vakiokerroin  $\alpha \geq 0.997$ ). Toisaalta, algoritmin suoritusaika kasvaa ja ratkaisun fitness-arvo paranee (pienempi fitness-arvo tarkoittaa parempaa ratkaisua), kun aloituslämpötilaa nostetaan tai jäähdytysaikataulua hidastetaan. Parempia ratkaisuja saadaan siis silloin, kun algoritmin annetaan etsiä ratkaisua pidempään eli se suorittaa useampia iteraatioita ratkaisun löytämiseksi.

Alustavassa testauksessa käytetystä SIMANL-menetelmästä kehitettiin paranneltu versio suorituskkyvertailua varten (aliluvut 8.2 ja 8.3), joten alustavassa testauksessa saatuja tuloksia ei voi suoraan vertailla myöhempänä työssä esiintyviin tuloksiin. Parannellussa SIMANL-menetelmässä palautetaan paras koko tehtävän suorituksen aikana löydetty ratkaisuyksilö, kun taas alkuperäisessä palautettiin se ratkaisuyksilö, johon oli päädytty algoritmin pysähtymishetkellä.

Komponenttityyppien kappalemäärä  $t = 20$   
 Ladontamoduulien kappalemäärä  $m = 8$   
 Ladontapäättyyppien kappalemäärä  $h = 4$   
 Suutintyyppien kappalemäärä  $n = 10$

Komponenttien tyyppikohtaiset ladontamäärät:

$c_1 = 60, c_2 = 45, c_3 = 40, c_4 = 30, c_5 = 30, c_6 = 20, c_7 = 18, c_8 = 15, c_9 = 12, c_{10} = 10, c_{11} = 10, c_{12} = 8, c_{13} = 6, c_{14} = 5,$   
 $c_{15} = 3, c_{16} = 3, c_{17} = 2, c_{18} = 1, c_{19} = 1, c_{20} = 1$

Ladontapää-suutin -yhteensopivuus (HN):

		Suutintyytit									
Ladontapäätyypit	1	1	1	1	1	0	0	0	0	0	0
	0	0	0	0	0	1	1	1	0	0	0
	0	0	0	0	0	0	0	0	1	1	0
	0	0	0	0	0	0	0	0	0	0	1

Komponentti-suutin -yhteensopivuus (CN):

		Suutintyytit									
Komponenttityypit	1	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	0	0
	0	1	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	0	0
	0	0	1	0	0	0	0	0	0	0	0
	0	1	0	0	0	0	0	0	0	0	0
	0	1	0	0	0	0	0	0	0	0	0
	0	0	1	0	0	0	0	0	0	0	0
	0	0	0	1	0	0	0	0	0	0	0
	0	0	0	0	1	0	0	0	0	0	0
	0	0	0	0	0	1	0	0	0	0	0
	0	0	0	0	0	0	1	0	0	0	0
	0	0	0	0	0	0	0	1	0	0	0
	0	0	0	0	0	0	0	0	1	0	0
	0	0	0	0	0	0	0	0	0	1	0
	0	0	0	0	0	0	0	0	0	0	1
	0	0	0	0	0	0	0	0	0	0	1
	0	0	0	0	0	0	0	0	0	0	1
	0	0	0	0	0	0	0	0	0	0	1

Komponenttien leveydet:

$w_1 = 1, w_2 = 1, w_3 = 2, w_4 = 1, w_5 = 2, w_6 = 3, w_7 = 3, w_8 = 2, w_9 = 2, w_{10} = 4, w_{11} = 4, w_{12} = 5, w_{13} = 5, w_{14} = 4, w_{15} = 5,$   
 $w_{16} = 6, w_{17} = 7, w_{18} = 7, w_{19} = 8, w_{20} = 8$

Ladontapäättyyppien suutinkapasiteetit:

$ht_1 = 8, ht_2 = 4, ht_3 = 2, ht_4 = 1$

Moduulikohtaiset komponenttien käsittelyajat:

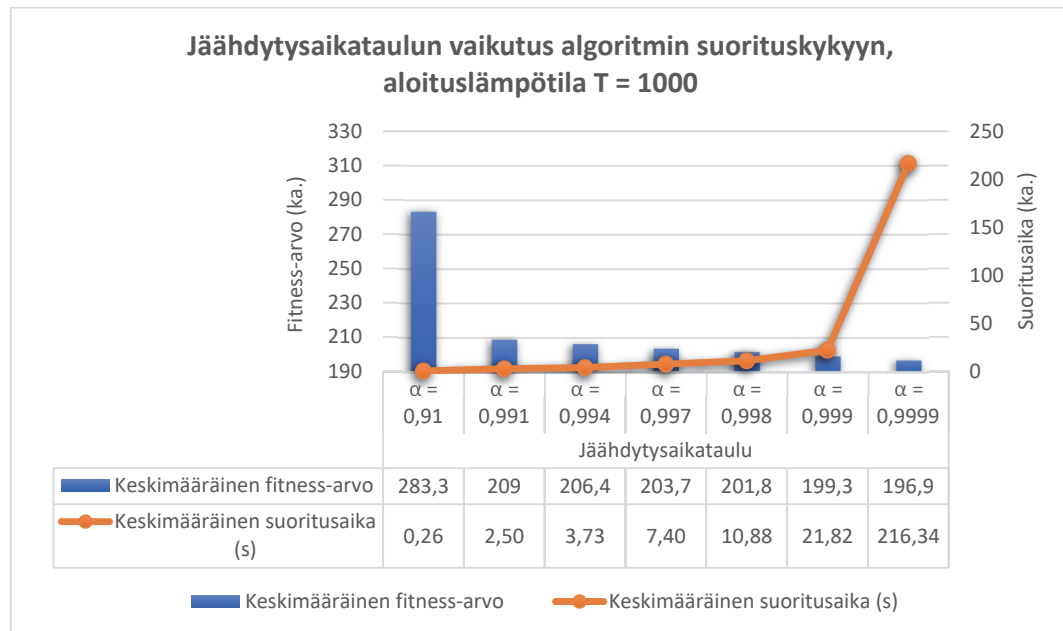
Ladonta-aika (th):  $th_1 = 1, th_2 = 1, th_3 = 1, th_4 = 1, th_5 = 1, th_6 = 1, th_7 = 1, th_8 = 1$

Noukinta-aika (tp):  $tp_1 = 1, tp_2 = 1, tp_3 = 1, tp_4 = 1, tp_5 = 1, tp_6 = 1, tp_7 = 1, tp_8 = 1$

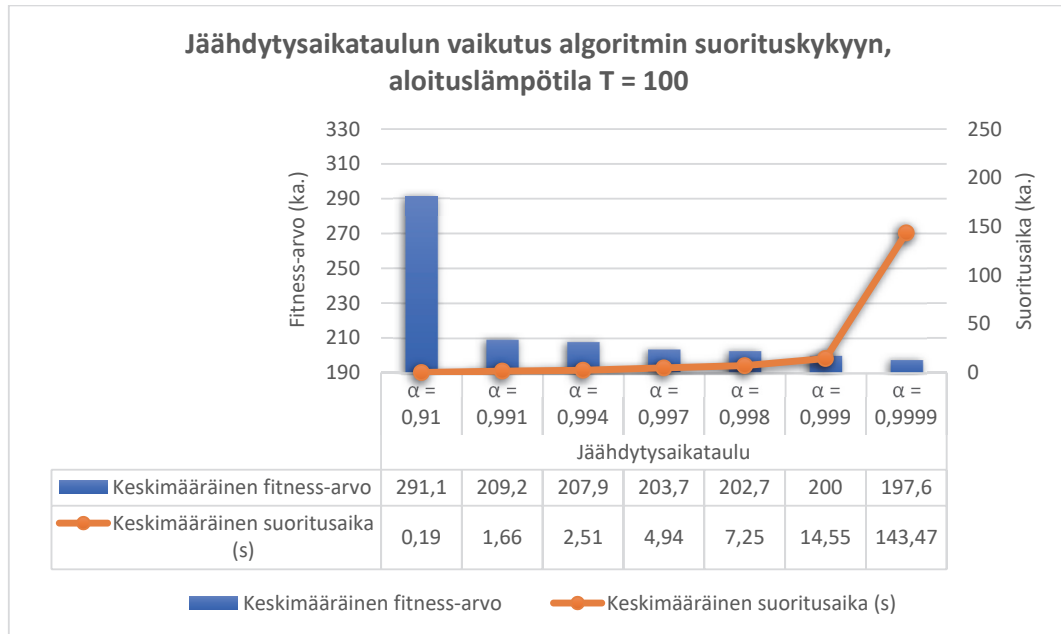
Syötinkapasiteetti (fc) = 30

Kuva 7: Testiongelma 4 [15].

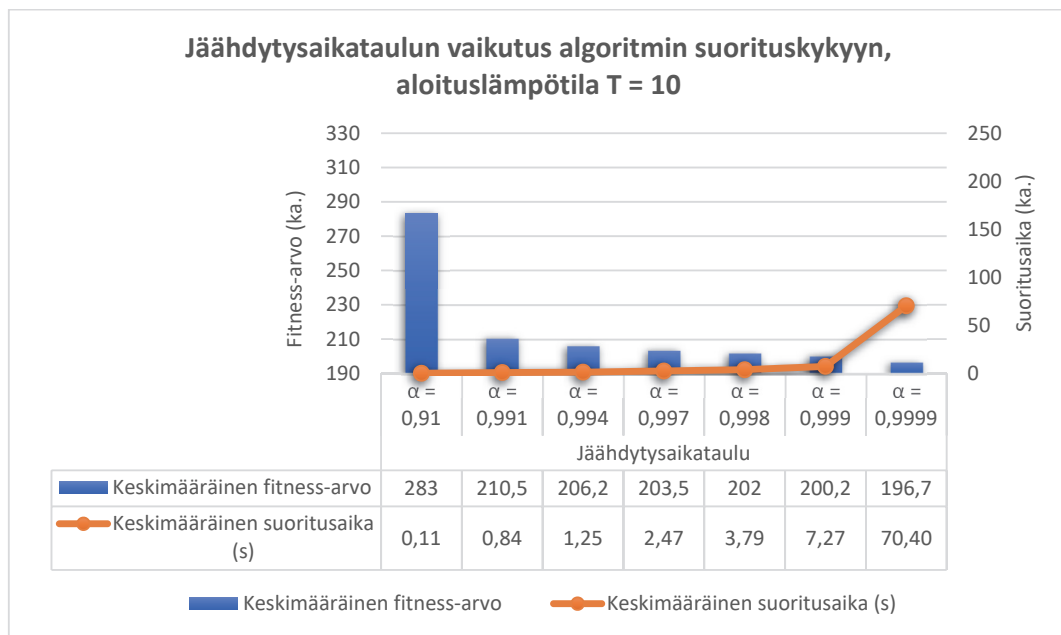
Testauksen perusteella huomataan, että jäähdystisaikataulun hidastaminen on aloituslämpötilan nostamista tehokkaampi tapa parantaa saatujen ratkaisujen hyvyyttä. Käyttämällä pienempää aloituslämpötilaa ja hitaampaa jäähdystisaikataulua, verrattuna korkeampaan aloituslämpötilaan ja nopeampaan jäähdystisaikatauluun saatiin testauksessa lähes poikkeuksetta parempia ratkaisuja samassa suoritusajassa. Kuvissa 8 - 13 havainnollistetaan jäähdystisaikataulun vaikutusta algoritmin suorituskykyyn erilaisilla aloituslämpötiloilla. Systemin liian nopea jäähdystys saa algoritmin tuottamaan heikkoja tuloksia, kun taas erityisen hidas jäähdystys hidastaa ratkaisun löytymistä. Hidas jäähdystys tuottaa kuitenkin fitness-arvoltaan parhaat ratkaisut. Kuvaajista nähdään selkeästi myös, kuinka korkeampi aloituslämpötila käytännössä vain hidastaa algoritmin suoritusta, ilman että algoritmi tuottaisi parempia tuloksia, kun jäähdystisaikataulu on valittu oikein.



Kuva 8: Jäähdystisaikataulun vaikutus simuloitua jäähdystystä hyödyntävän algoritmin suorituskykyyn, kun aloituslämpötilaksi on valittu T = 1000. Syötteenä on käytetty testiongelmaa 4 (320 ladontaa) (kts. kuva 7). Pysty akselin siniset palkit kuvaavat ratkaisujen keskimääräisiä fitness-arvoja ja oranssi kuvaaja vastaavasti keskimääräisiä suoritusajoja, kun jäähdystisaikataulua (vaaka-akseli) hidastetaan.

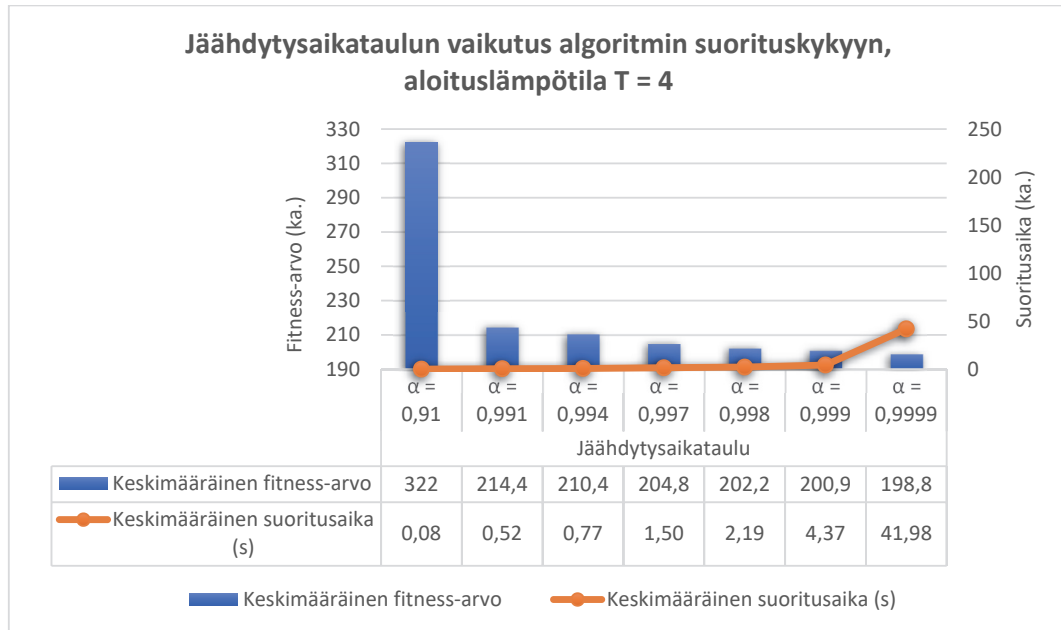


Kuva 9: Jäähdytysaikataulun vaikutus simuloitua jäähdytystä hyödyntävän algoritmin suorituskykyyn, kun aloituslämpötilaksi on valittu T = 100. Syötteenä on käytetty testiongelmaa 4 (320 ladontaa) (kts. kuva 7). Pysty akselin siniset palkit kuvaavat ratkaisujen keskimääräisiä fitness-arvoja ja oranssi kuvaaja vastaavasti keskimääräisiä suoritusajoja, kun jäähdytysaikataulua (vaaka-akseli) hidastetaan.

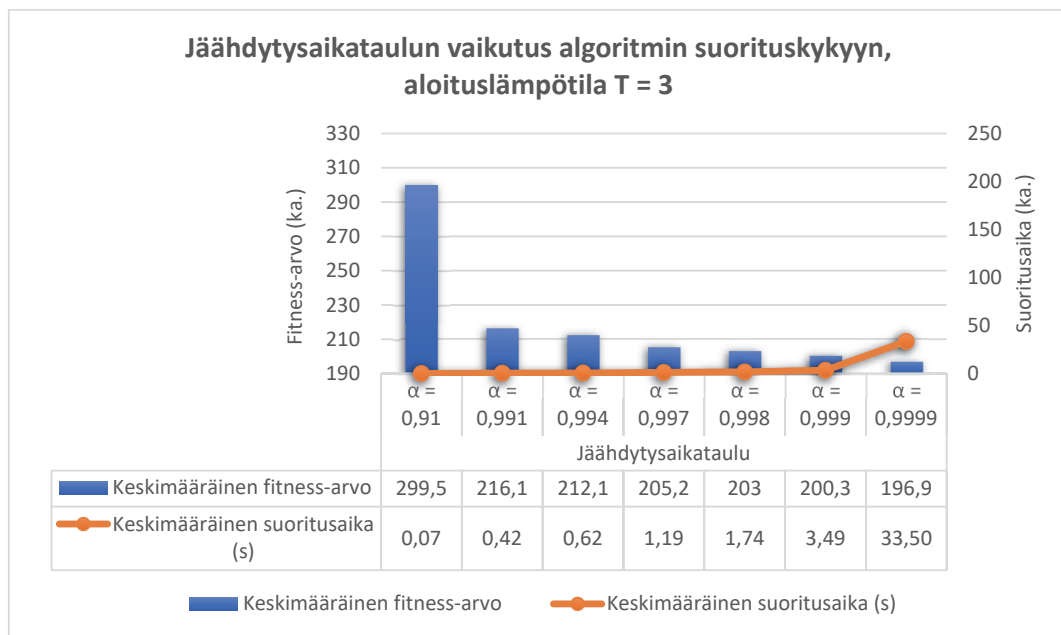


Kuva 10: Jäähdytysaikataulun vaikutus simuloitua jäähdytystä hyödyntävän algoritmin suorituskykyyn, kun aloituslämpötilaksi on valittu T = 10. Syötteenä on käytetty testiongelmaa 4 (320 ladontaa) (kts. kuva 7). Pysty akselin siniset palkit kuvaavat ratkaisujen keskimääräisiä fitness-arvoja ja oranssi kuvaaja vastaavasti keskimääräisiä suoritusajoja, kun jäähdytysaikataulua (vaaka-akseli) hidastetaan.

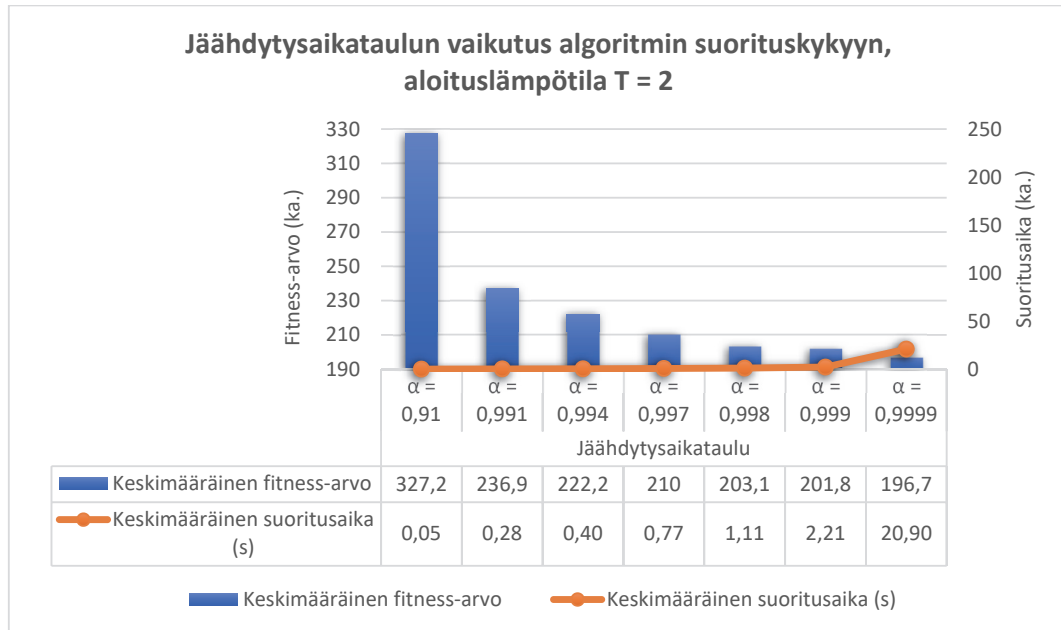




Kuva 11: Jäähdytysaikataulun vaikutus simuloitua jäähdytystä hyödyntävän algoritmin suorituskykyyn, kun aloituslämpötilaksi on valittu T = 4. Syötteenä on käytetty testiongelmaa 4 (320 ladontaa) (kts. kuva 7). Pysty akselin siniset palkit kuvaavat ratkaisujen keskimääräisiä fitness-arvoja ja oranssi kuvaaja vastaavasti keskimääräisiä suoritus aikoja, kun jäähdytysaikataulua (vaaka-akseli) hidastetaan.



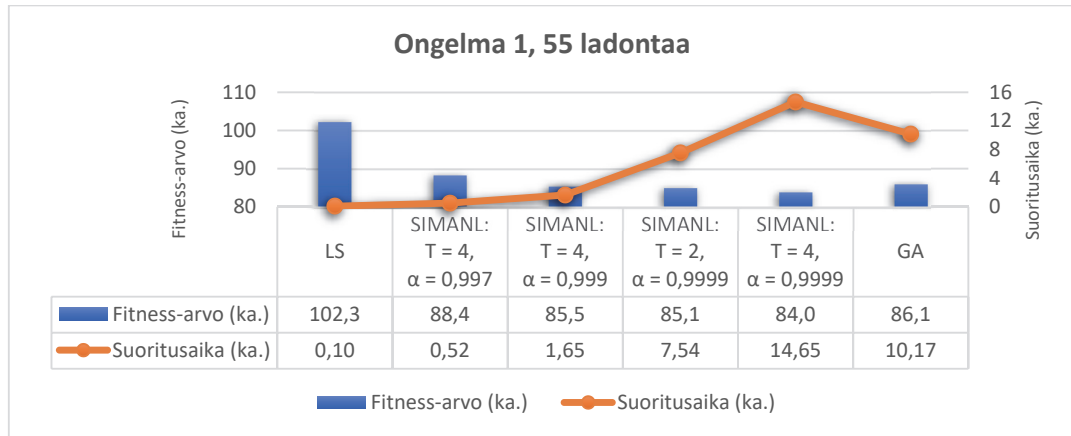
Kuva 12: Jäähdytysaikataulun vaikutus simuloitua jäähdytystä hyödyntävän algoritmin suorituskykyyn, kun aloituslämpötilaksi on valittu T = 3. Syötteenä on käytetty testiongelmaa 4 (320 ladontaa) (kts. kuva 7). Pysty akselin siniset palkit kuvaavat ratkaisujen keskimääräisiä fitness-arvoja ja oranssi kuvaaja vastaavasti keskimääräisiä suoritus aikoja, kun jäähdytysaikataulua (vaaka-akseli) hidastetaan.



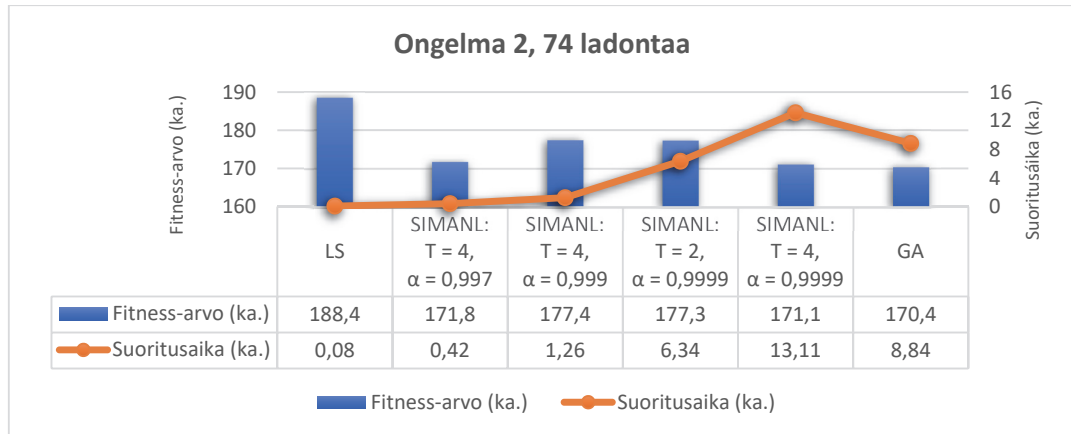
Kuva 13: Jäähdytysaikataulun vaikutus simuloitua jäähdytystä hyödyntävän algoritmin suorituskykyyn, kun aloituslämpötilaksi on valittu T = 2. Syötteenä on käytetty testiongelmaa 4 (320 ladontaa) (kts. kuva 7). Pystyakselin siniset palkit kuvaavat ratkaisujen keskimääräisiä fitness-arvoja ja oranssi kuvaaja vastaavasti keskimääräisiä suoritusajoja, kun jäähdytysaikataulua (vaaka-akseli) hidastetaan.

## 8.2 Simuloidun jäähdytyksen suorituskyky verrattuna muihin menetelmiin

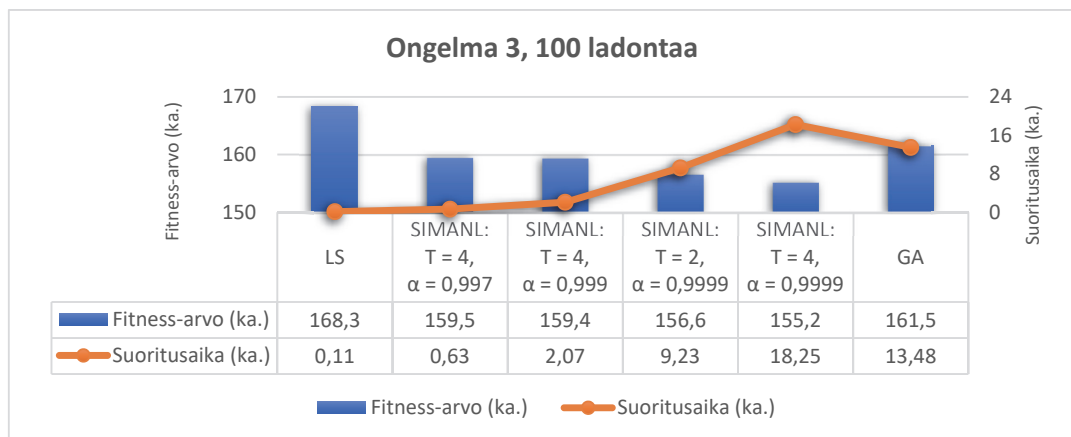
Suorituskykymittausten toisessa osiossa simuloitua jäähdytystä verrataan muihin MCLB-ongelman ratkaiseviin menetelmiin. Simuloidun jäähdytyksen aloituslämpötilaksi ja jäähdytysaikatauluksi valittiin testauksen ensimmäisen vaiheen perusteella neljä eri asetusta, jotka on ilmoitettu kuvissa 14 - 18. Esimerkkiongelmina käytettiin kaikkia viittä ongelmaa, joiden perustiedot on esitelty aiemmin kuvassa 6. Tehtävälölköjen aikakerroin pysyy samana kuin alustavassa testauksessa, eli  $a = 10$ . Vertailukohdaksi simuloitua jäähdytystä käyttävälle menetelmälle valittiin Tóthin, Knuutilan ja Nevalaisen artikkelissa esitellyistä menetelmistä paikallista hakua (kuvissa käytetty lyhennettä LS) ja geneettistä algoritmia (kuvissa käytetty lyhennettä GA) käyttävät menetelmät. Vertaaminen normaaliin paikallista hakua käyttävään algoritmiin on mielenkiintoista siksi, että simuloitua jäähdytystä käyttävä menetelmä poikkeaa käytännön toteutukseltaan paikallisesta hausta varsin vähän, mutta sillä pitäisi teoriassa saada aikaan fitness-arvoltaan huomattavasti parempia ratkaisuja. Geneettisen algoritmin valinta toiseksi vertailukohdaksi oli itsestään selvää, sillä Tóthin, Knuutilan ja Nevalaisen artikkelissa geneettinen algoritmi suoriutui testeistä muita menetelmiä paremmin [15, s. 90].



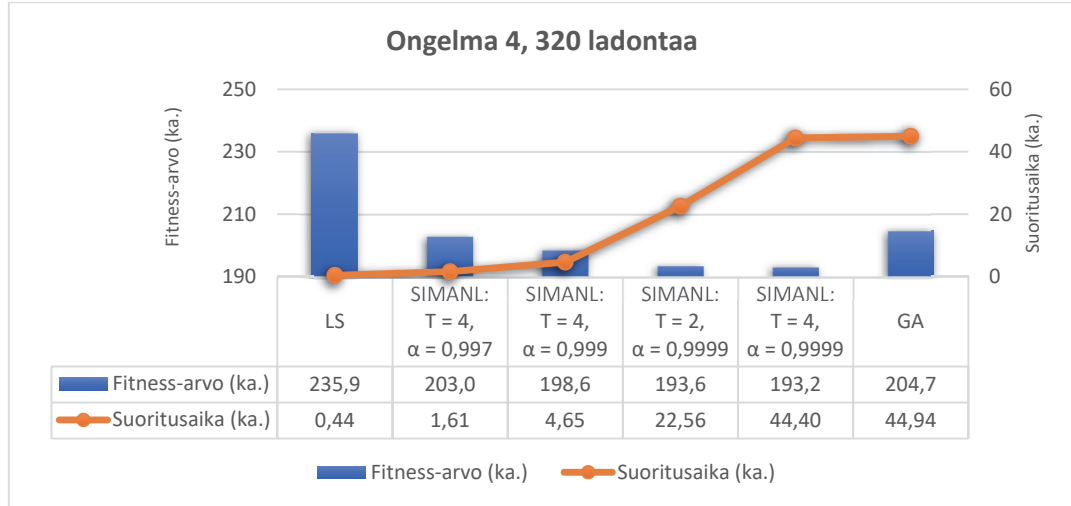
Kuva 14: Ongelman 1 ratkaiseminen eri menetelmillä. Sininen palkki kuvaa kunkin ratkaisun fitness-arvojen keskiarvoa. Oranssi kuvaaja kertoo vastaavasti, kuinka kauan kunkin menetelmän kestää keskimäärin löytää ratkaisu.



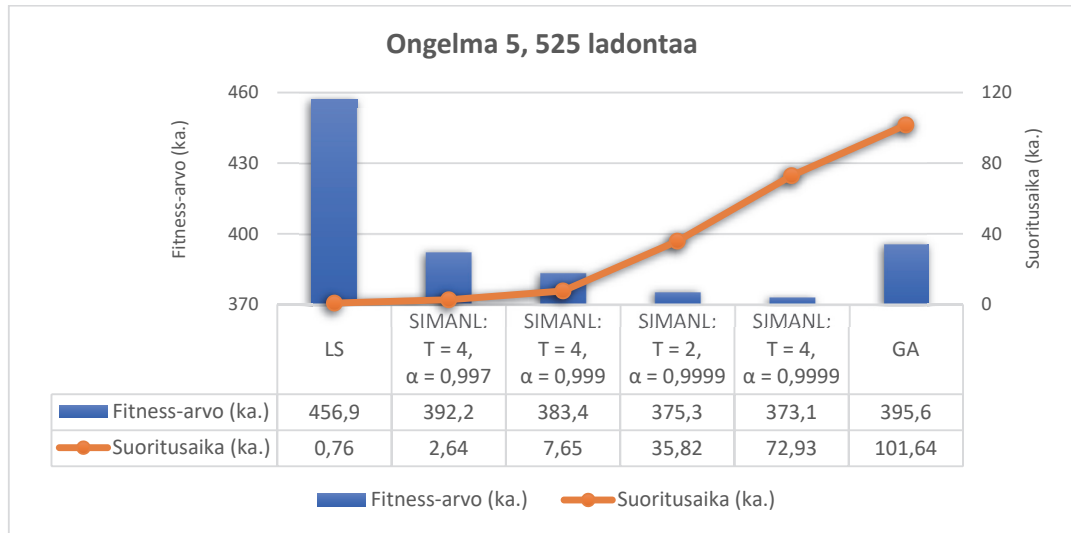
Kuva 15: Ongelman 2 suorituskyky erilaisilla menetelmillä. Sininen palkki kuvaa kunkin ratkaisun fitness-arvojen keskiarvoa. Oranssi kuvaaja kertoo vastaavasti, kuinka kauan kunkin menetelmän kestää keskimäärin löytää ratkaisu.



Kuva 16: Ongelman 3 suorituskyky eri menetelmillä. Sininen palkki kuvaa kunkin ratkaisumenetelmän fitness-arvojen keskiarvoa. Oranssi kuvaaja kertoo vastaavasti, kuinka kauan kunkin menetelmän kestää keskimäärin löytää ratkaisu.



Kuva 17: Ongelman 4 ratkaiseminen erilaisilla menetelmillä. Sininen palkki kuvaa kunkin ratkaisun fitness-arvojen keskiarvoa. Oranssi kuvaaja kertoo vastaavasti, kuinka kauan kunkin menetelmän kestää keskimäärin löytää ratkaisu.



Kuva 18: Ongelman 5 ratkaiseminen erilaisilla menetelmillä. Sininen palkki kuvaa kunkin ratkaisun fitness-arvojen keskiarvoa. Oranssi kuvaaja kertoo vastaavasti, kuinka kauan kunkin menetelmän kestää keskimäärin löytää ratkaisu.

Vertailutestien perusteella paikalliseen hakuun perustuva algoritmi on menetelmistä nopein kaikkien ongelmatapauksien ratkaisussa. Toisaalta paikallisen haun löytämät ratkaisut ovat myös fitness-arvoiltaan kaikkein heikompia. Fitness-arvoltaan parhaaksi menetelmäksi osoittautuu simuloitu jäädytys kaikissa ongelmatapauksissa lukuun ottamatta ongelmaa 2.

Ongelmaan 1, jossa on 55 ladontaa, saadaan parhaan fitness-arvon omaava ratkaisu käyttäen simuloitua jäähdystä ( $T = 4$ ,  $\alpha = 0,9999$ ). Paikalliseen hakuun verrattuna fitness-arvo on kyseisessä ongelmassa simuloitulla jäähdystyksellä n. 22 % parempi. Geneettiseen algoritmiin verrattuna saavutetaan vastaavasti noin 2,4 % parempi fitness-arvo. Geneettinen algoritmi on simuloitua jäähdystä n. 31 % nopeampi.

Ongelma 2 (74 ladontaa) saadaan ratkaistua parhaiten geneettisellä algoritmilla, tosin ratkaisujen keskimääräinen fitness-arvo on vain 0,4 % parempi kuin simuloitua jäähdystä ( $T = 4$ ,  $\alpha = 0,9999$ ) hyödyntävällä algoritmilla. Geneettinen algoritmi toimii myös n. 33 % nopeammin kuin simuloitu jäähdystys.

Ongelmaan 3 (100 ladontaa) saadaan fitness-arvoltaan paras ratkaisu käyttäen simuloitua jäähdystä ( $T = 4$ ,  $\alpha = 0,9999$ ). Saatu ratkaisu on 3,9 % parempi ja se löydetään n. 26 % hitaammin, kuin ratkaisemalla sama ongelma käyttäen geneettistä algoritmia.

Ongelma 4, jossa on 320 ladontaa, on jo selkeästi monimutkaisempi kuin edellä mainitut ongelmat. Parhaan ratkaisun ongelmaan löytää simuloitu jäähdystys ( $T = 4$ ,  $\alpha = 0,9999$ ), jonka ratkaisu on fitness-arvoltaan keskimäärin 5,6 % parempi kuin geneettisen algoritmin löytämä ratkaisu. Simuloitu jäähdystys löytää ratkaisun keskimäärin n. 1,2 % nopeammin kuin geneettinen algoritmi.

Ongelmaan 5 (525 ladontaa), paras ratkaisu löydetään simuloitulla jäähdystyksellä ( $T = 4$ ,  $\alpha = 0,9999$ ). Tällä menetelmällä saadun ratkaisun fitness-arvo on keskimäärin 5,7 % parempi kuin geneettiseen algoritmiin perustuvalla ratkaisumenetelmällä. Simuloitu jäähdystys löytää ratkaisun keskimäärin n. 28 % nopeammin kuin geneettinen algoritmi.

Saatujen tulosten perusteella simuloitulla jäähdystyksellä on mahdollista päästä geneettistä algoritmia selvästi parempiin tuloksiin erityisesti tehtävien vaikeutuessa (ongelmat 4 ja 5). Simuloitu jäähdystys on myös selkeästi nopeampi ongelman 5 ratkaisemisessa. Se kykenee tuottamaan geneettistä algoritmia parempia ratkaisuja myös nopeammin toimivilla jäähdystysaika-alueilla ( $T = 2$ ,  $\alpha = 0,9999$  ja  $T = 4$ ,  $\alpha = 0,999$ ) kaikilla esimerkkiongelmilla lukuun ottamatta esimerkkiongelmaa 2.

### 8.3 Tehtävälohkojen aikakertoimen vaikutus ratkaisumenetelmien suorituskykyyn

Kaikissa edellä tehdyissä testeissä on MCLB-ongelman kohdefunktiossa tehtävälohkojen aikakertoimeksi määritelty  $a = 10$ . Tóthin, Knuutilan ja Nevalaisen artikkelissa [15] tehdyissä ratkaisumenetelmiä vertailevissa suorituskykymittauksissa on tarkasteltu myös, miten tehtävälohkojen aikakertoimen muutokset vaikuttavat ratkaisumenetelmien suorituskykyyn. Tóthin, Knuutilan ja Nevalaisen mukaan MCLB-ongelman kohdefunktiossa olevan aikakertoimen  $a$  arvoa ei ole syytä lukita mihinkään tiettyyn arvoon, sillä se määräytyy sekä käytössä olevan ladontakoneen, että ladottavana olevan piirilevyn mittojen mukaan. Artikkelissa tehty testaaminen suoritettiin  $a$ :n arvoilla 0, 1, 2, 5 sekä 10. Kun  $a = 0$ , kuvaa se tapausta, jossa ladottavana olevan piirilevyn koko on suuri verrattuna piirilevyn ja syöttölaitteen väliseen etäisyyteen toisistaan. Kun  $a$ :n arvoa kasvatetaan kahteen tai viiteen, asetetaan MCLB-ongelman kohdefunktiossa suurempi painoarvo tehtävälohkojen lukumäärälle. Kun  $a = 10$ , tarkoittaa se sitä, että ladontapään liikkeet piirilevyn ja syöttimen alueella ovat minimaalisia. [15, sivut 90 - 91]

Tóthin, Knuutilan ja Nevalaisen artikkelissa tehdyssä testauksessa saatujen tuloksien perusteella geneettinen algoritmi päihitti muut artikkelissa vertailtavana olleet ratkaisumenetelmät kaikilla  $a$ :n arvoilla. Sen tuottamissa ratkaisuisa parhaan ja keskimääräisen ratkaisun fitness-arvot säilyvät lähellä toisiaan. Kasvatettaessa  $a$ :n arvoa, artikkelissa testattujen ratkaisumenetelmien tuottamat ratkaisut muuttuivat hajanaisemmiksi. Kun  $a$ :n arvoksi määriteltiin 10, olivat myös geneettisen algoritmin osalta saatujen ratkaisujen fitness-arvot hajanaisia. [15, sivut 90 - 91]

Simuloidun jäähtymisen suorituskykyä verrataan geneettisen algoritmin suorituskykyyn tehtävälohkojen erilaisilla aikakertoimilla, kuten Tóthin, Knuutilan ja Nevalaisen artikkelissa on verrattu erilaisia MCLB-ongelman ratkaisumenetelmiä. Yhteenveto saaduista tuloksista on esitetty taulukossa 2. Testauksessa etsitään ratkaisut kaikille edellä mainitulle esimerkkiongelmille (esimerkkiongelmät 1 – 5). Simuloidun jäähtymisen parametreiksi valittiin kaikille ongelmille aloituslämpötilaksi  $T = 4$  ja vakiokertoimeksi  $\alpha = 0,9999$ . Parametreja ei voida valita siten, että ne olisivat optimaalisia kaikille mahdollisille ongelmatapauksille, mutta edellisessä aliluvussa tehdyn testauksen perusteella simuloitu jäähtymis toimii edellä mainituilla parametreilla riittävän hyvin

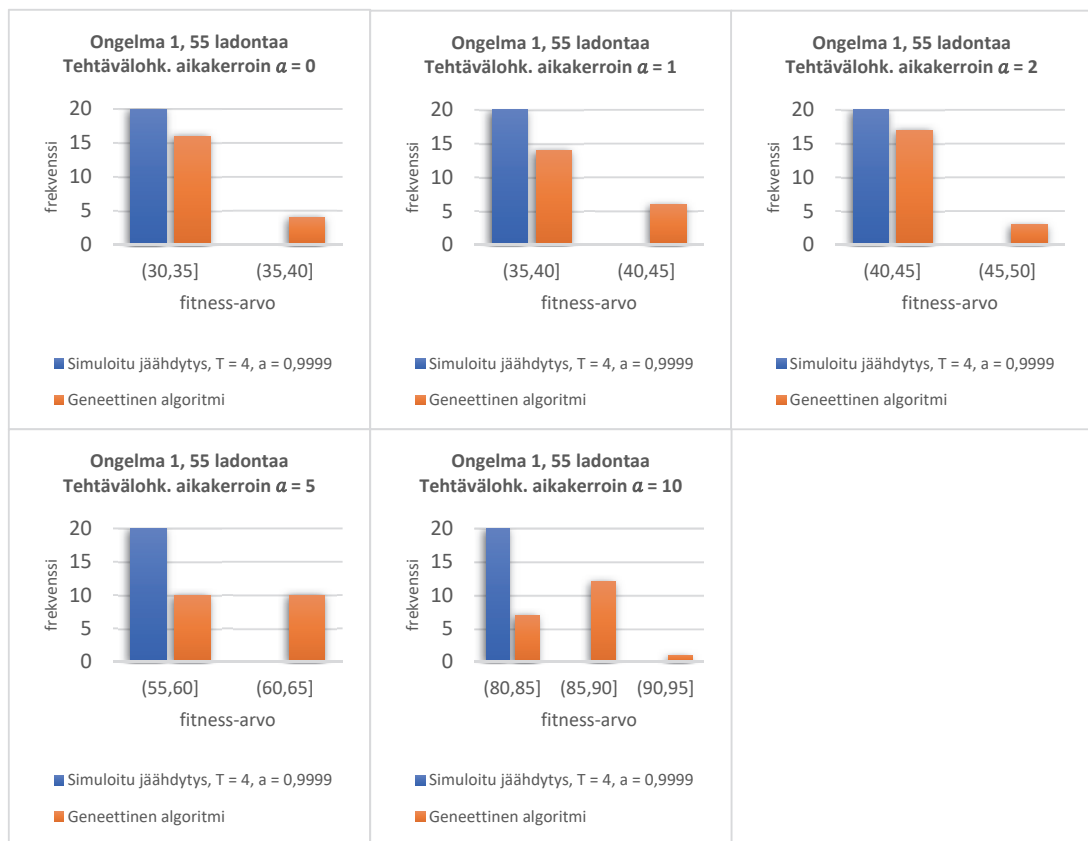
riippumatta ongelmasta. Simuloidun jäähtymisen ja geneettisen algoritmin testaaminen tapahtuu tehtävälölköjen aikakertoimen  $a$  arvoilla 0, 1, 2, 5 sekä 10. Jokainen esimerkkiongelma ratkaistiin 20 kertaa.

		Simuloitu jäähtymys		Geneettinen algoritmi	
Ongelma		Fitness-arvojen keskiarvo	Paras fitness-arvo	Fitness-arvojen keskiarvo	Paras fitness-arvo
Aikakerroin $a = 0$	1	<u>34,00</u>	34,00	34,50	34,00
	2	68,00	68,00	68,00	68,00
	3	90,00	90,00	90,00	90,00
	4	<u>112,20</u>	112,00	113,40	112,00
	5	<u>250,00</u>	250,00	250,10	250,00
Aikakerroin $a = 1$	1	<u>39,00</u>	39,00	39,95	39,00
	2	<u>78,00</u>	78,00	78,90	78,00
	3	<u>97,00</u>	97,00	97,85	96,00
	4	<u>121,50</u>	121,00	122,75	122,00
	5	<u>263,40</u>	263,00	265,50	265,00
Aikakerroin $a = 2$	1	<u>44,00</u>	44,00	44,50	44,00
	2	<u>88,00</u>	88,00	89,30	88,00
	3	<u>104,00</u>	104,00	105,30	104,00
	4	<u>130,10</u>	130,00	132,40	130,00
	5	<u>276,60</u>	276,00	280,60	278,00
Aikakerroin $a = 5$	1	<u>59,00</u>	59,00	60,30	59,00
	2	<u>118,00</u>	118,00	119,60	118,00
	3	<u>123,55</u>	122,00	126,85	123,00
	4	<u>153,70</u>	152,00	158,75	155,00
	5	<u>314,55</u>	312,00	322,45	319,00
Aikakerroin $a = 10$	1	<u>84,00</u>	84,00	86,10	84,00
	2	171,10	168,00	<u>170,40</u>	168,00
	3	<u>155,20</u>	152,00	161,50	158,00
	4	<u>193,20</u>	192,00	204,70	198,00
	5	<u>373,10</u>	368,00	395,60	388,00

Taulukko 2: Tehtävälölköjen aikakertoimen vaikutus ratkaisumenetelmien suorituskykyyn.

Testien perusteella tehtävälölköjen aikakertoimen muutoksilla on selvä vaikutus saatujen ratkaisujen fitness-arvoihin. Simuloidun jäähtymisen ja geneettisen algoritmin löytämät ratkaisut pienemmillä  $a$ :n arvoilla (0, 1 ja 2) eivät eroa toisistaan merkittävästi. Simuloitu jäähtymys löytää kuitenkin suurimmassa osassa ongelmia keskimäärin hieman paremman

ratkaisun kuin geneettinen algoritmi. Eri ratkaisumenetelmillä saadut parhaan fitness-arvon omaavat ratkaisut eivät pienillä  $a$ :n arvoilla myöskään eroa toisistaan suuresti. Suurimmat erot kahden menetelmän välillä syntyvät  $a$ :n arvolla 5 (esimerkkiongelmalla 5) ja  $a$ :n arvolla 10 (esimerkkiongelmilla 4 ja 5), jolloin simuloitua jäähdystä hyödyntävä ratkaisumenetelmä löytää selkeästi parempia ratkaisuja. Keskimääräisten fitness-arvojen eroavaisuudet eri menetelmillä selittyvät ainakin osittain löydettyjen ratkaisujen hajautumisella. Kuvissa 19 - 23 esitetään ongelmien ratkaisujen fitness-arvojen jakautumista  $a$ :n eri arvoilla.

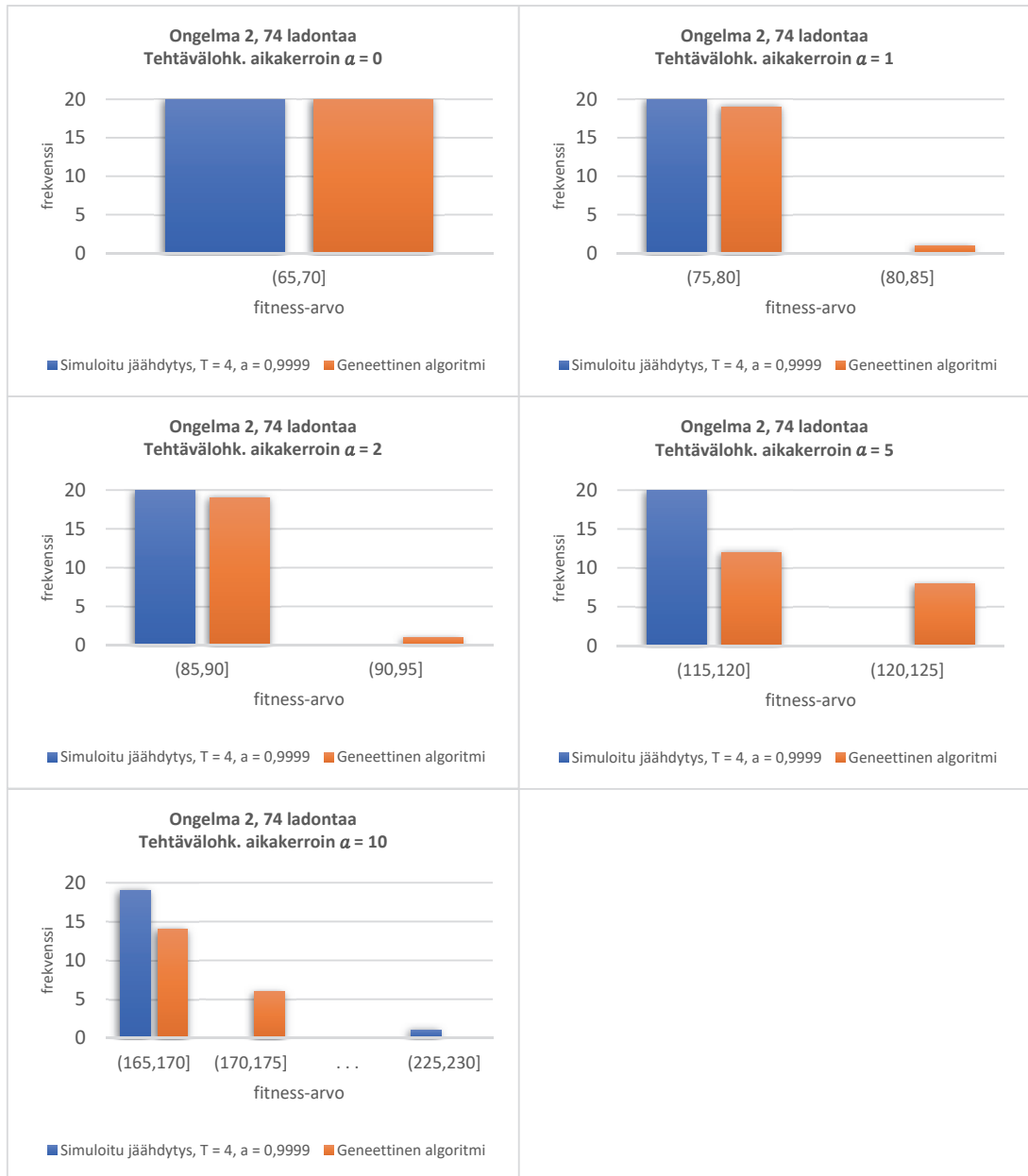


Kuva 19: Tehtävälötköjen aikakertoimen vaikutus ratkaisumenetelmien suorituskykyyn, esimerkkiongelmalla 1. Palkit kuvaavat millä frekvenssillä ratkaisujen fitness-arvot kuuluvat tiettyyn arvoalueeseen.

Kuvassa 19 nähdään tehtävälötköjen aikakertoimen vaikutus esimerkkiongelman 1 ratkaisujen fitness-arvoihin. Fitness-arvojen hajautuminen on voimakkaampaa geneettisellä algoritmilla erityisesti silloin, kun  $a$ :n arvoa kasvatetaan. Simuloitu jäähdys toimii erittäin vakaasti esimerkkiongelman 1 ratkaisussa riippumatta  $a$ :n arvosta.

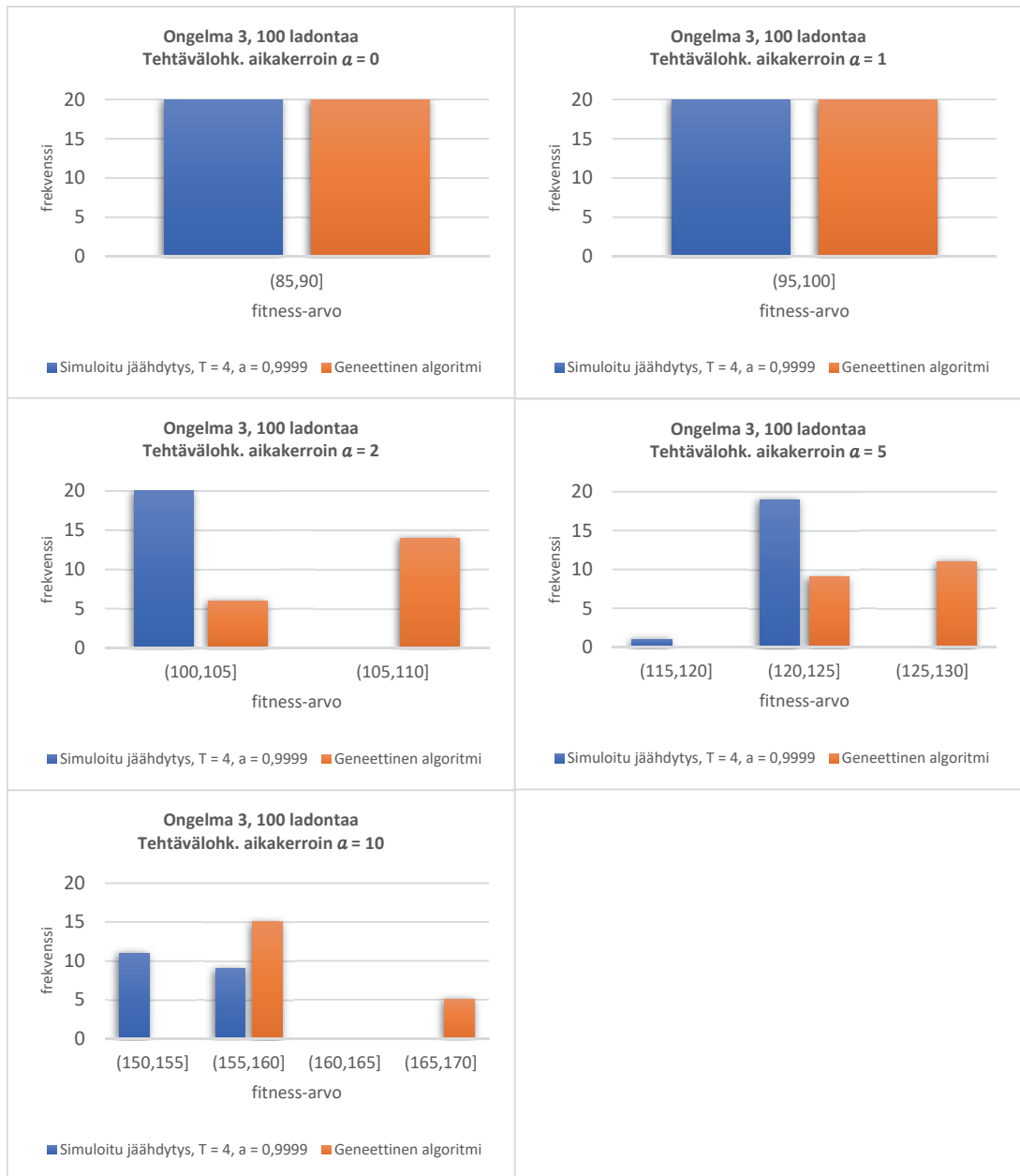


Esimerkkiongelman 2 (kts. kuva 20) ratkaisujen fitness-arvojen hajautuminen kasvaa vähän aikakertoimen  $a$  kasvaessa. Ratkaisumenetelmien keskinäinen suorituskyky pienillä  $a$ :n arvoilla on lähes identtistä. Simuloitu jäähdytys toimii keskimäärin heikommin  $a$ :n arvolla 10 johtuen yksittäisestä fitness-arvoltaan selvästi heikommasta ratkaisusta.



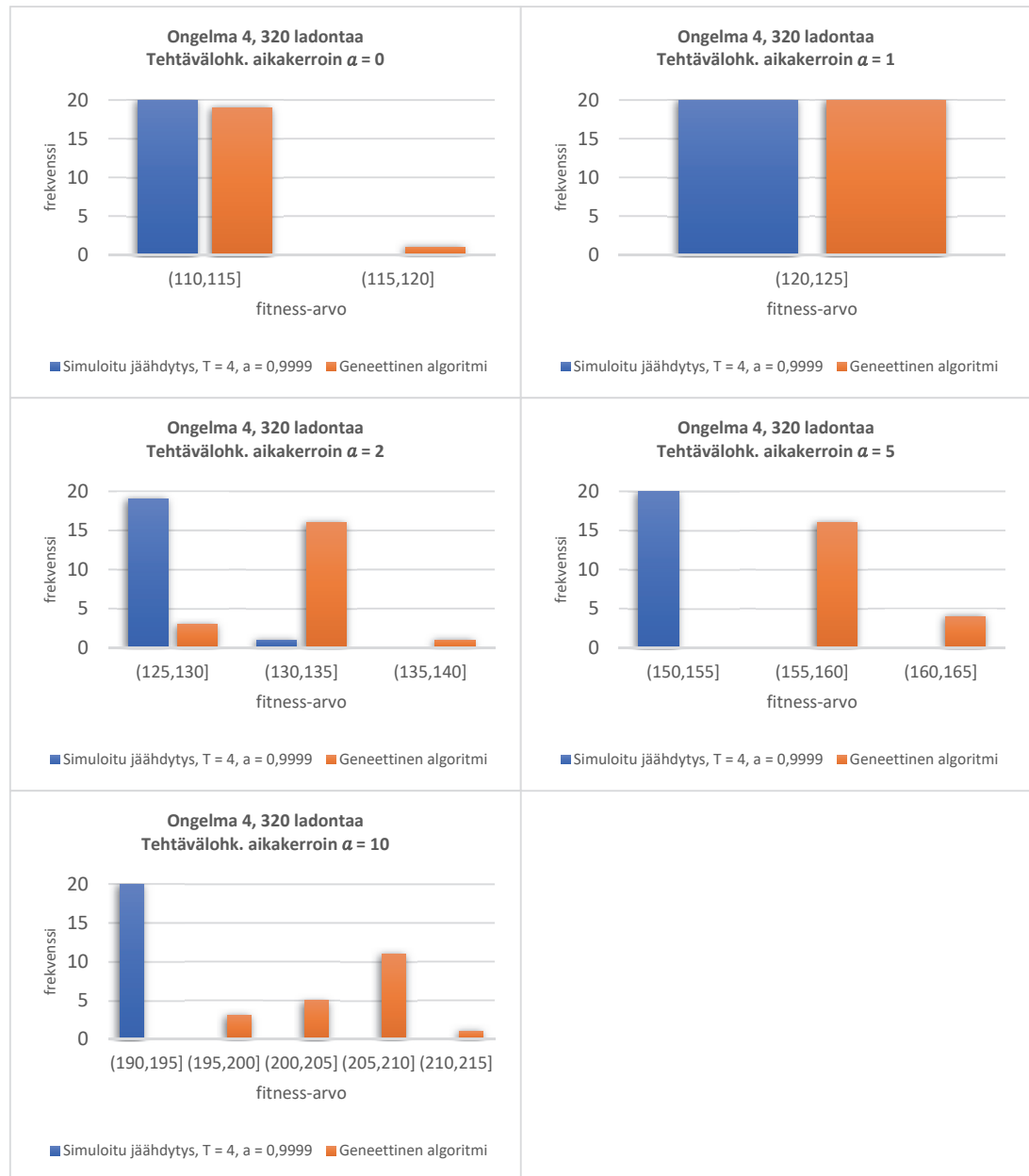
Kuva 20: Tehtävälötkojen aikakerroimen vaikutus ratkaisumenetelmien suorituskykyyn, esimerkkiongelman 2.

Kuvassa 21 nähdään aikakerroin  $a$ :n muutosten vaikutus ratkaistaessa esimerkki-ongelmaa 3. Kun  $a = 0$  tai  $a = 1$ , molemmat menetelmät toimivat erittäin stabiilisti. Saatujen ratkaisujen fitness-arvot hajautuvat geneettisellä algoritmilla, kun  $a$ :n arvoa kasvatetaan. Erityisesti silloin, kun  $a = 10$ , geneettisen algoritmin löytämien ratkaisujen fitness-arvot hajautuvat voimakkaasti.

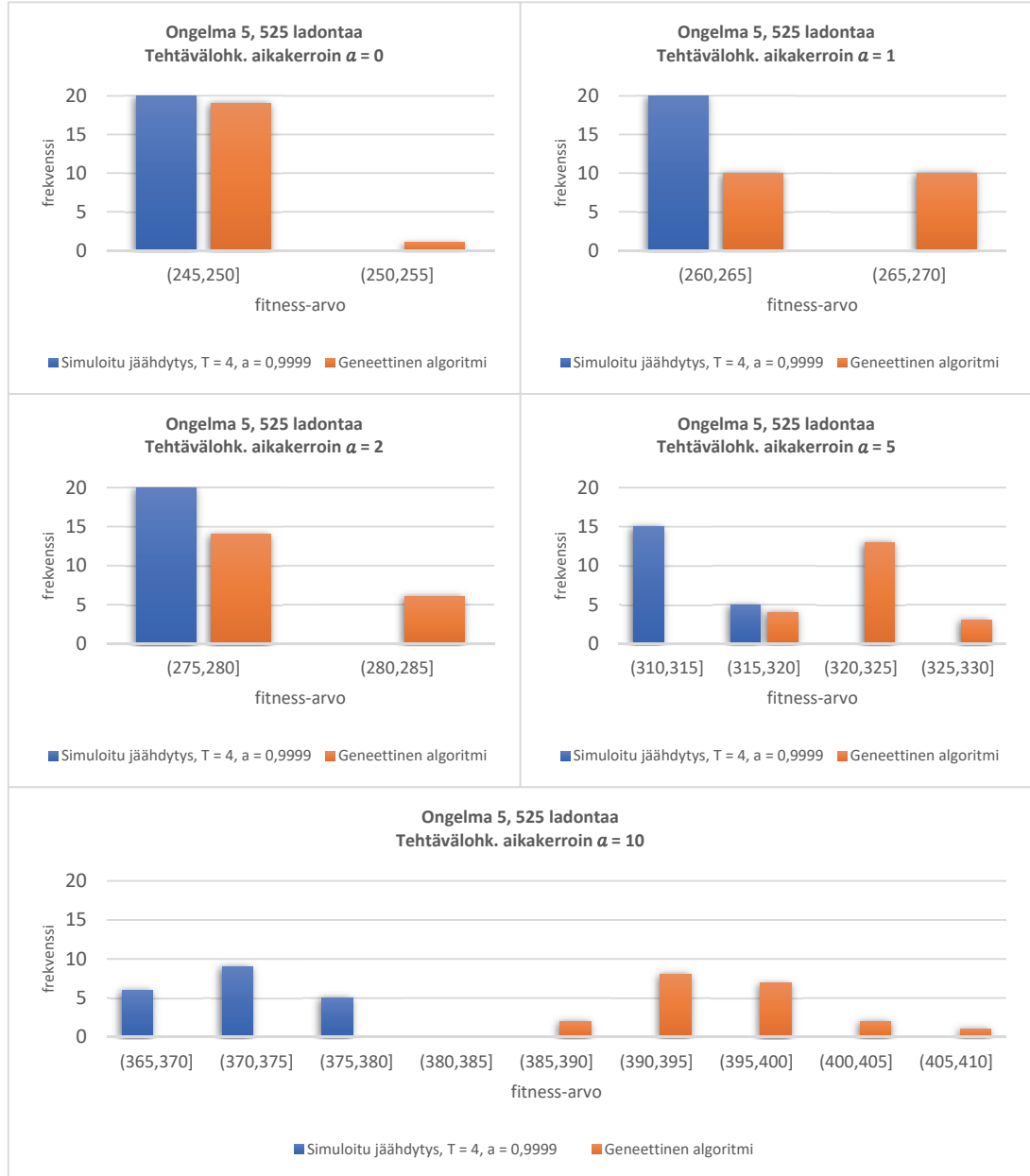


Kuva 21: Tehtävälököjen aikakerroimen vaikutus ratkaisumenetelmien suorituskykyyn, esimerkki-ongelma 3.

Simuloitu jäähditys toimii erittäin vakaasti esimerkkiongelman 4 ratkaisemisessa riippumatta tehtävälohkojen aikakertoimesta (kts. kuva 22). Geneettinen algoritmi sen sijaan toimii lähes yhtä hyvin  $\alpha$ :n arvoilla 0 ja 1, mutta suuremmilla aikakertoimen arvoilla sen löytämät ratkaisut alkavat hajaantua voimakkaasti. Kun  $\alpha = 10$ , toimii geneettinen algoritmi jo selkeästi heikommin kuin simuloitu jäähditys.



Kuva 22: Tehtävälohkojen aikakertoimen vaikutus ratkaisumenetelmien suorituskykyyn, esimerkkiongelma 4.



Kuva 23: Tehtävälohkojen aikakertoimen vaikutus ratkaisumenetelmien suorituskykyyn, esimerkkiongelma 5.

Esimerkkiongelman 5 ratkaisujen fitness-arvojen jakautumien eri  $\alpha$ : n arvoilla on esitetty kuvassa 23. Kun aikakertoimeksi valitaan  $\alpha = 0$ , ei vertailtavana olevien menetelmien välillä ole suurta eroa. Aikakertoimen  $\alpha$  arvoilla 1 ja 2 simuloitu jäädytys toimii edelleen erittäin vakaasti, sen sijaan geneettisen algoritmin löytämät ratkaisut hajaantuvat jonkin verran. Kun  $\alpha = 5$  tai  $\alpha = 10$ , syntyvät suurimmat erot eri ratkaisumenetelmien välillä.

Tällöin molemmilla menetelmillä saatujen ratkaisujen fitness-arvot hajautuvat, mutta simuloitu jäähditys saavuttaa selkeästi parempia ratkaisuja. Ratkaisujen hajautumisen perusteella tehtävälököjen aikakertoimen arvolla  $a = 10$  ongelman 5 voidaan katsoa olevan haastava molemmille ratkaisumenetelmille.

Testauksessa simuloitun jäähdityksen ja geneettisen algoritmin suorituskykyä vertailtiin sekä erikokoisilla ongelmatehtävillä, että eri kokoisilla tehtävälököjen aikakertoimen arvoilla. Erot ratkaisumenetelmien välillä ovat hyvin pieniä, kun syötteenä annettut tehtävät ovat helppoja (pienet ladontamäärät ja pienet tehtävälököjen kertoimen  $a$  arvot). Testien perusteella simuloitu jäähditys on kuitenkin parempi menetelmä lähes kaikissa testitapauksissa. Erityisesti vaikeissa tehtävissä, joissa ladontamäärät ovat suuria ja tehtävälököjen aikakerroin on suuri, simuloitu jäähditys toimii geneettistä algoritmia selvästi paremmin. Verrattuna Tóthin, Knuutilan ja Nevalaisen artikkelissa [15] esiteltyihin ratkaisumenetelmiin (satunnainen, ahnas ja paikallinen haku), tuottavat sekä simuloitu jäähditys että geneettinen algoritmi selkeästi parempia ja fitness-arvoltaan vähemmän hajautuvia ratkaisuja [15, s. 93]

## 9 LOPUKSI

Tutkielmassa käsiteltiin piirilevyjen koneellista ladontaa käyttäen modulaarisia ladontakoneita. Kyseinen ladontakonetyyppi palvelee erityisen hyvin modernin elektroniikkateollisuuden tarpeita, jossa valmistettavana olevat tuotteet vaihtuvat suhteellisen tiheästi.

Työssä esitettiin simuloitua jäähdystä hyödyntävä menetelmä MCLB-ongelman ratkaisemiseksi. Työssä käsiteltyä MCLB-ongelmaa oli monilta osin yksinkertaistettu. Esimerkiksi tietyntyyppisiä komponentteja pystytään tyypillisesti käsittelemään erilaisilla suuttimilla, mutta tässä työssä oli tehty yksinkertaistus, jossa jokaiselle komponenttityypille oli olemassa vain yksi yhteensopiva suutintyyppi. Lisäksi MCLB-ongelmassa käsitelty ladontakone oli konfigurointiominaisuuksiltaan varsin rajoitettu verrattuna niihin mahdollisuuksiin, joita esitettiin työn neljännessä pääluvussa.

Työssä esitelty ratkaisumenetelmä osoittautui hyväksi menetelmäksi ratkaista MCLB-ongelma. Verrattuna geneettiseen algoritmiin, sen suorituskyky oli monilla ongelmasyötteillä käytännöllisesti katsoen samaa tasoa. Vaikeampien ongelmien ratkaisussa simuloitua jäähdystä hyödyntävä menetelmä sen sijaan onnistui löytämään parempia ratkaisuja. Geneettiseen algoritmiin verrattuna, simuloitua jäähdystä voidaan pitää huomattavasti helpompana toteuttaa.

Simuloidun jäähdytyksen toiminta riippuu suuresti sille annetuista parametreista. Käyttäjän on osattava valita sopiva jäähdystysaikataulu, aloituslämpötila sekä systeemin jäähtymistähti. Koska kyseisten parametrien valinnalla on vaikutusta sekä ratkaisujen laatuun, että algoritmin nopeuteen, käytettiin työssä melko suuri aika edellä mainittujen parametrien testaamiseen erilaisilla ongelmasyötteillä. Tästä työvaiheesta voidaan käyttää myös termiä 'opettaminen'. Tähän seikkaan liittyen joihinkin työssä saatuihin tuloksiin on suhtauduttava varauksella. Simuloidun jäähdytyksen tarvitsemat optimaaliset parametrit valittiin testauksen perusteella, jossa käytettiin ainakin osittain samaa syötemateriaalia kuin lopullisessa suorituskykyarvioinnissa. Optimaalisessa tapauksessa opetuksessa käytetyn datan pitäisi koostua eri materiaalista, kuin lopullisessa

testauksessa käytetty data. Muutoin saattaa käydä niin, että opetusmateriaalilla hyvin toimineen algoritmin suorituskyky heikkenee oleellisesti, kun se saa syötteekseen ennalta tuntematonta syötemateriaalia. Kyseisestä virheestä käytetään termiä ylisovittaminen. Suorituskykymittauksia käsittelevässä luvussa simuloitun jäähtymisen suorituskyvyn oleellista heikkenemistä ei kuitenkaan havaittu, kun testimateriaalina käytettiin opetusdatasta poikkeavaa syötemateriaalia. Suorituskykytesteissä esiintyneen geneettisen algoritmin kehitysvaiheessa käytetystä opetusdatasta ei ole saatu tietoa.

Simuloitua jäähtymistä hyödyntävän ratkaisumenetelmän jäähtymisestä vastaa työssä eksponentiaalinen jäähtytysaikataulu. Eriolaisten jäähtytysaikataulujen vaikutus ratkaisumenetelmän tuottamiin ratkaisuihin olisi mielekäs jatkotutkimuksen kohde. Lisäksi työssä esitettyä MCLB-ongelman mallia voitaisiin kehittää esimerkiksi vähentämällä siinä esiintyviä yksinkertaistuksia, jolloin se vastaisi paremmin käytännön optimointitehtäviä.

## LÄHDELUETTELO

- [1] Statista: ”*Electronics | Statista*”, [WWW-sivu]. Saatavilla osoitteesta: <https://www.statista.com/markets/407/topic/929/electronics/>. [Viitattu 25. elokuuta 2018].
- [2] Fortune: ”*Fortune Global 500 List 2018: See Who Made It*”, [WWW-sivu]. Saatavilla osoitteesta: <http://fortune.com/global500/list/>. [Viitattu 25. elokuuta 2018].
- [3] Zentralverband Elektrotechnik- und Elektronikindustrie e.V.: ”*The Global Electrical & Electronic Industry – Facts & Figures*”, [PDF]. Saatavilla osoitteesta: [https://www.zvei.org/fileadmin/user\\_upload/Presse\\_und\\_Medien/Publikationen/2018/Juli/Die\\_globale\\_Elektroindustrie\\_Daten\\_Zahlen\\_Fakten/Fact-Sheet-International-2018.pdf](https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/2018/Juli/Die_globale_Elektroindustrie_Daten_Zahlen_Fakten/Fact-Sheet-International-2018.pdf). [Viitattu 29. elokuuta 2018].
- [4] M. Grunow, H. O. Günther ja M. Schleusener: ”*Component allocation for printed circuit board assembly using modular placement machines*”, julkaisussa *International Journal of Production Research* **41**(6), sivut 1311 - 1331, 2003.
- [5] M. Ayob ja G. Kendall: ”*The optimisation of the single surface mount device placement machine in printed circuit board assembly: a survey*”, julkaisussa *International Journal of Systems Science*, **40**(6), sivut 553 - 569, 2009.
- [6] C. Raduly-Baka, T. Knuutila, M. Johnsson ja O. S. Nevalainen: ”*Construction of component tapes for radial placement machines*”, julkaisussa *Computers & Operations Research*, **37**(8), sivut 1488 - 1499, 2010.
- [7] Intel: ”*Intel's Packaging Databook Chapter 7: Leaded Surface Mount Technology (SMT)*”, Intel, 2000, [PDF]. Saatavilla osoitteesta: <https://www.intel.sg/content/dam/www/public/us/en/documents/packaging-databooks/packaging-chapter-07-databook.pdf> [Viitattu 13. syyskuuta 2017].



- [8] J. Ahonen, T. Haiko ja K. Salonen: *Elektroniikka*, Sanoma Pro Oy, 2016.
- [9] M. Ayob ja G. Kendall: ”*A survey of surface mount device placement machine optimisation: Machine classification*”, julkaisussa *European Journal of Operational Research*, **186**(3), sivut 893 - 914, 2008.
- [10] A. Rong, A. Tóth, O. S. Nevalainen, T. Knuutila ja R. Lahdelma: ”*Modeling the machine configuration and line-balancing problem of a PCB assembly line with modular placement machines*”, julkaisussa *The International Journal of Advanced Manufacturing Technology*, **54**(1), sivut 349 - 360, 2011.
- [11] T. Knuutila, T. Suomi, S. Emet, M. Johnsson ja O. S. Nevalainen: ”*Organizing the nozzle magazine of a gantry-type PCB assembly machine*”, julkaisussa *The International Journal of Advanced Manufacturing Technology*, **68**(5 - 8), sivut 1189 - 1202, 2013.
- [12] Panasonic Factory Solutions Company of America: ”*NPM SERIES - Any-mix manufacturing solution for evolving assembly needs*”, 2015. [PDF].  
Saataavilla osoitteesta:  
[https://www.panasonicfa.com/sites/default/files/pdfs/Panasonic\\_NPM\\_Series\\_2015.pdf](https://www.panasonicfa.com/sites/default/files/pdfs/Panasonic_NPM_Series_2015.pdf). [Viitattu 25. syyskuuta 2017].
- [13] ASM Assembly Systems GmbH & Co: ”*SIPLACE SX-Series: Capacity-on-Demand – for maximum manufacturing flexibility*”, 2017. [PDF].  
Saataavilla osoitteesta:  
[http://www.asm-smt.com/addmindms/download.aspx?domid=10&d\\_id=b7564845-06bf-4922-a5be-d31f4153176b&fdl=0](http://www.asm-smt.com/addmindms/download.aspx?domid=10&d_id=b7564845-06bf-4922-a5be-d31f4153176b&fdl=0). [Viitattu 26. lokakuuta 2017].
- [14] Fuji Machine MFG. CO., LTD.: ”*Fuji NXT III - Fuji Scalable Placement Platform*”, 2015. [PDF]. Saataavilla osoitteesta:  
[www.amtech.cz/pdf/NXT\\_III.pdf](http://www.amtech.cz/pdf/NXT_III.pdf). [Viitattu 31. lokakuuta 2017].
- [15] A. Tóth, T. Knuutila ja O. S. Nevalainen: ”*Reconfiguring flexible machine modules of a PCB assembly line*”, julkaisussa *Production Engineering*, **4**(1), sivut 85-94, 2010.
- [16] P. Csaszar, P. C. Nelson, R. R. Rajbhandari ja T. M. Tirpak: ”*Optimization of automated high-speed modular placement machines using knowledge-based*

- systems*”, julkaisussa *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, **30(4)**, sivut 408 - 417, 2000.
- [17] K. Kallio, M. Johnsson ja O. S. Nevalainen: ”*Estimating the operation time of flexible surface mount placement machines*”, julkaisussa *Production Engineering*, **6(3)**, sivut 319 - 328, 2012.
- [18] Sulaon Oy: ”*Ohjeita*”, [WWW-sivu]. Saatavilla osoitteesta: <http://pintaliitos.fi/ohjeita/index.html>. [Viitattu 9. tammikuuta 2018].
- [19] S. van Gastel: ”*The Road Ahead: Single or Dual Lane?*”, julkaisussa *SMT Magazine*, **27(5)**, sivut 90 - 93, 2012.
- [20] FlexLink: ”*GENIUS PCB handling system*”, 2017. [PDF]. Saatavilla osoitteesta: [http://www.flexlink.com/zcms/zpublish/45/uploads/45/tech\\_lib/1448543020307727371\\_5868EN-Genius-Flexlink.pdf](http://www.flexlink.com/zcms/zpublish/45/uploads/45/tech_lib/1448543020307727371_5868EN-Genius-Flexlink.pdf). [Viitattu 6. kesäkuuta 2018].
- [21] Sulaon Oy: ”*SMT-Prosessi*”, [WWW-sivu]. Saatavilla osoitteesta: <http://pintaliitos.fi/smt-prosessi/index.html>. [Viitattu 11. lokakuuta 2017].
- [22] C.-H. Wu, D.-Z. Wang, A. Ip, D.-W. Wang, C.-Y. Chan ja H.-F. Wang: ”*A particle swarm optimization approach for components placement inspection on printed circuit boards*”, julkaisussa *Journal of Intelligent Manufacturing*, **20(5)**, sivut 535 - 549, 2009.
- [23] Saki Corporation: ”*SAKI Corporation*”, 2017. [WWW-sivu]. Saatavilla osoitteesta: <http://www.sakiglobal.com/solder-paste-inspection-spi.html>. [Viitattu 27. helmikuuta 2018].
- [24] Saki Corporation: ”*SAKI Corporation*”, 2017. [WWW-sivu]. Saatavilla osoitteesta: <http://www.sakiglobal.com/technology.html>. [Viitattu 23. helmikuuta 2018].
- [25] Saki Corporation: ”*2D and 3D automated optical inspection (AOI) systems - Saki Global*”, 2017. [WWW-sivu]. Saatavilla osoitteesta: <http://www.sakiglobal.com/automated-optical-inspection-aoi.html>. [Viitattu 28. helmikuuta 2018].

- [26] Saki Corporation: ”*True 3D X-ray Vision Inspection Systems - Saki America*”, 2017. [WWW-sivu]. Saatavilla osoitteesta:  
<http://www.sakiglobal.com/automated-x-ray-inspection-axi.html>.  
 [Viitattu 28. helmikuuta 2018].
- [27] Y. Crama, J. van de Klundert ja F. C. Spijksma: ”*Production planning problems in printed circuit board assembly*”, julkaisussa *Discrete Applied Mathematics*, **123**(1 - 3), sivut 339 - 361, 2002.
- [28] S. Guo, K. Takahashi, K. Morikawa ja Z. Jin: ”*An integrated allocation method for the PCB assembly line balancing problem with nozzle changes*”, julkaisussa *The International Journal of Advanced Manufacturing Technology*, **62**(1 - 4), sivut 351 - 369, 2012.
- [29] A. Tóth, T. Knuutila ja O. S. Nevalainen: ”*Machine configuration and workload balancing of modular placement machines in multi-product PCB assembly*”, julkaisussa *International Journal of Computer Integrated Manufacturing*, **31**(9), sivut 815 - 830, 2018.
- [30] F. Vainio, T. Pahikkala, M. Johnsson, O. S. Nevalainen ja T. Knuutila: ”*Estimating the production time of a PCB assembly job without solving the optimised machine control*”, julkaisussa *International Journal of Computer Integrated Manufacturing*, **28**(8), sivut 823 - 835, 2015.
- [31] D. Li ja S. W. Yoon: ”*PCB assembly optimization in a single gantry high-speed rotary-head collect-and-place machine*”, julkaisussa *The International Journal of Advanced Manufacturing Technology*, **88**(9 - 12), sivut 2819 - 2834, 2017.
- [32] Y. Crama, O. E. Flippo, J. Van De Klundert ja F. C. Spijksma, ”*The component retrieval problem in printed circuit board assembly*”, julkaisussa *International Journal of Flexible Manufacturing Systems*, **8**(4), sivut 287 - 312, 1996.
- [33] W. Ho ja P. Ji: ”*PCB assembly line assignment: a genetic algorithm approach*”, julkaisussa *Journal of Manufacturing Technology Management*, **16**(6), sivut 682 - 692, 2005.
- [34] S. Kirkpatrick, C. D. Gelatt ja M. P. Vecchi: ”*Optimization by simulated annealing*”, julkaisussa *Science*, **220**(4598), sivut 671 - 680, 1983.

- [35] V. Černý: ”*Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm*”, julkaisussa *Journal of optimization theory and applications*, **45**(1), sivut 41 - 51, 1985.
- [36] A. Luoto: *Simuloidun jäädytyksen suppenemislause*, Jyväskylän yliopisto, 2013. [Pro gradu –tutkielma].
- [37] J. Haataja, J. Heikonen, Y. Leino, J. Rahola, J. Ruokolainen ja V. Savolainen: *Numeeriset menetelmät käytännössä*, CSC-Tieteellinen laskenta, 2002.
- [38] W. Ben-Ameur: ”*Computing the initial temperature of simulated annealing*”, julkaisussa *Computational Optimization and Applications*, **29**(3), sivut 369 - 385, 2004.
- [39] E. Mäkinen ja T. Poranen: *Algoritmit*, 2. korjattu painos, Tampereen yliopisto, Tietojenkäsittelytieteiden laitos, 2009. [Verkojulkaisu]
- [40] Y. Nourani ja B. Andersen: ”*A comparison of simulated annealing cooling strategies*”, julkaisussa *Journal of Physics A: Mathematical and General*, **31**(41), sivut 8373 - 8385, 1998.
- [41] M. M. Mäkelä: *Matemaattinen optimointi II*, Turun yliopisto, Matematiikan ja tilastotieteen laitos, 2017. [Luentomoniste].

## Liite 1: Englanninkieliset termit ja niistä käytetyt suomennokset

Tässä tutkielmassa pyritään käyttämään suomenkielisiä termejä aina, kun se on järkevää. Suomenkielessä ei ole kaikille englanninkielisille tutkielmassa esitellyille termeille välttämättä vakiintuneita suomenkielisiä ilmaisuja. Tällöin lukija, jolle englanninkielinen termi voi olla sinänsä täysin tuttu, ei välttämättä ymmärrä tekstiä, jossa käytetään suomenkielistä termiä. Tässä lukijan avuksi tehdyssä listassa on lueteltu tutkielmassa käytettyjä termejä ja niiden alkuperäisiä, englanninkielisiä vastineita.

annostelupää	dispensing head
askelpalkkityyppinen liukuhihna	walking-beam conveyor
juotospasta	solder paste
kaksoissyöttöladontakone	dual-delivery placement machine
kohdistusmerkki	fiducial mark
kääntömoduuli	inverter
ladontakonekonfiguraation ja linjaston tasapainotuksen ongelma	machine configuration and line balancing problem, MCLB
ladontamoduuli	placement module
ladontapää	placement head
liikkuva ristikkopalkki	gantry
läpiladontatekniikka	through-hole technology
modulaarinen piirilevyldontakone	modular placement machine
moniasemaladontakone	multi-station placement machine
monipääladontakone	multi-head placement machine
peräkkäistoiminen nouki-ja-asetatyyppinen ladontakone	sequential pick-and-place machine
piirilevyjen kuljetusalusta	pallet
pintaliitostekniikka	surface-mount technology
revolveriladontakone	turret-type placement machine
risteysmoduuli	turn unit
simuloitu jäähdytys	simulated annealing
sivuttaissiirtomodulaari	side shuttle

sulatusuuni	reflow oven
suutin	nozzle
suutinpankki	nozzle bank, nozzle station
syöttölaite, syötinlaite	feeder
tarkastuspää	inspection head
uudelleenkonfigurointi lennossa	on-the-fly changeover

## Liite 2: Simuloidun jäädytyksen suorituskykymittaukset erilaisilla aloituslämpötiloilla ja jäädytysaikatauluilla

Alustuslämpötila	Jäädytysaikataulu $\alpha = 0,91$		Jäädytysaikataulu $\alpha = 0,991$		Jäädytysaikataulu $\alpha = 0,994$		Jäädytysaikataulu $\alpha = 0,997$		Jäädytysaikataulu $\alpha = 0,998$		Jäädytysaikataulu $\alpha = 0,999$		Jäädytysaikataulu $\alpha = 0,9999$	
	Fitness-arvo (ka.)	Suoritusajat (ka.)	Fitness-arvo (ka.)	Suoritusajat (ka.)	Fitness-arvo (ka.)	Suoritusajat (ka.)	Fitness-arvo (ka.)	Suoritusajat (ka.)	Fitness-arvo (ka.)	Suoritusajat (ka.)	Fitness-arvo (ka.)	Suoritusajat (ka.)	Fitness-arvo (ka.)	Suoritusajat (ka.)
T = 1000	283,3	0,26	209,0	2,50	206,4	3,73	203,7	7,40	201,8	10,88	199,3	21,82	196,9	216,34
T = 500	287,0	0,24	208,9	2,29	208,1	3,35	203,8	6,67	203,0	9,81	201,0	19,74	196,4	194,11
T = 250	267,9	0,22	208,5	2,01	206,4	2,97	203,0	5,92	201,1	8,74	199,8	17,57	197,4	177,30
T = 125	275,0	0,19	207,6	1,74	205,4	2,63	204,3	5,18	201,8	7,63	199,2	15,31	197,1	150,52
T = 100	291,1	0,19	209,2	1,66	207,9	2,51	203,7	4,94	202,7	7,25	200,0	14,55	197,6	143,47
T = 90	261,4	0,18	210,2	1,62	207,2	2,44	203,4	4,82	201,5	7,10	200,5	14,17	196,8	140,30
T = 80	284,5	0,18	212,1	1,59	206,2	2,37	204,4	4,69	201,1	6,98	199,1	13,89	197,1	136,63
T = 70	272,4	0,18	207,6	1,53	207,8	2,30	204,1	4,57	200,6	6,70	199,8	13,42	197,2	133,45
T = 60	290,7	0,17	208,8	1,47	207,8	2,23	204,2	4,37	202,4	6,46	200,9	12,89	197,2	127,97
T = 50	277,7	0,16	207,5	1,42	208,3	2,11	202,3	4,20	201,8	6,18	199,3	12,32	197,2	121,33
T = 40	277,8	0,16	209,0	1,34	205,1	2,01	203,7	3,96	201,8	5,81	199,0	11,64	197,6	114,17
T = 30	271,4	0,15	208,7	1,24	205,3	1,84	204,3	3,66	203,5	5,38	200,4	10,70	196,5	104,98
T = 20	286,5	0,13	208,8	1,09	205,2	1,63	206,5	3,22	203,1	5,01	200,4	9,47	198,0	92,27
T = 10	283,0	0,11	210,5	0,84	206,2	1,25	203,5	2,47	202,0	3,79	200,2	7,27	196,7	70,40
T = 9	277,1	0,10	210,6	0,81	207,9	1,20	203,1	2,38	201,4	3,46	200,4	6,95	196,7	67,12
T = 8	283,6	0,10	209,8	0,77	208,1	1,14	204,6	2,25	202,8	3,26	199,4	6,54	196,9	63,92
T = 7	293,2	0,09	210,7	0,76	206,2	1,07	203,5	2,09	202,6	3,05	200,3	6,10	197,2	59,37
T = 6	308,8	0,09	213,3	0,66	208,6	0,99	202,1	1,92	202,7	2,82	200,2	5,63	196,4	54,54
T = 5	331,1	0,08	212,8	0,60	207,2	0,88	203,3	1,73	203,2	2,54	201,6	5,05	197,4	48,71
T = 4	322,0	0,08	214,4	0,52	210,4	0,77	204,8	1,50	202,2	2,19	200,9	4,37	198,8	41,98
T = 3	299,5	0,07	216,1	0,42	212,1	0,62	205,2	1,19	203,0	1,74	200,3	3,49	196,9	33,50
T = 2	327,2	0,05	236,9	0,28	222,2	0,40	210,0	0,77	203,1	1,11	201,8	2,21	196,7	20,90

### Liite 3: MATLAB-ohjelmistaukset

```
% assign_simanl.m

function bestind = assign_simanl(ind)
%
% Määrätään komponentit koneille käyttäen simuloitua jäädytystä
%
global t m h n p MH;
stop = 0;
numb = 1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Alla muuttujat simuloitua jäädytystä varten %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

T = 4; % Lämpötila T, joka aluksi suuri. Pienenee algoritmin edetessä.
coolingRate = 0.0001; % Jäädytystahti, eli kuinka nopeasti systeemi jäähtyy.
                % Jäädytysaikataulun kerroin (alpha) = 1 - coolingRate
%coolingRate = 0.01;
current_energy = 0; % Nykyisen ratkaisun energiamäärä
new_energy = 0; % Uuden ratkaisun energiamäärä
best_energy = 0; % Historiallisesti parhaan ratkaisun energiamäärä

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Alla etsitään alustava ratkaisu käyttämällä satunnaismetodia %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
while numb < 10
    ind = assign_rnd(ind);
    if ~isempty_ind(ind)
        break
    else
        numb = numb + 1;
    end
end

newind = ind; % luodaan uusi yksilö, joka on kopio alustavan ratkaisun yksilöstä
bestind = ind; % luodaan paras yksilö, joka on kopio alustavan ratkaisun yksilöstä
heads = []; % luodaan tyhjä matriisi heads

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Etsitään konekonfiguraatiosta MH ne koneet, joille on määrätty enemmän kuin %
% 1 ladontapää. %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:h
    headi = find(MH==i); % aseta headi arvoksi machine-head vektorista MH:n i:s elementti
    if size(headi,2) > 1
        heads = [ heads headi ];
    end
end

numheads = size(heads,2); % ladontapäiden lukumäärä

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Suoritetaan algoritmia, kunnes systeemi on jäähtynyt %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

while ~stop
    whichhead = heads(unidrnd(numheads)); % Valitaan satunnainen ladontapää whichhead
    comps = find(ind.X(:,whichhead)); % comps = komponenttityypit, jotka ladotaan
                                        % ladontapäällä whichhead
    if isempty(comps)
```



```

    tries = 0;
    while isempty(comps)
        whichhead = heads(unidrnd(numheads));
        comps = find(ind.X(:,whichhead));
    end
end
whichcomp = comps(unidrnd(size(comps,2))); % Valitaan satunnainen komponentti
                                           % komponenttityyppien comps joukosta

thisheads = find(MH==MH(whichhead));
numh = size(thisheads,2);
headorder = randperm(numh);
for i=1:numh
    whereto = thisheads(headorder(i));
    if (whichhead ~= whereto) && (~is_conflict(ind,whichcomp,whereto))
        howmany = unidrnd(ind.X(whichcomp,whichhead));
        newind = add_xij(ind,-howmany,whichcomp,whichhead);
        newind = add_xij(newind,howmany,whichcomp,whereto);
        newind = nozzle_assignment_onehead(newind,whichhead);
        newind = nozzle_assignment_onehead(newind,whereto);
        break;
    end
end
newind = fitness(newind);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Alla katsotaan onko uusi ratkaisu parempi vai heikompi. Parempi %
% ratkaisu hyväksytään aina. Mikäli uusi ratkaisu on parempi kuin %
% historiallisesti paras ratkaisu, tallennetaan se uudeksi %
% historiallisesti parhaaksi ratkaisuksi. Heikompi ratkaisu %
% voidaan valita nykyiseksi ratkaisuksi hyväksymisfunktioista %
% riippuen. %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

current_energy = get_fitness(ind);
new_energy = get_fitness(newind);
best_energy = get_fitness(bestind);

if new_energy < best_energy
    bestind = newind;
end

if accepted(current_energy, new_energy, T)
    ind = newind;
end

T = T * (1 - coolingRate); % Systeemin jäähtytys (lineaarinen jäähtytysaikataulu)

if (T <= 1) % Algoritmia suoritetaan niin kauan, kun lämpötila T > 1.
    stop = 1;
end
end
end

```

```

% accepted.m

function canbeaccepted = accepted(current_energy, new_energy, T)

%
%ACCEPTED palauttaa tosi, jos uusi ratkaisu voidaan valita nykyiseksi ratkaisuksi.
%

% Mikäli uuden ratkaisun energia-arvo on pienempi kuin vanhan ratkaisun
% energia-arvo, asetetaan P:n arvoksi 1
if new_energy < current_energy
    P = 1;
else
    P = exp(-(new_energy - current_energy) / T);
end

% Mikäli saatu P:n arvo on suurempi kuin generoitava satunnaisluku, voidaan
% uusi ratkaisuehdotus hyväksyä ratkaisuksi
if P >= rand
    canbeaccepted = 1;
else
    canbeaccepted = 0;
end

end

```