

Ville-Matti Anttonen

**TEKOÄLYN SOVELTAMINEN SISÄLOGISTIIKAN AU-
TOMAATIOJÄRJESTELMIEN OHJAUKSESSA**



JYVÄSKYLÄN YLIOPISTO
TIETOJENKÄSITTELYTIETEIDEN LAITOS
2019

TIIVISTELMÄ

Anttonen, Ville-Matti

Tekoälyn soveltaminen sisälogistiikan automaatiojärjestelmien ohjauksessa

Jyväskylä: Jyväskylän yliopisto, 2019, 62 s.

Tietojärjestelmätiede, pro gradu -tutkielma

Ohjaaja: Seppänen, Ville

Sisälogistiikka on osa organisaation logistiikkaa ja toimitusketjun hallintaa, jotka ovat keskeisessä osassa nykypäivän organisaatioiden toimintaa. Tehokas sisälogistiikan hallinta auttaa organisaatiota saavuttamaan etua kilpailijoihinsa nähden. Automaatiojärjestelmät, jotka korvaavat ihmisen tekemän manuaalisen työn koneilla, ovat yksi keino tehostaa sisälogistiikan toimintoja. Viime vuosien nopea kehitys tekoälyn eri teknologioissa ja niiden onnistunut soveltaminen useilla eri toimialoilla herättävät kiinnostuksen siitä, voisiko tekoälyn avulla parantaa myös automaatiojärjestelmien toimintaa sisälogistiikassa. Tämän tutkielman tavoitteena on tutkia kyseistä aihetta ja tutkimuskysymyksenä on selvittää, kuinka tekoälyä voi hyödyntää sisälogistiikan automaatiojärjestelmien ohjauksessa. Tutkielma tehdään toimeksiantona suomalaiselle sisälogistiikan automaatiojärjestelmiä valmistavalle yritykselle. Tutkimusmenetelmänä käytetään Design Science Research Methodology -prosessia. Tutkimuksen tuloksena syntyvällä artefaktilla arvioidaan tekoälyn piiriin kuuluvien geneettisten algoritmien sopivuutta ratkaisemaan tuotesijoitteluongelma yksinkertaistetussa portaalirobotivarastossa. Luotu artefakti osoittaa geneettisten algoritmien potentiaalin reaali maailman automaatiojärjestelmissä, mutta niiden hyödyntäminen tuotantokäytössä vaatii vielä simulaation tarkentamista. Tutkielman tuloksena selviää myös, että tekoälyn käyttöä on pääasiassa tutkittu itseohjautuvia AGV-vaunuja tai korkeavarastoja käyttävissä automaatiojärjestelmissä. Näissä tutkimuksissa on käytetty muun muassa moniagenttijärjestelmiä, geneettisiä algoritmeja ja vahvistusoppimista. Kaiken kaikkiaan aihetta ei ole kovinkaan paljon tutkittu aiemmin. Tutkielma tarjoaa tiettävästi ensimmäisen kirjallisuuskatsauksen, jossa tekoälyn käyttöä olisi yleisesti tarkasteltu sisälogistiikan automaatiojärjestelmissä.

Asiasanat: sisälogistiikka, tekoäly, automaatiojärjestelmä, geneettiset algoritmit, varaston tilausten keräily, automaatiovarasto, AS/RS, Design Science, DSRM

ABSTRACT

Anttonen, Ville-Matti

Master's Thesis: Using Artificial Intelligence in the Controlling of Intralogistics Automation Systems

Jyväskylä: University of Jyväskylä, 2019, 62 pp.

Information Systems Science, Master's Thesis

Supervisor: Seppänen, Ville

Intralogistics is a part of organization's logistics and supply chain management, which are central functions of modern organizations. Effective management of intralogistics helps an organization to gain competitive advantage over its competitors. Automation systems which replace human manual labor with machines are one way to make intralogistics functions more efficient. The rapid advancements of different artificial intelligence technologies and their successful applications in various domains has piqued the interest, whether artificial intelligence could also be used to make automation systems more efficient in the field of intralogistics. The purpose of this thesis is to research the aforementioned topic, and the research question is to figure out how artificial intelligence could be used in controlling intralogistics automation systems. The thesis is done as a commission for a Finnish company which makes intralogistics automation systems. Research method applied was Design Science Research Methodology process. The resulting artifact assesses the feasibility of genetic algorithms, one of the technologies included in the field of artificial intelligence, to solve the warehouse product assignment problem in a simulated portal robot warehouse. The created artifact proves the potential of genetic algorithms in a real world automation systems but using them in a production environment would still require making the simulation more accurate. In addition, the results of the thesis show that the use of artificial intelligence has mainly be researched in automation systems using high bay warehouses or self-driving AGVs. These studies use multi-agent systems, genetic algorithms and reinforcement learning among other technologies. All in all the topic has not received much research interest. As far as is known, the thesis offers the first literature review where the application of artificial intelligence is examined broadly in the domain of intralogistics automation systems.

Keywords: intralogistics, artificial intelligence, automation system, genetic algorithms, warehouse order picking, automated storage and retrieval system, AS/RS, Design Science, DSRM

KUVIOT

KUVIO 1 Esimerkki automaatiojärjestelmästä.....	8
KUVIO 2: Tekoälyn alueita ja teknologioita.....	13
KUVIO 3: Kaksi AGV-vaunua, joista toiseen on lastattu kuormaa.....	20
KUVIO 4: Kaksi portaalirobottisolua, joissa kussakin on kaksi portaalirobottia	21
KUVIO 5: DSRM-prosessi.....	29
KUVIO 6: Geneettisten algoritmien perusrakenne.....	33
KUVIO 7: Varastokartta, jota algoritmista käytetään.....	36
KUVIO 8: Yksinkertaistetun genotyypin ja fenotyypin eroavaisuus ja vastaa- vuus.....	36
KUVIO 9: Etäisyydet varastopaikkoihin punaisella korostetusta poisvientipai- kasta.....	38
KUVIO 10: Vanhempien valinta <i>roulette wheel selection</i> -periaatteen mukaan....	40
KUVIO 11: Lapsikromosomin muodostaminen <i>order 1 crossover</i> -periaatteella..	41
KUVIO 12: Algoritmin tuottama satunnainen ratkaisu.....	42
KUVIO 13: Yksi algoritmin tuottamista ratkaisuista.....	45
KUVIO 14: Esimerkkiratkaisun kelpoisuusfunktion tuloksen kehittyminen.....	46
KUVIO 15: Populaation koon vaikutus kelpoisuusfunktion keskimääräiseen tu- lokseen.....	47
KUVIO 16: Populaation koon vaikutus algoritmin ajoaikaan.....	48
KUVIO 17: Mutaatioiden todennäköisyyden suhteellinen vaikutus tulokseen..	48
KUVIO 18: Mutaatioiden todennäköisyyden vaikutus tulokseen.....	49
KUVIO 19: Esimerkkiratkaisu, kun tarttujan kapasiteetti on 1.....	50
KUVIO 20: Esimerkkiratkaisu, kun tarttujan kapasiteetti on 1 ja jokaista tuotetta on vain yksi pino.....	51

TAULUKOT

TAULUKKO 1 Tekoälyä sisälogistiikan automaatiojärjestelmissä hyödyntäviä tutkimuksia.....	27
TAULUKKO 2: Kelpoisuusfunktion keskimääräinen tulos eri parametrijhdis- telmillä.....	44

SISÄLLYS

TIIVISTELMÄ.....	2
ABSTRACT.....	3
KUVIOT.....	4
TAULUKOT.....	4
SISÄLLYS.....	5
1 JOHDANTO.....	7
2 TEKOÄLY.....	10
2.1 Mitä tekoäly on?.....	10
2.2 Lyhyt historia.....	12
2.3 Tekoälyyn liittyviä teknologioita.....	13
2.3.1 Koneoppiminen, neuroverkot ja syväoppiminen.....	14
2.3.2 Luonnollisen kielen käsittely.....	15
2.3.3 Puheentunnistus.....	15
2.3.4 Asiantuntijajärjestelmät.....	15
2.3.5 Suunnittelu, aikataulutusta ja optimointi.....	16
2.3.6 Robotiikka.....	16
2.3.7 Konenäkö ja kuvantunnistus.....	16
2.4 Yhteenveto.....	17
3 SISÄLOGISTIIKAN AUTOMAATIOJÄRJESTELMÄT.....	18
3.1 Sisälogistiikka.....	18
3.2 Automaatiojärjestelmät.....	19
3.3 Erilaisia sisälogistiikan automaatiojärjestelmiä ja niiden ominaisuuksia.....	20
3.3.1 Tavaroiden siirtäminen.....	20
3.3.2 Tavaroiden välivarastointi.....	21
3.3.3 Tilausten keräily.....	21
3.3.4 Tietojärjestelmät ja ohjelmistot.....	22
3.4 Yhteenveto.....	22
4 TEKOÄLYN HYÖDYNTÄMINEN SISÄLOGISTIIKAN AUTOMAATIOJÄRJESTELMISSÄ.....	23
4.1 Kirjallisuuskatsaus.....	23
4.1.1 Tilauksen keräilyongelma.....	24
4.1.2 Moniagenttijärjestelmät.....	24
4.1.3 Geneettiset algoritmit.....	25

4.1.4 Industry 4.0 ja Logistics 4.0.....	25
4.1.5 Muut esille nousseet teemat ja tutkimukset.....	26
4.2 Yhteenveto.....	26
5 TUTKIMUSMENETELMÄ.....	28
5.1 Design Science ja DSRM-prosessi tietojärjestelmätieteessä.....	28
5.2 DSRM-prosessi sovellettuna tähän tutkimukseen.....	29
5.2.1 Ongelman tunnistaminen ja motivointi.....	29
5.2.2 Ratkaisun tavoitteet.....	30
5.2.3 Suunnittelu ja toteutus.....	30
5.2.4 Demonstrointi.....	30
5.2.5 Arviointi.....	30
5.2.6 Kommunikointi.....	31
5.3 Aineisto.....	31
6 ARTEFAKTIN SUUNNITTELU JA TOTEUTUS.....	32
6.1 Artefaktin tavoitteet.....	32
6.2 Geneettiset algoritmit.....	32
6.2.1 Teoria.....	33
6.2.2 Geneettiset algoritmit sovellettuna portaalirobottivarastoon....	34
6.3 Tietorakenne varastokartan kuvaamiseen.....	35
6.4 Kelpoisuusfunktio.....	37
6.4.1 Tilaukset kelpoisuusfunktion syötteenä.....	37
6.4.2 Tilausten keräily.....	37
6.4.3 Pisteiden laskenta.....	38
6.5 Algoritmin toiminta ja parametrit.....	39
6.5.1 Algoritmin alustus.....	39
6.5.2 Vanhempien valinta ja jälkikasvun muodostaminen.....	39
6.5.3 Mutaatiot.....	41
6.5.4 Toteutuksessa käytetyt teknologiat.....	41
6.6 Algoritmin tuottama ratkaisu.....	41
7 ARTEFAKTIN DEMONSTROINTI JA ARVIOINTI.....	43
7.1 Demonstrointi.....	43
7.2 Tulokset ja arviointi.....	44
7.2.1 Populaation koon vaikutus algoritmin toimintaan.....	46
7.2.2 Mutaatioiden todennäköisyyden vaikutus algoritmin toimintaan.....	48
7.2.3 Muiden parametrien vaikutukset.....	49
8 POHDINTA.....	52
9 YHTEENVETO.....	55
LÄHTEET.....	58

1 JOHDANTO

Sisälogistiikan automaatiojärjestelmät ovat keskeinen osa monien nykypäivän tehtaiden, yritysten ja instituutioiden sisätilojen materiaalinhallintaa. Sisälogistiikka on osa laajempaa logistiikan käsitettä, mutta se keskittyy nimensä mukaisesti sisätilojen materiaalinhallintaan ja lisäksi sisälogistiikan järjestelmillä on selkeä rajapinta ulkopuolisiin järjestelmiin (Nagel ym., 2008). Organisaatioiden logistiikan hallinnalla voi olla merkittävä vaikutus kilpailuedun saavuttamiseen muihin organisaatioihin nähden (Christopher, 2011, s. 2), joten logistiikan eri osa-alueiden toiminnan parantaminen ja niiden vaikutus liiketoimintaan tulisi olla huomioituna organisaatioissa.

Tekoäly on ollut yksi viime vuosien puhutuimmista trendeistä, joka näyttää levittäytyvän yhteiskuntamme lähes jokaiselle osa-alueelle: älypuhelimien henkilökohtaisista assistenteista¹ itseohjautuviin autoihin² ja yritysten johtoryhmien jäseneksi³. Tekoäly on 1950-luvun alkuajoistaan elänyt useita nykypäivän kaltaisia täynnä optimismia olevia jaksoja, mutta toisaalta myös ajanjaksoja, jolloin on huomattu, että tekoäly ei ainakaan vielä täytä sille asetettuja suuria odotuksia (Russell & Norvig, 1995). Pessimistisistä ajanjaksoista huolimatta tekoälyyn liittyvät teknologiat ovat kehittyneet vuosikymmenten kuluessa ja tekoälyä hyödyntämällä on VTT:n raportin mukaan mahdollista kasvattaa tuottavuutta merkittävästi (VTT, 2017). PwC:n raportti tunnistaa, että tekoälyä on mahdollista hyödyntää organisaatioiden lähes kaikilla alueilla, mukaan lukien toimitusketjun ja logistiikan hallinnassa (PwC, 2018).

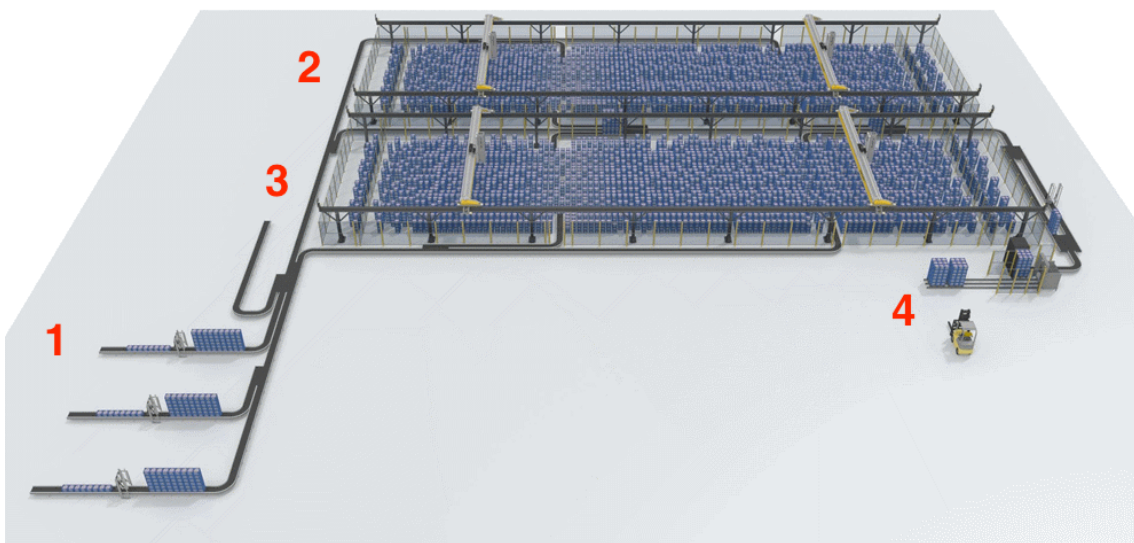
Tässä tietojärjestelmätieteen pro gradussa käsitellään kahta edellä mainittua käsitettä: sisälogistiikan automaatiojärjestelmiä ja tekoälyä. Tarkoituksena on selvittää, miten tekoälyä on mahdollista hyödyntää sisälogistiikan automaatiojärjestelmissä ja miten sitä on tähän mennessä hyödynnetty kyseisellä alueella. Tutkielma tehdään toimeksiantona Cimcorp Oy:lle, joka tekee sisälogistiikan automaatiojärjestelmiä eri teollisuuden aloille, kuten elintarviketehtaiden ja vähittäiskaupan varastoihin ja lähettämöihin, sekä rengasteollisuuteen (Cimcorp, 2019a).

¹ <https://www.apple.com/siri/>

² <https://waymo.com/>

³ <https://www.tieto.com/fi/uutishuone/kaikki-uutiset-ja-tiedotteet/blogit/2018/tekoaly-suorittaa-ihminen-johtaa/>

Tutkimusongelma voidaan havainnollistaa seuraavan kuvion avulla (kuvio 1), jossa on esimerkki tehtaan automaatiojärjestelmästä. Tehtaan kultakin tuotantolinjalta tulee tietyn tyyppistä tuotetta muovilaatikkopinoina varastoon. Esimerkkikuvan kohdassa 1 on kolme tuotantolinjaa, josta kustakin tulee tietyn tyyppistä tuotetta. Tuotantolinjoilta tulevat laatikkopinot varastoidaan portaali-robottisoluihin. Kuvassa on kaksi portaalirobottisolua (kohdat 2 ja 3), joissa kussakin on kaksi portaalirobottia (engl. gantry robot). Robotit ovat kuviossa harmaakeltaisia ja ne ovat kiskojen päällä varastoalueen yläpuolella solujen vasemmalla ja oikealla puolella. Robottien tehtävänä on keräillä varastolta tilauksia. Tyypillisesti yksi tilaus koostuu useasta eri tuotteesta. Kun tilauksen kaikki tuotteet ovat keräilty, ne kootaan kuormalavoille, joilla ne siirretään eteenpäin (kohta 4), esimerkiksi kuvion mukaisesti trukilla.



KUVIO 1 Esimerkki automaatiojärjestelmästä

Etukäteen ei kuitenkaan tiedetä, mistä eri tuotteista kukin tilaus koostuu. Ongelmana on siis selvittää, miten tuotantolinjoilta tulevat tuotteet kannattaa sijoittaa varastoon ja miten tilaukset kannattaa jakaa robottien kesken, jotta robottien tilausten keräily olisi mahdollisimman tehokasta.

Tehokkaimmillaan keräily onnistuisi, jos yksi robotti voisi keräillä kokonaan yhden keräilytilauksen, esimerkiksi yhden kuormalavan. Usein käy kuitenkin niin, että yhden robotin toiminta-alueelta ei löydy vaadittavaa tuotetta ja muut robotit joutuvat osallistumaan tilauksen keräilyyn. Tämä laskee tehokkuutta, koska robotit joutuvat synkronoimaan toimintaansa toistensa mukaan tai robotit joutuvat siirtämään laatikoita toisilleen. Monesti keräilyalgoritmit ovat tällä hetkellä kiinteitä, jotka määritetään kunkin projektin kuluessa. Asiakkaan toimintaympäristö, kuten tilausrakenne, voi kuitenkin muuttua, jolloin kiinteät algoritmit eivät ole enää optimaalisia muuttuneeseen tilanteeseen. Tekoälyä hyödyntämällä voitaisiin mahdollisesti luoda oppivia algoritmeja keräilyyn, jotka pystyisivät mukautumaan vallitsevaan tilanteeseen ja ennustamaan tulevia tilauksia aiemman historiatiedon perusteella. Näin robottien keräily olisi mahdollista pitää tehokkaampana kuin kiinteillä algoritmeilla.

Tutkimusongelmana on siis selvittää, voidaanko tekoälyä soveltamalla parantaa sisälogistiikan automaatiojärjestelmien ohjausta. Tavoitteena on vastata kysymykseen:

- Miten tekoälyä voi soveltaa sisälogistiikan automaatiojärjestelmien ohjauksessa?

Tutkimus toteutetaan suunnittelutieteellisenä tutkimuksena käyttäen Design Science Research Methodology (DSRM) -prosessia. Tutkimuksen tuloksena syntyvässä artefaktissa ratkaistaan varaston tuotesijoitteluongelma portaalirobotivarastossa käyttäen *geneettisiä algoritmeja*. Tutkielmassa käydään myös kirjallisuuskatsauksena läpi aikaisempi tutkimus aiheesta, jonka avulla saadaan karotettua laajemmin tekoälyn hyödyntämismahdollisuudet. Kirjallisuuskatsauksen tuloksena selvisi, että aiempi kiinnostus aiheeseen on ollut vähäistä ja tekoälyn käyttöä on tutkittu lähinnä itseohjautuvien AGV-vaunujen ja korkeavarastojen kontekstissa. Eniten käytettyjä tekoälyn teknologioita olivat näissä tutkimuksissa moniagenttijärjestelmät, geneettiset algoritmit ja vahvistusoppiminen. Tutkielma tarjoaa tiettävästi ensimmäisen kirjallisuuskatsauksen, jossa tekoälyn käyttöä olisi yleisesti tarkasteltu sisälogistiikan automaatiojärjestelmissä.

Toteutetun artefaktin teoreettisena pohjana on geneettiset algoritmit, joita todettiin kirjallisuuskatsauksen perusteella olevan hyödynnetty muiden tyyppisissä sisälogistiikan automaatiojärjestelmissä, kuin portaalirobotivarastoissa. Koska kaiken tyyppiset sisälogistiikan automaatiojärjestelmät ratkaisevat pohjimmiltaan samankaltaisia ongelmia, geneettisten algoritmien oletettiin sopivan myös portaalirobotivarastoon. Artefaktin demonstroinnista saatujen tulosten perusteella geneettisten algoritmien arvioitiin ratkaisevan tuotesijoitteluongelman riittävän hyvin simuloitussa yksinkertaistetussa portaalirobotivarastossa. Simulaation rajoitteet huomioon ottaen geneettisiin algoritmeihin perustuva tuotesijoittelu osoittautui potentiaaliseksi myös reaali maailman automaatiojärjestelmissä.

Tutkielman toinen ja kolmas luku keskittyvät tutkimuksen aihealueen keskeisten käsitteiden selventämiseen ja määrittämiseen. Toisessa luvussa annetaan tarkempi johdatus tekoälyyn ja siihen liittyviin teknologioihin. Kolmas luku käsittelee sisälogistiikan automaatiojärjestelmiä. Neljännessä luvussa esitellään aiheeseen liittyvä aikaisempi tutkimus ja tapoja, kuinka tekoälyä voisi hyödyntää tutkimuksen kohteena olevissa automaatiojärjestelmissä. Viidennessä luvussa esitellään tutkimusmenetelmänä käytetty DSRM-prosessi ja sen soveltaminen tutkielman aihealueeseen. Kuudennessa ja seitsemännessä luvussa käsitellään artefaktin suunnittelua ja toteutusta, sekä demonstroitua ja arviointia. Tutkielma päättyy pohdintaan ja yhteenvetoon, jossa tarkastellaan tutkimusta kriittisesti ja arvioidaan siihen liittyviä rajoitteita. Lopuksi ehdotetaan mahdollisia jatkotutkimusaiheita.

2 TEKOÄLY

Tämän luvun tarkoituksena on tutustuttaa lukija tekoälyn käsitteeseen. Aluksi tarkastellaan eri määritelmiä tekoälylle. Sen jälkeen esitetään tekoälyn lyhyt historia, jonka avulla tekoälyn nykytilan ymmärtäminen on helpompaa. Lopuksi perehdytään tarkemmin muutamaa keskeiseen tekoölyyn liittyvään teknologiaan.

2.1 Mitä tekoäly on?

Tekoälylle ei ole yleisesti hyväksyttyä määritelmää, jonka seurauksena ihmisillä saattaa olla hyvin erilaisia käsityksiä siitä, mitä tekoäly on. Asiaa ei helpota myös sekään, että vuosikymmenten aikana kirjallisuudessa, elokuvissa ja muussa mediassa tekoölyyn liitetään vahvasti ihmistä muistuttavat humanoidi-robotit, mikä on helposti todennettavissa internetin hakukoneiden kuvahaulla tekoälystä. Toisaalta tekoälyn ymmärtämisen saattaa tehdä myös haastavaksi se, että usein on vaikeaa käsittää minkälaiset tehtävät ja ongelmat ovat koneille helppoja ja mitkä eivät. Esimerkiksi shakin peluu ammattilaistasolla vaati ihmiseltä vuosien opettelu, mutta onnistuu tietokoneilta verrattain helposti. Ihmiseltä esineiden hienovarainen poimiminen ja käsittely onnistuvat luonnostaan, mutta roboteille ja niitä ohjaaville tietokoneille ne ovat erittäin vaikeita tehtäviä (Elements of AI, 2018).

Useimmat määritelmät liittyvät tietokoneohjelmien tai koneiden luomiseen, jotka kykenevät toimimaan tavalla, jota kutsuttaisiin älykkääksi, jos ihminen tekisi saman. Vaikeuksia tekoälyn määrittämisessä aiheuttaa erityisesti käsitteen "äly" hankala määriteltävyys ja toisaalta se, että tekoälyn ajatellaan usein liittyvän ihmisen älykkyyteen. Ihmisen älykkyyden mittaaminen on vaikeaa, johon pelkkä älykkyydosamäärä ei riitä: yksi voi olla esimerkiksi musiikillisesti lahjakas ja toinen kehohallinnallisesti. (Kaplan, 2016, s. 1-4).

Kaplan (2016, s. 5) esittää tekoälyn ytimen olevan kyvyssä tehdä järkeenkäyviä yleistyksiä rajallisesta datasta kohtuullisessa ajassa. Toiminta on sitä älykkäämpää, mitä laajemmilla alueilla sitä voi käyttää, mitä nopeammin pää-

töksenteko tehdään, ja mitä vähemmällä tiedolla se kyetään saavuttamaan. Esimerkiksi monikaan ei pitäisi yksinkertaista ristinollapeliä pelaavaa ohjelmaa tekoälynä, mutta jos sama ohjelma kykenisi oppimaan pelaamaan myös shakkia tai mitä tahansa muuta lautapeliä, tilanne olisi toinen (Kaplan, 2016, s.5).

Borana (2016) tarjoaa konkreettisemmän määritelmän, jonka mukaan tekoäly on laaja käsite, joka viittaa keinotekoisien entiteettien, yleensä tietokoneen tai koneen, älykkyyteen ratkaista monimutkaisia ongelmia. Älykkyydellä tarkoitetaan muun muassa kykyä ajatella, muistaa, oppia, ymmärtää, tehdä valintoja, sekä mukautua (Borana, 2016).

Russell ja Norvig (1995) esittelevät tunnetussa kirjassaan joukon määritelmiä tekoälylle, jotka he jaottelevat neljään ryhmään, eli järjestelmiin jotka:

- ajattelevat kuten ihmiset
- käyttäytyvät kuten ihmiset
- ajattelevat rationaalisesti
- käyttäytyvät rationaalisesti

Osa määritelmistä painottavat siis ihmismäistä älykkyyttä ja osa rationaalisuutta, jolla viitataan oikean ratkaisun tekemiseen (Russell & Norvig, 1995, s. 4-5). Kirjassaan he omaksuvat kannan, jossa tekoäly on pääasiassa rationaalista toimintaa ja ideaalitulanteessa *älykäs agentti* toimii parhaimmalla mahdollisella tavalla kussakin tilanteessa (Russell & Norvig, 1995, s. 27). Bringsjord ja Schimanski (2003) eivät kuitenkaan ole vakuuttuneita Russellin ja Norvigin tavasta jakaa tekoälyn määritelmät, koska sanan "ajattelu" määrittäminen tulisi todennäköisesti aiheuttamaan yhtä paljon päänvaivaa kuin älyn määrittäminen. He ovat sen sijaan Kaplanin tavoin yhtä mieltä siitä, että tekoälyä määriteltäessä sana "äly" on avainasemassa ja paljon hankalammin määriteltävissä kuin sana "teko", jolla yksinkertaisesti viitataan johonkin keinotekoiseen. (Bringsjord & Schimanski, 2003).

Russellin ja Norvigin rationaalisesti toimiva agentti havainnoi ympäristöään sensoreilla ja käyttää efektoreja toimiakseen ympäristössä (Russell & Norvig, 1995; s. 31). Sensorina voi toimia esimerkiksi kamera ja efektorina servomoottori. Kuten edellisessä kappaleessa mainittiin, rationaalisuus viittaa oikean ratkaisun tekemiseen. Oikeaksi ratkaisuksi voidaan määrittää se toiminta, jonka seurauksena agentin toiminta on kaikkein onnistunein. Toiminnan onnistumisen arvioimiseksi pitää määrittää standardi, jonka perusteella agentin toimintaa voidaan arvioida, sekä ajankohdat jolloin arviointi suoritetaan. (Russell & Norvig, 1995; s. 31-32).

Tekoäly voidaan luokitella vahvaan ja heikkoon tekoälyyn. Ajatus vahvasa tekoälyssä on se, että tietokone kykenee ajattelemaan ja ymmärtämään kuten ihmiset (Searle 1980). Tällöin koneet kykenisivät kaikkeen mihin ihminenkin kykenisi. Nykyiset tekoälyn sovellukset ovat vielä kaukana tämänkaltaisesta ja on useita mielipiteitä siitä, onko vahvaa tekoälyä edes mahdollista saavuttaa (Borana, 2016). Paradoksaalista kuitenkin on, että tekoälyksi usein mielletään sellainen mitä nykyajan tietokoneilla ei olla vielä kyetty saavuttamaan, mutta kun asiaan löydetäänkin ratkaisu, se mielletään tekoälyn sijasta jo tunnettuina olleiden menetelmien ja algoritmien kokoelmana (VTT, 2017). Kaikki nykyiset

tekoälyn sovellukset voidaan luokitella kuuluvan heikon tekoälyn piiriin: ne keskittyvät tiettyyn, kapeaan aihealueeseen, eivätkä ne kykene toimimaan sen ulkopuolella (VTT, 2017).

Keskeisiä tekoälyn ominaisuuksia ovat autonomisuus ja adaptiivisuus (Elements of AI, 2018). Autonomisuudella viitataan kykyyn suorittaa tehtäviä monimutkaisissa ympäristöissä ilman käyttäjän jatkuvaa valvontaa ja ohjausta, eli käyttäytymisen määrittää tekoälyentiteetin oma aiempi kokemus (Elements of AI, 2018; Russell & Norvig, 1995, s. 35). Adaptiivisuudella viitataan kykyyn oppia kokemuksen perusteella ja parantaa näin tehokkuutta (Elements of AI, 2018). Tekoälyn käsittäminen autonomisuuden ja adaptiivisuuden avulla tarjoaa helpommin lähestyttävämmän tavan ymmärtää tekoälyn olemusta, koska niiden määrittäminen on yksikäsitteisempää kuin älyn tai ajattelun määrittäminen. Kyky toimia itsenäisesti ja oppia aiemman kokemuksen perusteella, yhdistettynä Russellin ja Norvigin rationaalisesti toimivaan älykkääseen agenttiin, tarjoaa käyttökelpoisen määritelmän tekoälylle.

2.2 Lyhyt historia

Tekoäly tieteenalana on varsin nuori, mutta siihen ovat vaikuttaneet monet teorit, ideat ja tekniikat jo vuosisatojen takaa filosofiasta, matematiikasta, psykologiasta, tietotekniikasta ja kielitieteistä. Näihin lukeutuvat muun muassa teorioita oppimisesta, päättelystä, logiikasta, päätöksenteosta, sekä kielten merkityksestä ja rakenteesta. (Russell & Norvig, 1995, 8-16). Viime vuosisadan puolivälistä lähtien tietokoneiden huikkea kehityskulku on mahdollistanut tekoälyn siirtymisen teorian tasolta käytäntöön.

Tekoäly terminä juontaa juurensa vuoteen 1956, jolloin John McCarthy muiden alan pioneerien, kuten Marvin Minskyn, Claude Shannonin ja Nathan Rochesterin kanssa, organisoivat aiheeseen liittyen konferenssin Dartmouthissa, Yhdysvalloissa. Tuolloin McCarthy ehdotti alalle nimeksi tekoälyä, erottaen sen jo paljon kypsyneemmästä kybernetiikan alasta. (Kaplan, 2016, s. 13). Varsinainen ensimmäinen tekoälyyn liittyvä työ on kuitenkin jo vuodelta 1943, jolloin Warren McCulloch ja Walter Pitts esittivät mallin keinotekoisista neurooneista (Russell & Norvig, 1995, s. 17).

Ensimmäiset vuosikymmenet tekoälyn historiassa olivat täynnä optimismeja, jolloin saavutettiin monia merkkipaaluja ja ala sai paljon rahallista tukea eri instituutioilta, kuten Yhdysvaltojen puolustusministeriön alaiselta DARPALta. 1950-luvulla Arthur Samuel loi tammea pelaavan ohjelman, joka kykeni oppimalla parantamaan suoritustaan ja saavutti lopulta huippupelaajan tason, sekä osoitti että tietokoneohjelma kykenee oppimaan pelaamaan peliä paremmin kuin ohjelman luoja. 1960-luvun lopulla Stanford Research Institutun kehittämä Shakey-robotti⁴ yhdisti ensimmäistä kertaa tekoälyohjelman fyysiseen liikkeeseen. Optimismi kuitenkin laantui, kun huomattiin että tekoälyn soveltaminen laajempiin ja monimutkaisempiin ongelmiin ei onnistunut enää pelkästään

⁴ <http://www.ai.sri.com/shakey/>

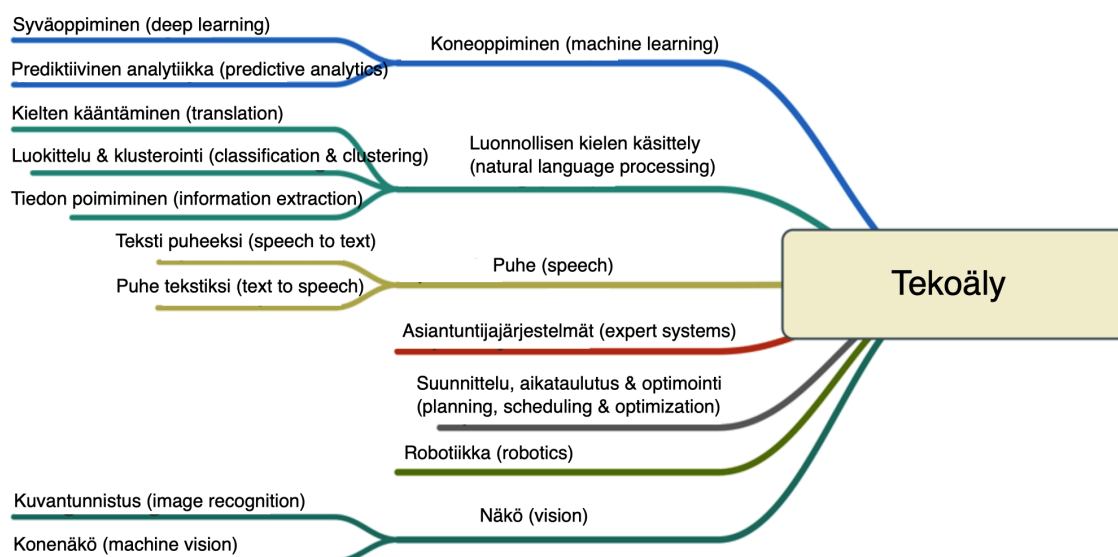
paremmalla laitteistolla. (Kaplan, 2016 s. 17-19; Russell & Norvig, 1995, s. 17-21).

Laajamittaiset tekoälyn kaupalliset sovellukset syntyivät 1980-luvulla asiantuntijajärjestelmien muodossa. Ne ovat yleensä hyvin kapealle kohdealueelle suunnattuja järjestelmiä, jotka tekevät päätelmiä järjestelmään syötettyjen kohdealueen faktojen, sääntöjen ja riippuvuussuhteiden perusteella. Asiantuntijajärjestelmät ovat nykypäivänäkin laajassa käytössä, mutta niiden ei enää katsota olevan aktiivisen tutkimuksen kohteena. (Kaplan, 2016 s. 22-24; Russell & Norvig, 1995, s. 24).

1990-alkupuolelta lähtien koneoppiminen ja sen eri lähestymistavat, kuten neuroverkot ja syväoppiminen, ovat herättäneet yhä kasvavammassa määrin kiinnostusta sekä tutkimuksessa että kaupallisissa sovelluksissa. Koneoppimisen nousuun ovat erityisesti vaikuttaneet tietokoneiden laskentakapasiteetin kasvu, datan saatavuus digitaalisesti ja digitaalisten sensoreiden kehittyminen, joilla dataa voidaan kerätä. 1990- ja 2000-luvulla saavutettiin myös monia tunnettuja merkkipaaluja: IBM:n Deep Blue voitti silloisen shakkimestari Garry Kasparovin vuonna 1997, IBM:n Watsonin vuoden 2011 voitto TV-visailuohjelma Jeopardyn parhaimmista pelaajista ja viimeisimpänä Googlen AlphaGo, joka voitti parhaimman go-pelaaja Lee Sedolin vuonna 2016. (Kaplan, 2016; s. 40-43).

2.3 Tekoölyyn liittyviä teknologioita

Alla olevassa kuviossa on esitetty kokonaiskuva tekoölyyn liittyvistä alueista ja teknologioista (kuvio 2). Tässä osiossa käydään läpi kuvion osa-alueet yleisellä tasolla.



KUVIO 2: Tekoölyn alueita ja teknologioita (Villanueva & Salenga, 2018, mukaillen)

2.3.1 Koneoppiminen, neuroverkot ja syväoppiminen

Ihmisen kykyä oppia on pitkään pidetty yhtenä älykkyyden keskeisenä osana. Kysymyksenä on selvittää, miten ihminen oppii ja voiko tietokone oppia samalla tavalla? Keskeistä oppimisessa ajatellaan päättelyn lisäksi olevan esimerkkien ja harjoituksen avulla oppiminen. (Kaplan, 2016, s. 27). Tähän perustuen, koneoppimisen tavoitteena on pyrkiä tekemään esimerkkidatan perusteella yleistyksiä, joita voidaan hyödyntää hyvin monenlaisiin tarkoituksiin (Domingos, 2012) ja jotka paranevat kokemuksen myötä (Jordan & Mitchell, 2015). Vaikka koneoppiminen ei ole uusi suuntaus tekoälyn alalla, kiinnostus siihen on vasta viime aikoina kasvanut erityisesti siksi, että digitaalisesti saatavan datan määrä on kasvanut valtavasti ja hinta suorittaa massiivista laskentaa on laskenut. Koneoppiminen on noussut suosituksi tavaksi ratkaista monia tekoälyyn liittyviä ongelmia, kuten luonnollisen kielen käsittelyä, konenäköä, puheentunnistusta ja robotiikkaa. (Jordan & Mitchell, 2015).

Koneoppimisessa on kolme pääajatusmallia: ohjattu oppiminen, ohjaamaton oppiminen ja vahvistusoppiminen. Ohjatun oppimisen (engl. supervised learning) keinot ovat kaikkein laajimmin käytettyjä koneoppimisessa ja niitä käytetään esimerkiksi sähköpostien roskapostisuodattimissa. (Jordan & Mitchell, 2015). Ohjatussa oppimisessa opetusdatasta tiedetään oikea lopputulos etukäteen. Ohjaamattomassa oppimisessa (engl. unsupervised learning) opetusdatasta ei tiedetä mitään. (Salinas, 2017). Esimerkkinä ohjaamattomasta oppimisesta on konenäköön liittyen kuvien luokittelu niiden sisällön perusteella. Vahvistusoppimista (engl. reinforcement learning) voidaan kuvata oppimiseksi yrityksen ja erehdyksen kautta. Vahvistusoppimiseen perustuva järjestelmä saa harjoitusdatasta ainoastaan tiedon, onko järjestelmän suorittama toiminta oikein vai ei. Jos toiminto on väärä, ongelmaksi jää oikean toiminnon löytäminen, ja jos toiminto on oikein, se ei välttämättä ole paras ratkaisu pitkällä aikavälillä. (Kaelbling ym., 1996). Kolmen pääajatusmallin lisäksi tutkimuksen kohteena on muitakin tapoja, muun muassa puoliohjattu oppiminen, joka yhdistää ohjatun oppimisen datan ohjaamattoman oppimisen dataan ja saavuttaa näin parempia tuloksia ohjaamattoman oppimisen datan luokittelussa (Zhu, 2005).

Keinotekoiset neuroverkot ovat koneoppimisen osa-alue, joka pyrkii ratkaisemaan oppimisen ongelmaa mukaillemalla biologisten neuroverkkojen, kuten ihmisen aivojen rakennetta (Jain ym., 1996). Keinotekoiset neuroverkot koostuvat suuresta määrästä toisiinsa kytkeytyneitä prosessointielementtejä, eli neuroneita. Oppiminen tapahtuu säätämällä neuronien välisten yhteyksien voimakkuutta. (Stergiou & Siganos, 1997). Neuroverkot voivat koostua useista kerroksista syöte- ja ulostulokerroksen välillä. Eteenpäin kytkeytyessä neuroverkossa signaalit kulkevat verkossa vain yhteen suuntaan ja takaisinkytketyssä verkossa signaalit voivat kulkea molempiin suuntiin (Jain ym., 1996).

Syväoppiminen viittaa joukkoon koneoppimisen tekniikoita, joissa hyödynnetään useista kerroksista koostuvia ei-lineaaraisia tiedonkäsittelytasoja tai hierarkkisia arkkitehtuureja (Yu & Deng, 2012). Esimerkiksi syvät keinotekoiset neuroverkot kuuluvat syväoppimisen piiriin. Syvissä neuroverkoissa on useita syöte- ja ulostulokerroksen välillä olevia piilokerroksia, jotka erottavat ne tavallisista neuroverkoista. Syvät neuroverkot ovat osoittautuneet erittäin toimiviksi

ratkaisuiksi ja ne ovat muun muassa voittaneet useita kansainvälisiä koneoppimisen ja hahmontunnistuksen kilpailuja (Schmidhuber, 2015).

2.3.2 Luonnollisen kielen käsittely

Luonnollisen kielen käsittely viittaa tieteenalaan, jossa tavoitteena on tietokoneiden avulla pyrkiä ymmärtämään, tuottamaan, manipuloimaan ja oppimaan luonnollista puhetta tai tekstiä. Ala pyrkii selvittämään, kuinka ihmiset ymmärtävät kieltä ja tätä tietoa hyödyntäen kehittämään tekniikoita ja työkaluja tietokoneita varten. (Chowdhury, 2003; Hirschberg & Manning, 2015). Kieltä tulkittaessa tietokoneiden täytyy kyetä tarkastelemaan kieltä monella eri tasolla, useiden lauseiden tasolta kontekstin selvittämiseksi aina yksittäisten sanojen päätteisiin. Monilla sanoilla on useampia merkityksiä ja lausetasolla sanajärjestyksestä riippuen lauseen merkitys saattaa muuttua. Nykyisin luonnollisen kielen käsittelyssä parhaita tuloksia on saavutettu käyttämällä koneoppimisen keinoja. (Hirschberg & Manning, 2015).

2.3.3 Puheentunnistus

Tarkka puheentunnistus vaatii ratkaisujen löytämistä monenlaisiin ongelmiin. Ensinnäkin itse puhe täytyy erottaa ympäröivästä taustahälystä. Toisekseen, sanat ja lauseet pitää osata erottaa toisistaan ja niiden merkitys tulkita oikein. Tämä ei ole yksinkertainen tehtävä ja siihen liittyy samaa problematiikkaa kuin luonnollisen kielen käsittelyssä: eri kielissä intonaatio voi muuttaa koko sanan tai lauseen merkityksen ja joidenkin sanojen lausumistapa saattaa olla sama vaikka niiden merkitys on eri. (Kaplan, 2016, s. 58). Näitä ongelmia on pyritty ratkaisemaan muun muassa Markovin piilomallin (engl. hidden Markov model) ja uusiutuvien syväneuroverkkojen (engl. deep recurrent neural network) avulla (Rabiner, 1989; Graves ym., 2013). Tämän päivän tunnetuimpia puheentunnistuksen ja luonnollisen kielen käsittelyn sovelluksia ovat älypuhelimien henkilökohtaiset assistentit, jotka pystyvät esimerkiksi luomaan kalenterimerkintöjä käyttäjän puheesta ja jopa käymään yksinkertaisia keskusteluja.

2.3.4 Asiantuntijajärjestelmät

Asiantuntijajärjestelmien pääajatus on siinä, että ihmisen kohdealueen asiantuntemus siirretään tietokoneelle, joka tiedon avulla pystyy tekemään päätelmiä (Liao, 2005). Asiantuntijajärjestelmät koostuvat tietopankista, johon kaikki kohdealueen tieto on koottu ja päättelykoneesta, joka käyttää tietopankkia johdopäätösten tekemiseen. Tietopankin kerääminen on työlästä ja vaatii usein manuaalista työtä. Viime vuosikymmenten aikana tietokoneiden tehon ja tiedontallennuskapasiteetin kasvu, sekä internetin ja muiden tietoverkkojen kehittyminen ovat vähentäneet perinteisten asiantuntijajärjestelmien houkuttelevuutta. (Kaplan, 2016, s. 22-24).

2.3.5 Suunnittelu, aikataulutus ja optimointi

Suunnittelun tavoitteena on päästä tietystä lähtötilasta haluttuun tavoitetilaan mahdollisimman tehokkaasti. Automaattinen suunnittelujärjestelmä tuottaa suunnitelman, joka sisältää toiminnot joita noudattamalla tavoitetilaa voidaan saavuttaa. Suunnittelujärjestelmän osana tai sen yhteydessä voi olla aikataulutaja, joka määrittää milloin ja miten toiminnot suoritetaan. (Nau, 2007). Sekä suunnittelu että aikataulutus ovat tyypillisesti NP-täydellisiä ongelmia ja useissa tapauksissa ratkaisuun käytetään heuristiikkoja (Nau, 2010; Kaplan, 2016 s. 25). Automaattisia suunnittelujärjestelmiä on kolmenlaisia. Kohdealuepesifiset järjestelmät ovat kaikkein yleisimpiä ja ne ovat tehty tiettyä kohdealuetta varten, eivätkä ne siten sovellu muiden kaltaisten suunnitteluongelmien ratkaisuun. Kohdealueerippumattomat järjestelmät ovat yleisluontoisia ja ne saavat syötteenä kuvauksen suunnitteluongelmasta ja järjestelmä tuottaa siihen ratkaisun. Kohdealuekonfiguroitavat järjestelmät sisältävät myös yleisluontoisen ratkaisijan, mutta saavat syötteenä suunnitteluongelman lisäksi kohdealuepesifistä tietoa, jonka avulla järjestelmän hakuavaruutta on mahdollista pienentää. Kohdealuepesifiset järjestelmät ovat suorituskyvyltään parhaimpia, mutta niiden konfigurointi vaatii eniten resursseja eikä niitä voi käyttää muissa kohdealueissa. Kohdealueerippumattomat ja -konfiguroitavat järjestelmät ovat yleiskäyttöisempiä, mutta niiden tehokkuus on matalampi, koska kaikkia kohdealuepesifejä optimointeja ei voida käyttää. (Nau, 2007).

2.3.6 Robotiikka

Robotiikan alalla tekoälyn haasteena on toimia fyysisten objektien kanssa. Robotiikan voidaan ajatella toimivan linkkinä havainnoinnin ja fyysisen toiminnan välillä. Fyysinen maailma tuo monenlaisia haasteita, jotka robotiikassa pitää huomioida. Näitä ovat esimerkiksi voimankäytön hallitseminen, epätarkkuuksien sietäminen ja objektien geometrinen mallintaminen. (Brady, 1985). Tavanomaisesti robotiikalla on automatisoitu tehtäviä, jotka ovat samankaltaisia luonteeltaan ja joissa käsitellään pitkälti samankaltaisia reaali maailman objekteja. Tekoälyllä on mahdollista luoda laitteita, jotka pystyvät suoriutumaan paljon monipuolisimmista tehtävistä, jotka liittyvät esimerkiksi eri muotoisten ja painoisten, sekä haurausasteeltaan vaihtelevien objektien käsittelyyn. Lisäksi tekoäly mahdollistaa erityisesti sellaiset robotiikan sovellukset, jotka kykenevät toimimaan ympäristöissä, jotka ovat ihmiselle liian vaarallisia tai muulla tavoin saavuttamattomissa. (Kaplan, 2016, s. 49-50). Tämän päivän tutkimuskohteita tekoälyn hyödyntämisessä robotiikassa ovat muun muassa ominaisuuksiltaan erilaisten objektien poimiminen, robottien toiminnan parantaminen erilaisissa ympäristöissä, kuten epätasaisessa maastossa, sekä robottien ohjauksen ja ohjelmoinnin parantaminen (IFR, 2018).

2.3.7 Konenäkö ja kuvantunnistus

Konenäön ja kuvantunnistuksen tavoitteena on tekoälyn avulla tulkita kuvia. Modernit menetelmät perustuvat koneoppimiseen piiriin kuuluviin konvoluuti-

tioneuroverkkoihin (engl. convolutional neural networks) (Simonyan & Zisserman, 2014) ja niiden avulla pystytään tunnistamaan kuvia noin 95 prosentin tarkkuudella (Russakovsky ym., 2015). Yksi tapa ajatella konenäköä on, että syötteenä annetaan kaksiulotteinen kuva kolmiulotteisesta maailmasta ja tavoitteena on tulkita syöte kolmiulotteisena ja tehdä sen avulla päätelmiä (Kaplan, 2016, s. 55). Konenäköä ja kuvantunnistusta voi soveltaa monenlaisilla alueilla. Selkeitä hyödyntämisalueita on robotiikassa esimerkiksi itseohjautuvien autojen objektien tunnistuksessa, sekä muilla alueilla muun muassa kasvojen ja tekstin tunnistuksessa.

2.4 Yhteenveto

Tekoäly on monia eri alueita yhteen kokoava termi, joka toimii yhtenä syynä sille, miksi jokaisella on omanlaisensa käsitys tekoälystä ja miksi yleisesti hyväksyttyä määritelmää ei ole. Toisaalta tämä johtuu myös älykkyyden vaikeasta määriteltävyydestä. Keskeistä tekoälylle on kuitenkin autonomisuus, toiminnan rationaalisuus ja kyky oppia aiemmasta kokemuksesta. Tekoäly on historiansa aikana käynyt läpi monia nousu- ja laskusuhdanteita ja viimeisten vuosien aikana innostus tekoälyyn on jälleen kasvanut. Monilla tekoälyn osa-alueilla ja sovelluksissa onkin saavutettu huomattavaa kehitystä, kuten itseohjautuvissa autoissa ja koneoppimisessa. Erityisesti koneoppiminen ja sen eri tekniikat ovat viime vuosina olleet uutisotsikoissa ja alan kehittymistä on vauhdittanut muun muassa suurten datamäärien helpompi saatavuus.

3 SISÄLOGISTIIKAN AUTOMAATIOJÄRJESTELMÄT

Tämän luvun tarkoituksena on tarjota lukijalle yleinen käsitys sisälogistiikan automaatiojärjestelmistä. Aihetta lähestytään selvittämällä ensin, mitä yleisesti sisälogistiikalla ja automaatiojärjestelmillä tarkoitetaan. Tämän jälkeen tutustutaan erityyppisiin sisälogistiikan automaatiojärjestelmiin.

3.1 Sisälogistiikka

Ennen sisälogistiikan tarkempaa käsittelyä on syytä avata ensin toimitusketjun hallinnan ja logistiikan hallinnan käsitteet, joiden tunteminen auttaa asettamaan sisälogistiikan laajempaan kontekstiinsa organisaatioiden toiminnassa. Toimitusketjun hallinnalla tarkoitetaan kaikkien niiden toimintojen koordinointia ja integrointia sujuvaksi prosessiksi, joita liittyy tuotteen toimittamiseen raaka-aineista alkaen lopulliselle asiakkaalle (Lummus & Vokurka, 1999). Keskeistä on toimitusketjun toimittajien ja asiakkaiden suhteiden hallinta, jolla tavoitellaan parhaita arvon luontia asiakkaalle mahdollisimman alhaisilla kustannuksilla koko toimitusketjulle (Christopher, 2011).

Logistiikka on yksi osa toimitusketjuun liittyvistä prosesseista ja sillä tarkoitetaan materiaalien, osien ja valmiiden lopputuotteiden hankinnan, kuljetuksen ja varastoinnin strategista hallinnointia organisaation ja sen markkinointikanavien läpi (Christopher, 2011). Logistiikan hallintaan sisältyy yllä mainittujen asioiden tehokas suunnittelu, toteutus ja hallinta (Lambert & Cooper, 2000). Logistiikan hallinta on siis suppeampi käsite kuin toimitusketjun hallinta. Toimitusketjun hallinta käsittää paljon enemmän tehtäviä ja kattaa laajuudeltaan useita eri organisaatioita, kun taas logistiikka keskittyy yhden organisaation materiaalivirtojen hallintaan (Lambert & Cooper, 2000).

Sisälogistiikka viittaa eri organisaatioiden, kuten teollisuusyri-tysten, tukku- ja vähittäiskauppojen, sekä valtion instituutioiden *sisätilojen* materiaalikäsitteilyyn liittyviin teknisiin järjestelmiin, palveluihin ja siihen liittyvään muuhun liiketoimintaan (Nagel ym., 2008). Logistiikan ja sisälogistiikan ratkaisevin

ero on se, että sisälogistiikan järjestelmät ovat täysin hallittavissa olevia suljettuja järjestelmiä, joilla on selvä rajapinta ulkopuolella oleviin järjestelmiin (Nagel ym., 2008).

Modernin sisälogistiikan historia alkaa 1950-luvulta, jolloin esimerkiksi ensimmäiset trukit tulivat markkinoille ja standardoitiin kuormalavojen koko (Kartnig ym., 2012). 1960-luvulla varastointiteknologiat kehittyivät nopeasti ja alalla siirryttiin työvoimaintensiivisistä toimintatavoista pääomaintensiivisiin toimintatapoihin. Tuolloin kehitettiin muun muassa ensimmäiset automatisoidut korkeavarastot. 1970-luvulla korostuu mikroprosessorien kehitys, joiden avulla automaatio valtasi alaa yhä enenevässä määrin. 1980-luvulla tietokoneiden kehityksen ohella erityisesti viivakoodien tuominen sisälogistiikkaan auttoi tehostamaan toimintaa, koska tuotteet voidaan viivakoodin avulla tunnistaa automaattisesti. 1990-luvulta lähtien tietoverkot mahdollistivat tietokoneiden ja oheislaitteiden tiiviimmän integroimisen ja 2000-luvulla langattomat verkot tehostivat kommunikointia ja laitteiden, kuten itseohjautuvien AGV-vaunujen ohjaamista. Lisäksi RFID-tagien kehitys on vaikuttanut merkittävästi sisälogistiikkaan, koska niihin voidaan sisällyttää paljon enemmän tietoa kuin viivakodeihin. (Kartnig ym., 2012).

3.2 Automaatiojärjestelmät

Automaatio pyrkii korvaamaan ihmisen suorittamat manuaaliset tehtävät automaattisilla laitteilla ja tietokoneilla (Bainbridge, 1983). Automaatio viittaa järjestelmään tai laitteeseen, joka osittain tai kokonaan suorittaa toimintoja, jotka olivat ennen tai olisivat olleet ihmisen suorittamia (Parasuraman ym., 2000). Automaation voidaan ajatella olevan seuraava kehitysaskel mekanisaatiosta. Mekanisaation kehittyminen mahdollisti aikanaan fyysisen työn korvaamisen laitteilla, mutta vaati aina ihmisen toimimaan laitteen operoijana. Automaatio mahdollistaa, että mentaalinen työ voidaan myös korvata laitteilla, eikä ihmistä välttämättä tarvita automatisoidussa tehtävässä lainkaan. (Hitomi, 1994).

Vaikka automaation avulla voitaisiin korjausta ja huoltoa lukuun ottamatta tehdä ihmisen läsnäolo tarpeettomaksi, kaikki automaatiojärjestelmät eivät kuitenkaan tee jokaista toimintoa itse, vaan jättävät tarkoituksella joitakin päätöksiä ihmisen tehtäväksi. Automaation tasoja onkin akateemisessa kirjallisuudessa määritelty useita riippuen siitä, kuinka paljon ihminen osallistuu järjestelmän päätöksentekoon (Parasuraman ym., 2000). Yksinkertaisimmillaan järjestelmä voi tarjota eri ratkaisu- tai päätösvaihtoehdot ja jättää ihmiselle vastuun päätöksenteosta. Toisena ääripäänä on järjestelmä, joka päättää ja suorittaa kaiken itsenäisesti, sivuuttaen ihmisen kokonaan. Automaatio ei siis ole yksinkertaisesti ”kaikki tai ei mitään”-tyyppinen ratkaisu. (Parasuraman ym., 2000).

3.3 Erilaisia sisälogistiikan automaatiojärjestelmiä ja niiden ominaisuuksia

Nykyään sisälogistiikan automaatiojärjestelmiä on toteutettu moniin eri tarkoituksiin. Karkeasti jaettuna sisälogistiikan automaatiojärjestelmiä on kahteen tarkoitukseen: tavaroiden siirtämiseen ja tavaroiden välivarastointiin. Keskeisenä osana sisälogistiikan automaatiojärjestelmiä ovat myös tilausten keräily, ohjelmistot, sekä oheislaitteet ja -koneet. Sisälogistiikan automaatiojärjestelmä ei välttämättä kata esimerkiksi koko jakelukeskuksen kaikkia toimintoja, vaan järjestelmä voi olla ”automaatiosaarekkeena” muutoin manuaalisessa ympäristössä.

3.3.1 Tavaroiden siirtäminen

Esimerkkinä tavaroiden siirtämiseen keskittyvästä automaatiojärjestelmästä ovat lentokenttien matkatavaroiden siirtämiseen tarkoitetut järjestelmät, jotka huolehtivat muun muassa siitä, että matkatavarat päätyvät matkatavaran luovutuksesta oikeaan lentokoneeseen. Tavaroiden siirtämiseen keskittyviä sisälogistiikan automaatiojärjestelmiä voidaan toteuttaa useita eri teknologioita ja laitteita hyödyntäen. AGV-vihivaunut (engl. automatic guided vehicle) ovat itsenäisesti liikkuvia robotteja, jotka kulkevat vapaasti tietyllä alueella (kuvio 3).



KUVIO 3: Kaksi AGV-vaunua, joista toiseen on lastattu kuormaa (Muratec USA, 2019)

AGV-vaunut voivat esimerkiksi noutaa tuotteita varaston hyllyltä ja kuljettaa ne eteenpäin käsiteltäviksi. AGV-vaunut voivat käyttötarkoituksesta riippuen olla pieniä, kapasiteetiltaan kymmeniä tai satoja kiloja, tai hyvinkin suuria raskaaseen teollisuuteen tarkoitettuja, jotka pystyvät siirtämään kymmenien tonnien kuorman (Savant Automation, 2019). RGV-vaunut (engl. rail guided vehicle) ovat samankaltaisia kuin AGV-vaunut, mutta vapaan liikkumisen sijaan ne kulkevat kiskolla. EMS-järjestelmissä (engl. electric monorail system) tavara

kuljetetaan myös kiskolla, mutta lattian sijaan kiskot ovat asennettu ylhäälle. Lisäksi tavaroiden liikuttamiseen voidaan käyttää esimerkiksi rulla- tai hihnakuljettimia.

3.3.2 Tavaroiden välivarastointi

Välivarastointiin tarkoitettuja sisälogistiikan automaatiojärjestelmiä ovat esimerkiksi portaalirobotisolut ja korkeavarastot. Portaalirobotisolussa toimii yksi tai useampi portaaliroboti, joka varastoi sisään tulevan tavaran ja keräilee ulos lähetettävät tilaukset (kuvio 4).



KUVIO 4: Kaksi portaalirobotisolua, joissa kussakin on kaksi portaalirobotia (Cimcorp 2019b)

Portaaliroboti (engl. gantry robot) liikkuu puomien päällä varastoalueen yläpuolella x-akselin suuntaisesti. Robotin tarttuja liikkuu y-akselin suuntaisesti ja kykenee siten poimimaan tavaran miltä tahansa varastopaikalta. Portaalirobotisoluissa tavaroita käsitellään päällekkäin kasattuina pinoina, esimerkiksi muovilaatikkopinoina, ja portaaliroboti kykenee ottamaan yhden tai useamman laatikon kerrallaan. Korkeavarastot (engl. high bay warehouse) koostuvat jopa useiden kymmenien metrien korkuisista hyllyistä. Hyllyjen välissä käytävillä operoi hyllystöhisseejä (engl. stacker crane), jotka siirtävät tavaroita varastoon ja varastosta pois.

3.3.3 Tilausten keräily

Tilauksen keräily on asiakkaan pyynnön mukaisten tiettyjen tuotteiden noutamista varastosta (De Koster ym., 2007) ja se on osa useimpien sisälogistiikan automaatiojärjestelmien toimintoja. Keräiltävät yksiköt riippuvat automaatiojärjestelmästä. Portaalirobotit voivat esimerkiksi keräillä muovilaatikoita ja korkeavarastot kokonaisia kuormalavallisia tuotteita. Korkeavarastot voivat keräillä myös pienempiä yksiköitä, kuten pahvi- tai muovilaatikoita, jolloin puhutaan miniloat-teknologiaa hyödyntävistä korkeavarastoista (Jurczak, 2018).

Ihminen voi myös toimia osana keräilyä, mutta manuaalista keräilyä hyödyntävät järjestelmät eivät luonnollisesti ole aina täysin automaattisia. Puhekeräilyssä ihminen saa kuulokkeiden kautta puheena tiedon seuraavaksi keräiltävästä tuotteesta (Battini ym., 2015). Kun tuote on keräilty, se vahvistetaan keräi-

lyä ohjaavalle järjestelmälle, joka sen jälkeen antaa tiedon seuraavasta tuotteesta. Pick-to-light-järjestelmässä varaston hyllyihin on asennettu valoja, jotka osoittavat keräilijälle varastopaikan, josta tuotteita pitää seuraavaksi keräillä. (Battini ym., 2015). Sekä pick-to-light-järjestelmässä että puhekeräilyssä järjestelmä siis automaattisesti suunnittelee parhaimman reitin tilauksen keräilylle, mutta itse keräily jätetään ihmisen suoritettavaksi.

3.3.4 Tietojärjestelmät ja ohjelmistot

Varastoissa olevia sisälogistiikan automaatiojärjestelmiä ohjaa WCS-järjestelmä (engl. warehouse control system). WCS on reaaliaikainen järjestelmä, joka kontrolloi laitetasolla tavaroiden kulkua eri automaatiolaitteilla (Envista, 2019). WCS kommunikoi ylemmällä tasolla olevan WMS-järjestelmän kanssa (engl. warehouse management system). WMS on tietojärjestelmä, jonka avulla voidaan hallita varaston materiaalivirtaa tavaroiden vastaanotosta aina niiden lähteykseen (Faber ym., 2002). WMS puolestaan kommunikoi organisaation ERP-järjestelmän kanssa. Erona WCS-järjestelmään on siis se, että WMS on hallinnollinen tietojärjestelmä, kun taas WCS on tekninen järjestelmä (Faber ym., 2002). Joissakin tapauksissa WMS voi hallinnoida myös useampien varastojen tai jakelukeskusten toimintaa (Envista, 2019).

3.4 Yhteenveto

Sisälogistiikka on osa organisaatioiden toimitusketjun hallintaa, jonka tehokkuutta parantamalla on mahdollista saavuttaa kilpailuetu muihin organisaatioihin nähden. 1950-luvulta alkaen erilaiset teknologiset innovaatiot ovat mahdollistaneet sisälogistiikan tehokkuuden huomattavan parantamisen. Teknologian kehittyminen on näkynyt myös automaatiojärjestelmien kehittämisessä ja nykyään on jo mahdollista automatisoida lähes kaikki sisälogistiikan tehtävät, jolloin ihmistä tarvittaisiin enää vain toiminnan valvomiseen, poikkeustilanteiden käsittelyyn ja laitteiden huoltoon. Sisälogistiikan automaatiojärjestelmien voidaan ajatella olevan tavaroiden siirtämiseen tai välivarastointiin tarkoitettuja järjestelmiä, mutta usein ne suorittavat kumpaakin tehtävää. Tilausten keräily on myös olennainen osa näitä automaatiojärjestelmiä. Automaatiojärjestelmiä voidaan toteuttaa useilla eri tekniikoilla ja lähestymistavoilla, riippuen tilanteesta ja käyttötarkoituksesta. Erityyppiset automaatiojärjestelmät ja niiden toteutustekniikat eivät kuitenkaan ole toisiaan poissulkevia, sillä esimerkiksi samassa järjestelmässä saatetaan käyttää sekä portaalirobotteja että korkeavarastoja. Lisäksi automaatiojärjestelmä saattaa olla vain yksi osa muutoin manuaalisessa toimintaympäristössä. Esimerkiksi AGV-vaunut voivat kuljettaa tilauksen tavarat ihmisen luo, joka pakkaa ne ja toimittaa eteenpäin vietäväksi.

4 TEKOÄLYN HYÖDYNTÄMINEN SISÄLOGISTIIKAN AUTOMAATIOJÄRJESTELMISSÄ

Tässä luvussa pyritään tunnistamaan tekoälyn hyödyntämiskohteita edellisessä luvussa esiteltyihin sisälogistiikan automaatiojärjestelmiin. Luvussa käydään läpi aiheeseen liittyvä aikaisempi tutkimus, jonka perusteella johdetaan tekoälyn sovellusalueet sisälogistiikan automaatiojärjestelmissä.

4.1 Kirjallisuuskatsaus

Tekoälyn käyttöä sisälogistiikan automaatiojärjestelmissä selvitettiin kirjallisuuskatsauksen avulla, joka suoritettiin maaliskuussa 2019 käyttäen Google Scholar -hakukonetta. Hakusanoina käytettiin alan englanninkielisiä termejä (intralogistics, artificial/computational intelligence, automated warehouse, warehouse order picking, automation system) ja niiden yhdistelmiä. Tämän jälkeen lähdemateriaalia pyrittiin löytämään käyttämällä hakusanoina myös aihealueen tarkempia ja laajempia, sekä niitä sivuavia termejä (logistics, industry 4.0, machine learning, internet of things). Kustakin hakutermistä katsottiin ensin otsikkotasolla noin sata ensimmäistä hakutulosta, jonka jälkeen relevanteimpiin artikkeleihin tutustuttiin tarkemmin.

Kirjallisuuskatsauksen avulla selvisi, että suurin kiinnostus aiemmassa tutkimuksessa on kohdistunut varastoissa liikkuviin itseohjautuviin vaunuihin, kuten AGV-vaunuihin. Muutamassa tutkimuksessa käytettiin lähestymistapana moniagenttijärjestelmiä, joita sovellettiin esimerkiksi AGV-vaunujen keräilyyn. Näiden lisäksi joissakin tutkimuksissa esille nousivat geneettiset algoritmit, joita on sovellettu esimerkiksi AGV-vaunuihin ja tuotteiden sijoitteluun varastossa. Tosin geneettiset algoritmit voidaan luokittelutavasta riippuen katsoa kuuluvan tekoälyn sijasta myös kybernetiikan alueelle. Kaiken kaikkiaan tekoälyä ja sisälogistiikkaa käsitteleviä tutkimuksia oli löydettävissä vähän. Seuraavissa luvuissa käydään tarkemmin läpi kirjallisuuskatsauksessa esiin nousseita teemoja.

4.1.1 Tilauksen keräilyongelma

Tilauksen keräily varastosta, eli asiakkaan pyynnön mukaisten tiettyjen tuotteiden noutaminen varastosta, (engl. warehouse order picking) on pitkään tutkittu aihe logistiikan alueella (De Koster ym., 2007). Tilauksen keräily voidaan esittää tunnettuna kauppamatkustajan ongelmana (Ratliff & Rosenthal, 1983), jossa kauppamatkustajan tavoitteena on löytää lyhin tai halvin reitti, joka käy kaikissa ennalta määrätyissä kaupungeissa. Sisälogistiikkaan sovellettuna keräilijän tavoitteena on löytää nopein reitti, joka kulkee kaikkien tilauksen tuotteiden varastopaikkojen kautta. Muun muassa Ratliff ja Rosenthal (1983) ovat jo 1980-luvun alkupuolella esittäneet algoritmin keräilyongelmaan, jonka avulla ongelma voidaan ratkaista. Ratliffin ja Rosenthalin algoritmi perustuu parhaimman reitin löytämiseen eli reititykseen, mutta tilauksen keräilyyn voidaan vaikuttaa myös muilla tavoin. De Koster ym. (2007) ovat tehneet tilausten keräilystä kattavan kirjallisuuskatsauksen, jonka perusteella keräilyn tehokkuuteen vaikuttavat:

- varaston pohjapiirros, eli layout (hyllyjen paikka, käytävien määrä, jne.)
- tuotteiden sijoittelu (mihin tuotteet laitetaan varastossa)
- varaston aluejako (yksi keräilijä poimii tuotteita vain tietyltä alueelta)
- tilausten yhdistely (usean eri tilauksen tuotteita haetaan yhdellä kertaa)
- reititys

Ratliffin ja Rosenthalin algoritmi, sekä De Kosterin kirjallisuuskatsauksessa läpi käydyt tutkimukset eivät kuitenkaan käsittele tilausten keräilyä tekoälyn näkökulmasta. Tutkimuksia, jossa ongelman ratkaisussa löytyisi tekoälylle ominaisia autonomisuuden ja oppimisen piirteitä on huomattavasti vaikeampi löytää. Yksi aikaisimmista tutkimuksista on Seidmannin (1988) tekemä, jossa hän käyttää tekoälyksi katsottavia algoritmeja parantamaan korkeavaraston toimintaa. Hänen kehittämänsä algoritmit kykenevät mukautumaan ja järjestelmä oppii esimerkiksi sijoittamaan pienen vaihtuvuuden tuotteet varaston perälle (Seidmann, 1988).

4.1.2 Moniagenttijärjestelmät

Moniagenttijärjestelmät ovat yksi tekoälyn tutkimusalueista, jonka tavoitteena on pyrkiä tutkimaan keinoja toteuttaa ja koordinoita useista itsenäisistä agenteista koostuvia järjestelmiä (Stone & Veloso, 2000). Esimerkkinä moniagenttijärjestelmiä sisälogistiikassa hyödyntävästä tutkimuksesta on Güllerin ja Hegmannsin (2014) tutkimus, jossa moniagenttijärjestelmän avulla pystytään simuloimaan automaattisen keräilyjärjestelmän toimintaa ja jonka avulla voidaan suunnitella tehokkaampia järjestelmiä. Beyer ym. (2015) käyttävät myös moniagenttijärjestelmään perustuvaa lähestymistapaa sisälogistiikan järjestelmien suunnitteluun.

Shen ym. (2006) tekemän kirjallisuuskatsauksen mukaan agenttipohjaiset järjestelmät sopisivat muun muassa AGV-järjestelmiin. Tämänlaisia järjestelmiä on myös onnistuneesti toteutettu. AI Magazine käsittelee artikkelissaan entisen Kiva Systemsin, nykyisen Amazon Roboticsin AGV-varastojärjestelmää, jossa tekoälyä käytetään multiagenttijärjestelmän lisäksi AGV-vaunujen reittien suunnittelussa ja resurssien allokoinnissa (Wurman, D'Andrea & Mountz, 2008). AGV-vaunut ovat edustettuina myös muissa tutkimuksissa. Mansouri ym. (2015) esittävät tekoälyä hyödyntävän keinon hallinnoida joukkoa autonomisia robotteja, kuten AGV-vaunuja, jotka pystyvät muun muassa mukautumaan uusiin ympäristöihin.

4.1.3 Geneettiset algoritmit

Geneettiset algoritmit saavat inspiraationsa biologisesta evoluutiosta, jossa vahvimmat yksilöt pärjäävät heikompia paremmin. Geneettiset algoritmit luovat ratkaisuvaihtoehtoja ongelmaan evoluution tavoin yhdistelemällä eri ratkaisujen osia, arvioimalla niiden suorituskykyä ja pitämällä vahvimmat vaihtoehdot ”hengissä”. (Whitley, 1994). Dou, Chen ja Yang (2015) soveltavat geneettisiä algoritmeja ja koneoppimisen piiriin kuuluvaa vahvistusoppimista AGV-varastojärjestelmän tehokkuuden lisäämisessä. Tutkimuksessa geneettisiä algoritmeja käytetään tehtävien allokointiin ja vahvistusoppimista reititykseen ja törmäyksen estoon. Kłosowski, Gola ja Amila (2018) soveltavat myös geneettisiä algoritmeja heidän simulaatiossaan, joka parantaa AGV-varastojärjestelmän tehokkuutta. Tämän lisäksi he hyödyntävät myös sumeaa logiikkaa. He osoittavat ratkaisunsa soveltuvan erityisesti suuriin varastoihin, joissa tavanomaiset algoritmit eivät heidän simulaatiossaan kyenneet toimimaan tehokkaasti. (Kłosowski ym., 2018). Muita tutkimuksia, jotka käyttävät geneettisiä algoritmeja keräilyn optimointiin ovat muun muassa Enen ja Öztürkin (2011), jossa niitä sovelletaan autoteollisuuden varastoihin, sekä Kuon ym. (2016), jossa hyödynnetään myös parvioptimointia. Lisäksi geneettisten algoritmien avulla on pyritty optimoimaan varastojen automaatiojärjestelmien korkeita energiankulutuspiikkejä (Cárdenas ym., 2009). Geneettiset algoritmit valittiin niiden sopivuutensa perusteella myös tämän tutkimuksen simulaation teoreettiseksi pohjaksi. Tarkemmin geneettisten algoritmien teoriaa käsitellään luvussa 6.2.1.

4.1.4 Industry 4.0 ja Logistics 4.0

Industry 4.0 on yksi teollisuuden ja logistiikan muotisansanoista, joka pääpiirteissään tarkoittaa teollisuustoiminnan digitalisointia, desentralisointia, tiedon välityksen lisäämistä eri entiteettien välillä, sekä siirtymistä kohti autonomista ohjausta ja organisaatiota (Delfmann ym., 2018). Keskeisiä teknologioita Industry 4.0 -siirtymässä ovat muun muassa esineiden ja palvelujen internet (IoT&S) ja tekoäly. Logistics 4.0 on käytännössä Industry 4.0 logistiikan kontekstissa, jonka tavoitteena on digitalisoida toimitusketju ja sen hallinta (Bukova ym., 2018). Logistics 4.0 tehtävänä on lisätä automaatiota ja tekoälyä niihin soveltuville alueille, löytää sopiva tasapaino ihmisten ja laitteiden yhteistoiminnassa, sekä muut-

taa toimintatapoja uusien teknologioiden myötä tulevien tarpeiden mukaisiksi. Uusiin toimintatapoihin liittyvät esimerkiksi tarve hallinnoida ja analysoida kasvavia datamääriä, sekä mukautua entistä nopeammin muuttuvampaan ympäristöön. (Bukova, 2018). Muun muassa Wang (2016) on esittänyt konseptin Logistics 4.0 -ratkaisusta, mutta se ei suoraan käsittele sisälogistiikan automaatiojärjestelmiä. Lee ym. (2018) ovat tutkimuksessaan suunnitelleet Industry 4.0 -vision mukaisen esineiden internet -pohjaisen (IoT) WMS-järjestelmän, jossa tutkijoiden mukaan hyödynnetään koneoppimista informaation prosessointiin ja päätöksentekoon.

4.1.5 Muut esille nousseet teemat ja tutkimukset

Yllä mainittujen suurempien teemojen lisäksi kirjallisuuskatsauksessa löytyi yksittäisiä tutkimuksia, jotka joko suoraan tai epäsuoraan liittyvät tämän tutkielman aihealueeseen. Drakaki ja Tzionas (2017) käyttävät varaston keräilyn aika-tilalutukseen vahvistusoppimista ja osoittavat sen simulaatiossaan toimivan tehokkaasti. Brezovnik ym. (2014) puolestaan hyödyntävät parviälyä optimoimaan varastolayoutin kokoa, jotta tilaa saataisiin säästettyä. Weber ja Schütte (2019) tarkastelevat koneoppimisen vaikutusta vähittäiskaupan alalla ja ovat tutkimuksessaan tunnistaneet eri käyttökohteita koneoppimiselle. Yhtenä koneoppimisen hyödyntämisen alueena mainitaan tavaroiden kuljetus, johon sisälogistiikka lukeutuu. Tällä alueella koneoppimisen tekniikoista voisi tutkimuksen mukaan hyödyntää optimointia ja ennustamista. (Weber & Schütte, 2019). Tarkempia keinoja tutkimus ei kuitenkaan tarjoa vaan antaa ainoastaan laaja-alaisen yleiskuvan koneoppimisen hyödyntämismahdollisuuksista vähittäiskaupan alalla.

4.2 Yhteenveto

Alla olevaan taulukkoon (taulukko 1) on luokiteltu kirjallisuuskatsauksessa löydettyjä tutkimuksia sen mukaan, miten ne käsittelevät tekoälyä. Taulukkoon on listattu vain ne tutkimukset, jotka suoraan käsittelevät sisälogistiikan automaatiojärjestelmiä ja jotka siten parhaiten vastaavat tämän tutkielman aihetta. Yllä mainituista tutkimuksista taulukkoon ei ole otettu esimerkiksi yleisesti varaston keräilyongelmaa käsitteleviä tutkimuksia, eikä suurinta osaa Industry 4.0- tai Logistics 4.0 -aiheita käsitteleviä tutkimuksia, koska ne eivät joko käsittele aihetta tekoälyn kannalta tai eivät esitä suoraa tekoälyn käyttökohteita sisälogistiikan automaatiojärjestelmille. Geneettisiä algoritmeja hyödyntävät tutkimukset ovat mukana listauksessa, koska ne voidaan luokitella kuuluvan myös koneoppimisen alueeseen kybernetiikan lisäksi.

Löydettyissä tutkimuksissa tekoälyn teknologioista käytettiin pääasiassa moniagenttijärjestelmiä, parviälyä ja -optimointia, geneettisiä algoritmeja, sumeaa logiikkaa, sekä koneoppimista vahvistusoppimisen muodossa. Näitä teknologioita sovellettiin muun muassa automaatiojärjestelmien suunnitteluun, AGV-vaunujen ohjaukseen, sekä varaston keräilyongelman ratkomiseen. Tutki-

mukset kohdistuivat joko AGV-vaunuihin tai korkeavarastoihin. Yhtään tutkimusta, jossa tekoälyteknologioita olisi sovellettu portaalirobottijärjestelmiin ei löytynyt. Mutta koska kaiken tyyppisissä sisälogistiikan automaatiojärjestelmissä varaston keräilyongelma on yhteinen, voidaan tekoälyn avulla olettaa voivan parantavan myös portaalirobottijärjestelmien toimintaa. Suurin ero portaalirobottisoluissa on se, että keräily tapahtuu ylhäältä. Varastoalueella ei näin ollen tarvita käytäviä, koska robotit liikkuvat varastoitavien tavaroiden yläpuolella.

TAULUKKO 1 Tekoälyä sisälogistiikan automaatiojärjestelmissä hyödyntäviä tutkimuksia

Tutkimus	Konteksti tai ongelma	Tekoälyn hyödyntäminen
Beyer ym., 2015	Materiaalinhallintajärjestelmien suunnittelu	Moniagenttijärjestelmät
Brezovnik ym., 2015	Automaatiovaraston suunnittelu	Parviäly
Cárdenas, 2009	Virrankulutuspiikkien optimointi automaatiovarastossa	Geneettiset algoritmit
Dou ym., 2015	AGV-vaunut, suunnittelu ja aikataulutus	Geneettiset algoritmit, vahvistusoppiminen
Drakaki & Tzionas, 2017	Teollisuusjärjestelmien aikataulutus, keräilyongelma	Vahvistusoppiminen
Ene & Öztürk, 2012	Keräilyongelma (tavaroiden sijoittelu)	Geneettiset algoritmit
Güller & Hegmanns, 2014	Keräilyongelma	Moniagenttijärjestelmät
Kłosowski ym., 2018	AGV-vaunut	Geneettiset algoritmit, sumea logiikka
Kuo ym., 2016	Keräilyongelma (tavaroiden sijoittelu)	Geneettiset algoritmit, parvioptimointi
Lee ym., 2018	WMS, IoT, Industry 4.0 -konseptin ideointi	Sumea logiikka
Mansouri ym., 2015	AGV-vaunut	Erilaisia päättelyyn kykeneviä algoritmeja
Seidmann, 1988	Keräilyongelma	Adaptiivisuuteen kykeneviä algoritmeja

5 TUTKIMUSMENETELMÄ

Tutkimusmenetelmänä käytetään Design Science Research Methodology (DSRM) -prosessia, jota käyttäen luodaan geneettisiä algoritmeja hyödyntävä simulaatio, jolla osoitetaan tekoälyn soveltuvuus portaalirobottipohjaisiin sisälogistiikan automaatiojärjestelmiin. Tämän luvun tarkoituksena on antaa yleiskuva DSRM-prosessista ja perustella sen käyttö tässä tutkimuksessa.

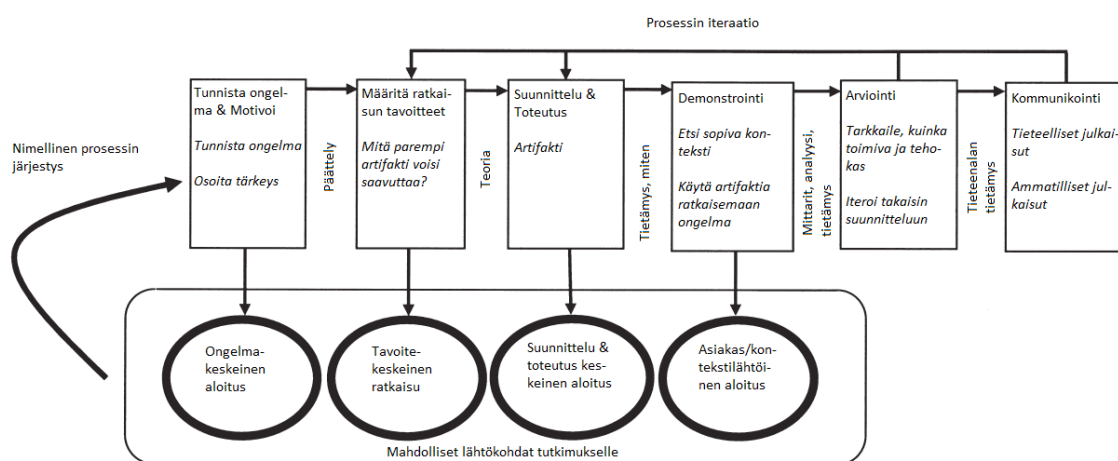
5.1 Design Science ja DSRM-prosessi tietojärjestelmätieteessä

Design Science (DS) on suunnittelutieteellinen tutkimusmenetelmä, jossa luodaan ja arvioidaan IT-artefakteja ratkaisuuina organisaatioissa tunnistettuihin ongelmiin. IT-artefaktit voivat ongelmasta riippuen olla esimerkiksi algoritmeja, toimintatapoja tai prototyyppijärjestelmiä. (Hevner ym., 2004). DS on suhteellisen uusi ja vähän käytetty menetelmä tietojärjestelmätieteen tutkimuksessa verrattuna perinteisiin behaviorismiin pohjautuviin tutkimusparadigmoihin, kuten teorioiden luomiseen ja niiden testaukseen (Peffers ym., 2007). Hevner ym. (2004) ovat kuvailleet behaviorismiin pohjautuvien menetelmien ja DS:n eroa analogialla, jonka mukaan behaviorismiin pohjautuvat menetelmät pyrkivät etsimään ”mikä on totta”, kun taas DS pyrkii luomaan ”mikä on tehokasta”.

Hevnerin ym. (2004) määrittelemien, tehokkaan DS-tutkimuksen mahdollistavien suuntaviivojen mukaisesti artefaktin tulee olla tarkoituksenmukainen ja tarjota teknologiaan pohjautuva ratkaisu merkitykselliseen ja relevanttiin ongelmaan. Artefaktin luomisessa ja arvioinnissa tulee noudattaa hyvin määritellyjä ja täsmällisiä tapoja, sekä tarvittaessa iteratiivisesti ja inkrementaalisesti kehittää artefaktia, jotta sille asetetut tavoitteet saadaan saavutettua ja mahdolliset rajoitteet huomioitua. Keskeistä DS:ssä on myös tuottaa tieteellistä kontribuutiota artefaktiin liittyviin aihealueisiin. Lisäksi DS-tutkimuksen kommunikoinnissa tulisi ottaa huomioon niin teknologiaorientoitunut yleisö, kuin johtoporaaseenkin kuuluva yleisö. (Hevner ym., 2004).

Design Science Research Methodology (DSRM) -prosessi toimii yleisesti hyväksyttynä viitekehyksenä DS-tutkimukselle tietojärjestelmätieteessä (Peffers

ym., 2007). DSRM-prosessissa DS-tutkimus on jaettu kuuteen aktiviteettiin (kuvio 5), jotka Peffers ym. (2007) ovat tunnistaneeet aiemmista vaikutusvaltaisista tietojärjestelmätieteen DS-tutkimuksista. DSRM-prosessin suoritusjärjestyksen ei pidä aina lähteä liikkeelle prosessin ensimmäisestä aktiviteetista, vaan tutkimus voi tilanteesta riippuen käynnistyä myös prosessin toisesta, kolmannesta tai neljännestä aktiviteetista (Peffers ym., 2007). Tässä tutkielmassa lähtökohtana toimii prosessin toisesta aktiviteetista alkava ”tavoitekeskeinen ratkaisu”, koska tutkimuksen lähtökohtana on toimeksiantajan tarve parantaa sisälogistiikan automaatiojärjestelmien tehokkuutta tekoälyllä, mikä voidaan ratkaista artefaktilla.



KUVIO 5: DSRM-prosessi (Peffers ym., 2007, mukailten)

5.2 DSRM-prosessi sovellettuna tähän tutkimukseen

Koska lähtökohtana tälle tutkimukselle on toimeksiantajayrityksen käytännön ongelma, Design Science soveltuu tutkimusmenetelmäksi erityisen hyvin, sillä DS:n pääajatuksena on organisaatioiden ongelmien ratkaiseminen (Hevner ym., 2004). IT-artefaktin, eli tässä tapauksessa tekoälyä geneettisten algoritmien muodossa hyödyntävän simulaation avulla voidaan osoittaa tekoälyn soveltuvuus sisälogistiikan automaatiojärjestelmien ohjaukseen. DS tarjoaa IT-artefaktin muodossa ratkaisuehdotuksen toimeksiantajalle ja laajentaa akateemista tietämystä tekoälyn käyttömahdollisuuksista suhteellisen harvaan tutkituissa sisälogistiikan automaatiojärjestelmissä. Seuraavissa alaluvuissa tarkastellaan kukin DSRM-prosessin aktiviteetti tämän tutkielman aiheeseen sovellettuna.

5.2.1 Ongelman tunnistaminen ja motivointi

DSRM-prosessin ensimmäisenä aktiviteettina on ongelman tunnistaminen ja motivointi, jonka avulla pystytään hahmottamaan ongelma ja arvo, jonka sen ratkaisemalla voidaan saavuttaa (Peffers ym., 2007). Tätä on käsitelty johdannossa, eli luvussa 1. Ongelmana on, että toimeksiantajan nykyiset robottien algoritmit eivät pysty mukautumaan muuttuvaan tilausrakenteeseen, koska ke-

räilyalgoritmit ovat staattisia ja toimivat siten optimaalisesti vain tietyssä tilanteessa. Tekoälyä hyödyntämällä voitaisiin aiemmista tilauksista oppia, mihin tuotteet kannattaa varastossa sijoitella, jotta tehokkuus säilyisi, vaikka tilausra-
kenne muuttuisikin.

5.2.2 Ratkaisun tavoitteet

Toisena aktiviteettina on ratkaisun tavoitteiden määrittäminen. Tavoitteet voivat olla niin määrällisiä kuin laadullisiakin (Peffer ym., 2007). Tässä tutkielmassa tavoitteena on geneettisten algoritmien avulla ratkaista tuotteiden sijoitteluun liittyvä ongelma, eli mihin paikkaan varastossa tuotteet kannattaa sijoitella, jotta niiden keräily olisi mahdollisimman tehokasta. Tutkimuksen tuloksena syntyvä ohjelmistoartefakti tulee simuloimaan portaalirobottivaraston toimintaa, jota tekoälyä soveltamalla pyritään parantamaan. Tavoitteena on siis osoittaa, että tekoälyä voi geneettisten algoritmien muodossa hyödyntää portaaliroboteista koostuvissa sisälogistiikan automaatiojärjestelmissä.

5.2.3 Suunnittelu ja toteutus

Kolmantena aktiviteettina on artefaktin haluttujen toiminnallisuuksien ja arkkitehtuurin suunnittelu, sekä toteutus (Peffer ym., 2007). Yhdeksi ratkaisuteknologiaksi valitaan geneettiset algoritmit, koska luvun 4 kirjallisuuskatsauksen perusteella selvisi, että geneettisiä algoritmeja on viime vuosien aikana sovellettu joihinkin sisälogistiikan automaatiojärjestelmien osa-alueisiin (mm. Cárdenas, 2009; Dou ym., 2015; Ene & Öztürk, 2012; Klosowski ym., 2018; Kuo ym., 2016). Näin ollen geneettisten algoritmien voidaan olettaa soveltuvan myös portaalirobottipohjaisiin automaatiojärjestelmiin, koska problematiikka on pohjimmiltaan samankaltainen kaiken tyyppisissä sisälogistiikan automaatiojärjestelmissä. Artefaktin tavoitteita, sekä suunnittelua ja toteutusta käsitellään tarkemmin luvussa 6.

5.2.4 Demonstrointi

Prosessin neljäntenä aktiviteettina olevan demonstroinnin tarkoituksena on osoittaa artefaktin ratkaisevan käsiteltävänä olleen ongelman (Peffer ym., 2007). Koska varaston toimintaa simuloiva artefakti on tietokoneohjelma, demonstrointina toimii simuloinnin ajaminen, joka antaa tuloksena tiedon siitä, mihin tuotteet kannattavat varastossa sijoitella.

5.2.5 Arviointi

Arviointiaktiviteetin tarkoituksena on havainnoida ja mitata, kuinka hyvän ratkaisun artefakti tarjoaa käsiteltävään ongelmaan (Peffer ym., 2007). Toimivien artefaktin tehokkuutta arvioivien mittareiden luominen on tärkeää DS-tutkimuksessa (Hevner ym., 2004). Artefaktin arvioinnissa käytetään historiatietoa vanhoista tilauksista. Arvioinnissa mittarina käytetään robotin kulkemaa yhteismatkaa, joka robotilta kuluu tietyn tilausjoukon tilausten kaikkien tuotteiden

den keräilyyn. Artefaktin demonstroitua ja arviointia käsitellään tarkemmin luvussa 7.

5.2.6 Kommunikointi

Viimeisenä aktiviteettina on levittää tietoa kommunikoimalla kaikki relevantit edellisissä aktiviteeteissa käsitellyt asiat, kuten ongelma, artefakti, sen suunnittelu, ja tulokset asianmukaiselle yleisölle. Tämän tutkimuksen tulokset kommunikoidaan tuloksena syntyvässä pro gradu -tutkielmassa.

5.3 Aineisto

Tutkielmassa käytetty aineisto, eli historiallinen tilausdata saadaan toimeksiantajalta. Tilausdata on kerätty tuotantokäytössä olevalta elintarvikkeita käsittelevältä automaatiojärjestelmältä. Kyseinen automaatiojärjestelmä koostuu neljästä portaalirobotisolusta, joissa käsitellään yhteensä yli 300 eri tuotetta. Yksittäisessä solussa ei kuitenkaan ole varastoituna kaikkia tuotteita, vaan ne ovat ryhmiteltyinä eri robotisoluihin vaaditun säilytyslämpötilan mukaan. Data on peräisin 16 päivän ajanjaksolta vuodelta 2019, ajoittuen tammi- ja helmikuun vaihteeseen. Tutkielmaa varten aineistosta suodatetaan vain yhdelle robotisolulle osoitetut tilaukset. Suodattamisen jälkeen aineistoon jäi käytettäväksi 9 815 tilausta. Tilaus koostuu tilausriveistä, jossa määritetään, mitä tuotetta tilataan ja kuinka paljon. Keskiarvoltaan yhdessä tilauksessa oli 4,9 tilausriviä, minimissään 1 ja maksimissaan 19 tilausriviä. Tilauksissa käsiteltiin yhteensä 124 eri tuotetta. Luodun artefaktin demonstroinnissa ja validoinnissa käytettiin osajoukkoa tästä aineistosta.

6 ARTEFAKTIN SUUNNITTELU JA TOTEUTUS

Luvun tarkoituksena on kuvata artefaktin suunnitteluun ja toteutukseen liittyviä päätöksiä ja toimintatapoja. Luvussa esitellään myös yleinen teoria geneettisistä algoritmeista, johon artefaktin toteutus perustuu.

6.1 Artefaktin tavoitteet

Tarkoituksena on rakentaa geneettisiä algoritmeja hyödyntävä sovellus, joka simuloi yksinkertaistetun portaalirobottisolun toimintaa. Sovelluksen toiminta rajoittuu sijoitteluongelman ratkaisemiseen, eli sovellus pyrkii löytämään optimaalisimman tavan sijoitella tuotteet varastoon, jotta niiden keräilemiseen menisi mahdollisimman vähän aikaa. Muut tavalliseen portaalirobottisoluun kuuluvat toiminnallisuudet ovat rajattu simulaatiosta pois.

Simulaatio mittaa matkan, joka robotilta kuluu kerätä tietyn tilausjoukon tuotteet kullakin algoritmin tuottamalla tuotteiden sijoittelujärjestyksellä varastossa. Tavoitteena on geneettisten algoritmien avulla tuottaa sellainen varaston tuotteiden sijoittelujärjestys, josta tilausjoukon tuotteet voitaisiin kerätä tavannoista nopeammin.

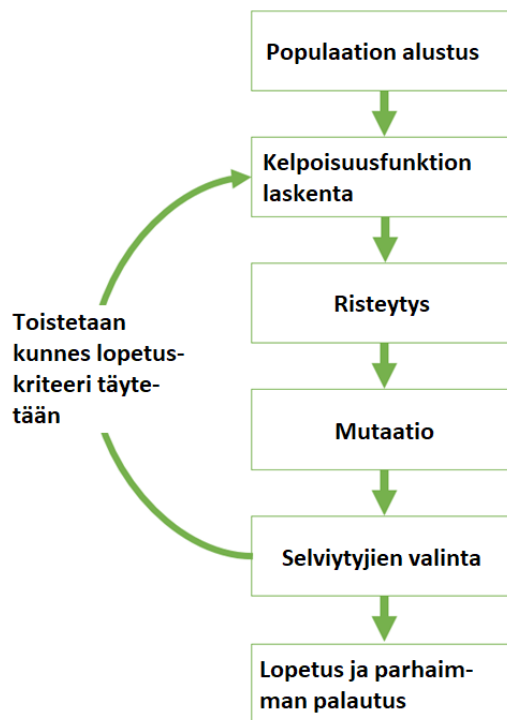
6.2 Geneettiset algoritmit

Geneettiset algoritmit saavat inspiraationsa evoluutiosta (Whitley, 1994) ja pääajatuksena on niin sanottu "survival of the fittest", eli tilanne, jossa parhaimmat yksilöt selviytyvät todennäköisemmin kuin heikommat yksilöt. Kuten luonnossakin, vahvimpien yksilöiden jälkikasvu on molempien vanhempien perimän risteytyksen ja mahdollisten mutaatioiden tulos. Geneettisten algoritmien evoluutio perustuu siis satunnaisiin muutoksiin, jotka saattavat tuottaa edellisten sukupolvien yksilöitä paremmin suoriutuvia jälkeläisiä. Geneettisten algorit-

mien pioneerina pidetään John Hollandia, jonka tutkimuksien myötä 1960- ja 1970-luvuilla ala on saanut alkunsa (Konak ym., 2006).

6.2.1 Teoria

Kuviossa 6 on esitetty geneettisten algoritmien perusrakenne, joka säilyy samana riippumatta kohdealueesta, johon algoritmia sovelletaan. Kohdealueeriippuvaisia ovat käytännössä ainoastaan ongelman esitysmuoto algoritmissa ja kelpoisuusfunktion toteutus (Whitley, 1994).



KUVIO 6: Geneettisten algoritmien perusrakenne (Tutorialspoint, 2019, mukailten)

Geneettisten algoritmien suoritus alkaa aina populaation alustuksella. Populaatio on osajoukko ongelman kaikista mahdollisista ratkaisuvaihtoehdoista (Tutorialspoint, 2019). Populaation yhtä ratkaisuvaihtoehtoa kutsutaan kromosomiksi. Kromosomit koostuvat geeneistä ja jokainen kromosomin geeni kontrolloi yhtä tai useampaa kromosomin ominaisuutta. (Konak ym., 2006). Arvoa, joka kussakin geenissä on, kutsutaan alleeliksi (Tutorialspoint, 2019). Populaatio alustetaan tyypillisesti joukolla satunnaisia kromosomeja (Whitley, 1994).

Populaation alustuksen jälkeen kromosomien kelpoisuus lasketaan kelpoisuusfunktion avulla ja kromosomit, jotka eivät täytä ongelma-alueen rajoitteita tai vaatimuksia, karsitaan pois (Haupt, 1995). Kelpoisuusfunktio määrittää ongelmakohteisesti ja sen tavoitteena on tuottaa tulos, jonka avulla eri kromosomien paremmuutta voidaan vertailla.

Kromosomien paremmuuden määrittämisen ja soveltumattomien karsinnan jälkeen valitaan vanhemmat, jotka tuottavat seuraavan sukupolven jälkeläiset. Jälkeläisiä tuotetaan niin paljon, että karsittujen kromosomien määrä saa-

daan korvattua. (Haupt, 1995). Vaihtoehtoisesti kaikki aiemman sukupolven kromosomit voidaan korvata uusilla, jolloin koko populaatio uusiutuu joka sukupolvella. Kummassakin tapauksessa populaation koko säilyy aina vakiona. On olemassa useita tapoja valita vanhemmat kromosomien joukosta, mutta yhteistä eri tavoille on se, että mitä paremmaksi kromosomi arvioidaan kelpoisuusfunktiossa, sitä suuremmalla todennäköisyydellä se valitaan vanhemmaksi. (Tutorialspoint, 2019).

Jälkikasvun muodostamisessa käytetään risteytys- ja mutaatio-operaatioita. Eri risteytystapoja on useita ja sopivin valitaan käsiteltävän ongelman mukaan. Tavoitteena on muodostaa jälkikasvu, joka on molempien vanhempien yhdistelmä. (Tutorialspoint, 2019). Risteytystavasta riippumatta, tietyn ajan kulluttua risteytys alkaa tyypillisesti tuottaa samoja ratkaisuvaihtoehtoja, jotka eivät välttämättä ole kovinkaan hyviä ratkaisuja ongelmaan. Tällöin algoritmi on löytänyt paikallisen optimin, mutta ei kykene löytämään parhainta globaalia optima. (Konak ym., 2006). Geneettisissä algoritmeissa mutaatiolla on tärkeä rooli populaation varianssin kasvattajana. Mutaatio tapahtuu yleensä yksittäisen geenin tasolla, jolloin pienellä todennäköisyydellä geenin arvo muuttuu satunnaisesti joksikin muuksi. Näin jälkikasvussa esiintyy myös, ominaisuuksia joita kummallakaan vanhemmista ei ole, ja näin todennäköisyys löytää globaali optimi kasvaa. (Konak ym., 2006).

Kun uusi populaatio on risteytysten ja mutaatioiden kautta luotu, prosessia toistetaan uudelleen niin kauan, kun jokin lopetusehto saavutetaan. Lopetusehtoina voivat toimia esimerkiksi ennalta määrätyn kelpoisuusfunktion tuloksen saavuttaminen. Lopetus voidaan myös määrittellä tapahtuvan silloin, kun sukupolvia on luotu tietty määrä tai tilanteessa, jossa ratkaisujen kelpoisuus ei enää kasva. (Tutorialspoint, 2019).

Jotta ongelma voidaan ratkaista geneettisiä algoritmeja käyttäen, pitää ongelma koodata reaali maailman versiosta laskennalliseen muotoon. Ongelman koodaus on kelpoisuusfunktion lisäksi toinen ongelmariippuvainen asia geneettisissä algoritmeissa (Whitley, 1994). Populaatiota, joka on kuvattu laskennallisessa muodossa, kutsutaan genotyyppiiksi ja populaation ratkaisuja, jotka ovat kuvattu kuten ne reaali maailmassa esiintyisivät, kutsutaan fenotyyppiiksi. Genotyypin kuvauksessa käytetään yleisimmin binääristä, kokonaisluku- tai reaali lukuesitystapaa, sekä permutaatioesitystapaa. (Tutorialspoint, 2019). Näissä esitystavoissa ongelma esitetään listana, joiden alkiot ovat geenejä ja alkioiden arvot alleeleja. Binäärisessä esitystavassa arvot ovat joko 0 tai 1, ja kokonaisluku- sekä reaali lukuesitystavassa vastaavasti kokonais- tai reaali lukuja. Permutaatioesitystavassa ongelman ratkaisu ilmaistaan arvojen keskinäisenä järjestyksenä listassa ja esitystapa sopii esimerkiksi kauppamatkustajan ongelman ratkaisemiseen (Tutorialspoint, 2019).

6.2.2 Geneettiset algoritmit sovellettuna portaalirobottivarastoon

Tuotteiden sijoittelu portaalirobottivarastoon on ongelmana samankaltainen kuin kauppamatkustajan ongelma, joten permutaatioesitystapa sopii genotyypin kuvaukseksi. Erona kauppamatkustajan ongelmaan on kuitenkin se, että varastossa samaa tuotetta voi olla useassa paikassa ja samaa tuotetta voidaan ti-

lata useaan kertaan. Kauppamatkustajan ongelmassa kaikki kaupungit ovat erilaisia ja niissä käydään vain kerran.

Tutkielman aihealueeseen sovellettuna kromosomi kuvaa tuotepinojen järjestystä varastossa. Yksi tuotepino koostuu rajatusta määrästä yhtä tuotetta. Jokaisessa kromosomissa on samat ennalta määritellyt tuotepinot sijoiteltuna. Kromosomien paremmuuden määrittämisessä käytetään samaa tilausdatan joukkoa. Jokaiselle kromosomille lasketaan yhteenlaskettu matka, joka robotilta kuluu ensin kuljettua jokainen tuotepino paikalleen varastossa ja sen jälkeen kerätä jokaisen tilausjoukon tilauksien kaikki tuotteet. Mitä pienempi on yhteenlaskettu matka, sitä parempi kromosomi on.

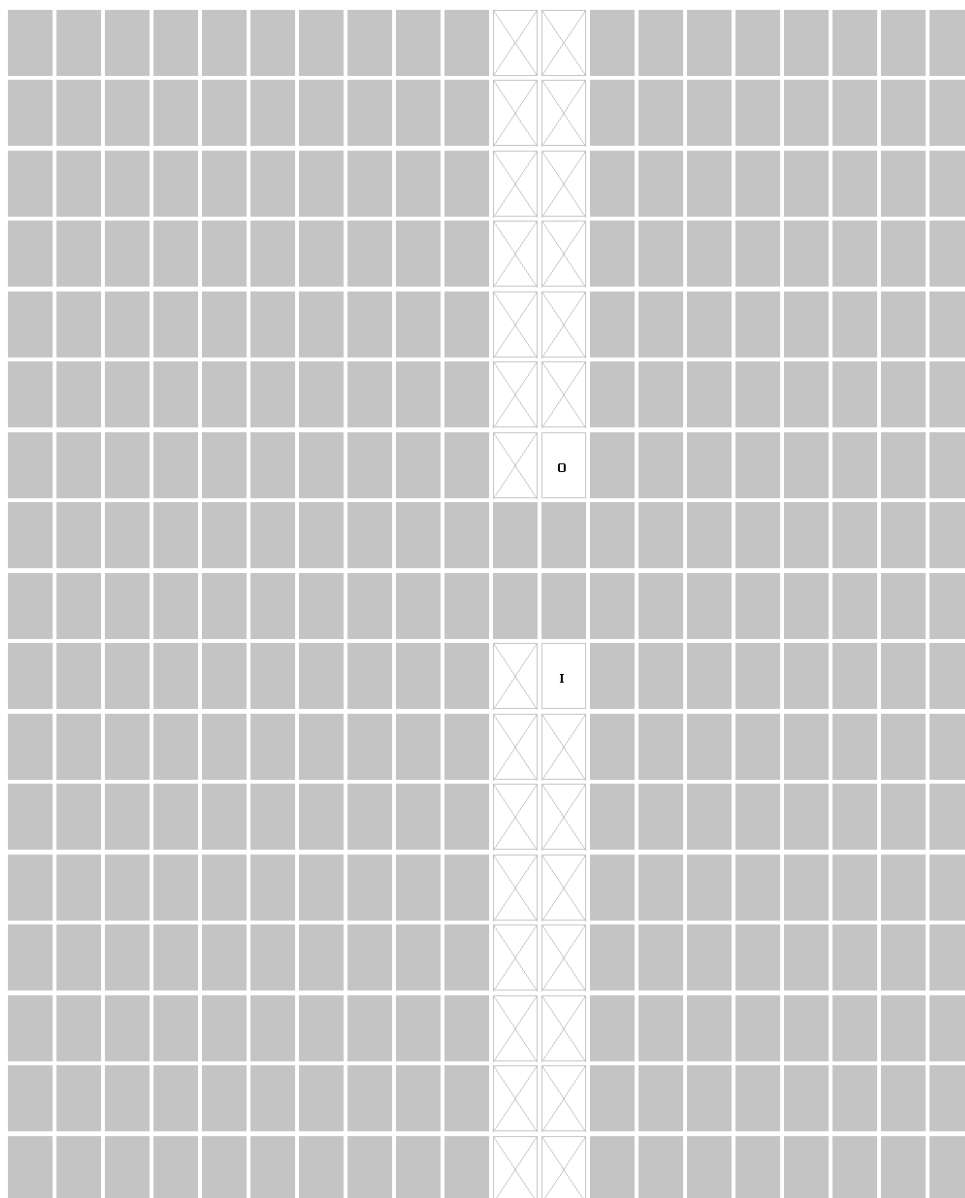
Tilausjoukon perusteella määritellään myös kromosomeissa käytetyt tuotepinot. Algoritmin alussa lasketaan, kuinka monta erilaista tuotetta tilauksissa on ja kuinka monta kertaa kutakin tuotetta tilataan. Näin saadaan jokaisen tuotteen osalta selville vähimmäismäärä, joka täytyy olla sijoiteltuna varastoon, jotta jokainen tilausjoukon tilaus saadaan keräiltyä. Jokaisessa kromosomissa sijoitellaan vain vähimmäismäärä tuotteita, eli varasto olisi täysin tyhjä sen jälkeen kun tilausjoukon jokainen tilaus olisi keräilty.

6.3 Tietorakenne varastokartan kuvaamiseen

Koska portaalirobotivarastot ovat nelikulmion muotoisia ja varastopaikat ovat pääsääntöisesti saman kokoisia, voitaisiin kromosomin genotyypin tietorakenteena käyttää kaksiulotteista taulukkoa, jossa genotyypin taulukon solu vastaisi samassa paikassa olevaa varastopaikkaa portaalirobotivarastossa. Tällöin genotyyppi vastaisi myös fenotyyppiä, eli ongelman kuvausta reaali maailmassa. Tämä ei kuitenkaan vastaisi todellisuutta, sillä portaalirobotivarastojen sisällä on myös kuljettimia, joita pitkin tuotepinoja kuljetetaan sisään ja ulos, ja jotka siis estävät tiettyjen varastopaikkojen käytön (ks. kuvio 7, jossa käytössä olevat varastopaikat ovat merkitty harmaalla ja kuljettimien päälle osuvat estetyt varastopaikat valkoisella).

Estettyjen varastopaikkojen huomioimiseksi genotyypin ja fenotyypin kuvauksen olisi edelleen voinut pitää samana estämällä estettyjä varastopaikkoja vastaavien genotyypin taulukon solujen käytön, mutta lopulta genotyypin ja fenotyypin kuvauksen päätettiin olevan erilainen, koska algoritmin muiden osien, erityisesti risteytyksen ja mutaatioiden, toteutus osoittautui näin yksinkertaisemmaksi. Genotyyppinä toimii yksiulotteinen lista, jossa jokainen listan alkioiden paikoista vastaa tiettyä varastopaikkaa. Listassa on vain käytössä olevat varastopaikat. Kromosomilla on myös tieto varastokartan, eli layoutin todellisesta rakenteesta, jotta varastopaikkojen väliset etäisyydet saadaan oikein laskettua.

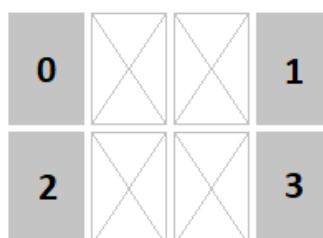
Kuvio 8 havainnollistaa yksinkertaistetusti fenotyypin ja genotyypin eroa. Kuvion esimerkkivarastossa on kahdeksan varastopaikkaa, joista neljä ovat estettyjä. Estetyt paikat ovat merkitty rastilla. Genotyypin listan koko on neljä, koska esimerkkivaraston kahdeksasta paikasta vain neljä on käytettävissä. Listan alkioiden paikkojen vastaavuus on merkitty numeroin fenotyypin kuvassa.



KUVIO 7: Varastokartta, jota algoritmissa käytetään

Genotyyppi: [0, 1, 2, 3]

Fenotyyppi:



KUVIO 8: Yksinkertaistetun genotyypin ja fenotyypin eroavaisuus ja vastaavuus

Sisällyttämällä kromosomissa tieto myös varastopaikkojen todellisesta sijainnista fenotyypissä, voidaan esimerkiksi laskea, että kuvion paikkojen 0 ja 1 välinen etäisyys on todellisuudessa suurempi kuin esimerkiksi paikkojen 0 ja 2, vaikka genotyypin listassa paikat 0 ja 2 ovatkin kauempana toisistaan. Samaa logiikkaa käyttäen algoritmi mallintaa kuvion 7 varastokartan genotyypin.

Varastokartta kuviossa 7 on se, jota algoritmissa käytettiin tulosten muodostamiseen. Varasto on samanlainen kuin se varasto, josta aineistona saatu tilausdata on peräisin, mutta leveydeltään puolet pienempi. Puolet pienempää varastoa käytettiin, koska algoritmi simuloi tilausten keräilyä yhdellä robotilla tavanomaisen kahden robotin sijaan. Varaston 340 paikasta 30 ovat estettyjä ja joiden kohdalla kuljettimet sijaitsevat. Kaksi estetyistä varastopaikoista, jotka ovat merkitty kuviossa kirjaimilla I ja O merkitsevät sisääntulo- ja poisvientipaikkojen sijainnin. Algoritmissa kaikki tuotepinot tulevat sisääntulopaikasta (I) varastoon, ennen kuin tilaukset keräillään. Jokaisen tilauksen tuotteet vie-dään poisvientipaikalle (O) ennen kuin seuraavaa tilausta aletaan keräillä.

6.4 Kelpoisuusfunktio

Kelpoisuusfunktion tavoitteena on simuloida portaalirobottivaraston tilausten keräilyä ja laskea yhteenlaskettu matka, jonka robotti kulkee sijoitellakseen aluksi kaikki tuotepinot ja kerätäkseen sen jälkeen kaikki tilausdatan tilaukset. Simulointi on hyvin yksinkertaistettu versio tuotantokäytössä olevista robottisoluohjainjärjestelmistä, ja jonka toiminta on siten vain suuntaa antava oikean järjestelmän käyttäytymisestä.

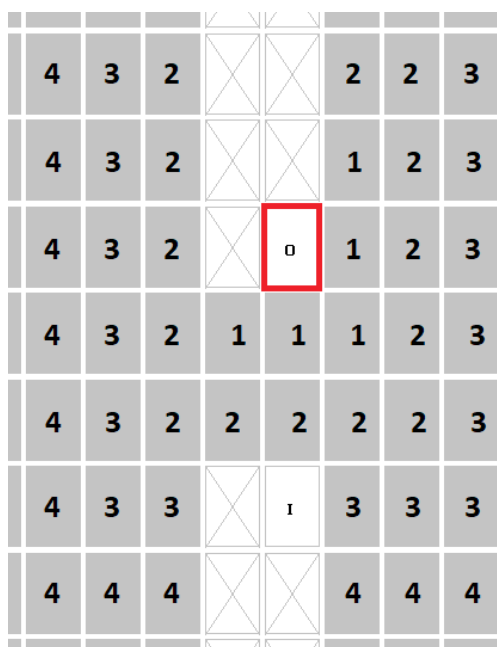
6.4.1 Tilaukset kelpoisuusfunktion syötteenä

Yksi tilaus koostuu yhdestä tai useammasta tilausrivistä, jossa määritetään, mitä tuotetta keräillään ja kuinka monta laatikkoa. Samaa tuotetta voidaan yhdessä tilauksessa tilata usealla tilausrivillä. Tilauksen tuotteilla on tietty keräilyjärjestys, jota on noudatettava, jotta tuotteet saadaan pinottua kuormalavoille asiakkaan vaatimusten mukaisesti. Kelpoisuusfunktio simuloi tietyn saman tilausjoukon keräilyä jokaiselle kromosomille ja laskee yhteen robotilta kuluneen matkan. Kun yhden tilauksen tuotteet ovat keräilty, kuljetetaan ne poisvientipaikalle (O). Robotti kykenee kantamaan kerralla rajatun määrän laatikoita, joten poisvientipaikalle voidaan joutua menemään myös tyhjentämään robotin tarttuja, jos se on kerätty täyteen.

6.4.2 Tilausten keräily

Simulaatiossa tilauksien tuotteet keräillään siis ennalta määrättyssä järjestyksessä. Kelpoisuusfunktioon toteutetussa keräilyalgoritmissa etsitään aina robotin nykyistä sijaintia lähimpänä oleva tuotepino, johon tilausjärjestyksessä seuraavana olevaa tuotetta on sijoitettu. Tuotepinon etsimisen jälkeen lasketaan etäisyys tuotepinoon, lisätään se robotin kulkemaan yhteismatkaan, ja merkitään

robotin uusi paikka kerätyn tuotepinon varastopaikan kohdalle. Robotin kulke-
ma matka kahden varastopaikan välillä lasketaan kaavalla $\max(\Delta x, \Delta y)$, missä Δx on kahden varastopaikan etäisyys x-akselilla ja Δy vastaavasti etäisyys y-akselilla. Etäisyydeksi määritellään x- ja y-akselin suurin arvo euklidisen etäisyyden sijasta, koska näin saadaan arvioitua todenmukaisemmin myös aikaa, jota robotilta kuluisi siirtyä kahden varastopaikan välillä. Portaalirobotissa x- ja y-akseli liikkuvat itsenäisesti toisiinsa nähden, joten ajallisesti siirtyminen kahden varastopaikan välille riippuu siitä akselista, joka joutuu liikkumaan pisimmän matkan, eikä niinkään absoluuttisesta matkasta näiden varastopaikkojen välillä. Kuviossa 9 on esimerkkinä laskettu etäisyydet poisvientipaikasta sen lähellä oleviin varastopaikkoihin. Simulaation yksinkertaistamiseksi kahden vierekkäisen varastopaikkojen välisen etäisyyden on määritelty olevan 1, sekä x- ja y-akselin liikkumisnopeuden on määritelty olevan sama.



KUVIO 9: Etäisyydet varastopaikkoihin punaisella korostetusta poisvientipaikasta

6.4.3 Pisteiden laskenta

Kelpoisuusfunktion laskema pisteytys, eli robotin kulkema yhteismatka, kullekin kromosomille koostuu kahden kokonaisuuden summasta:

1. Matka, joka robotilta kuluu sijoitella kaikki tuotepinot varastopaikoille.
2. Matka, joka robotilta kuluu jokaisen tilausjoukon tilauksen osalta keräillä tilauksen tuotteet ja kuljettaa ne poisvientipaikkaan, sekä tarvittaessa tyhjentää täyteen tullut tarttuja poisvientipaikalla.

6.5 Algoritmin toiminta ja parametrit

Algoritmi noudattaa toiminnoissaan luvun 6.2.1 mukaista geneettisten algoritmien yleistä teoriaa. Populaation alustuksen jälkeen kullekin kromosomille lasketaan kelpoisuusfunktion avulla pisteytys, joka määrittää kromosomin kelpoisuuden. Tämän jälkeen luodaan uusi joukko kromosomeja, joka korvaa kokonaisuudessaan aiemman sukupolven populaation. Poikkeuksena on kuitenkin aiemman sukupolven parhain kromosomi, joka automaattisesti siirretään seuraavaan sukupolveen muuttumattomana. Tätä periaatetta kutsutaan elitismiksi (Tutorialspoint, 2019). Algoritmin toimintaan voi vaikuttaa säätämällä seuraavia parametreja:

- **Tilausten määrä** määrittää kuinka monta eri tilausta käytetään kelpoisuusfunktion laskennassa kullekin kromosomille.
- **Populaation koko** määrittää kuinka monta kromosomia kussakin sukupolvessa on.
- **Mutaatioiden todennäköisyys** määrittää todennäköisyyden, jolla kromosomin genotyypin kussakin geenissä tapahtuu satunnainen muutos. Mutaatioista kerrotaan tarkemmin luvussa 6.5.3.
- **Pinon koko** määrittää kuinka monta laatikkoa yhdessä tuotepinossa voi enintään olla. Aina kun tuotepinosta poimitaan laatikoita, pinon laatikkomäärää vähennetään poimitulla määrällä.
- **Tarttujan kapasiteetti** määrittää kuinka monta laatikkoa robotti kykenee samalla kertaa kantamaan. Kun robotti on poiminut enimmäismäärän laatikoita, on sen tyhjennettävä tarttuja menemällä poisvientipaikkaan (O) ennen kuin lisää laatikoita voidaan poimia.

6.5.1 Algoritmin alustus

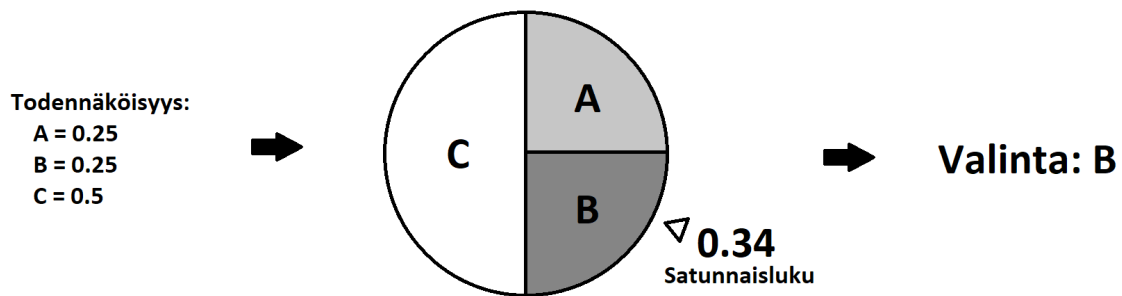
Ennen varsinaista tuotesijoittelun optimointia suorittavaa evoluutiosilmukkaa, algoritmi suorittaa populaation alustuksen ja tuotepinojen määrittelyn tilausdatan perusteella. Tuotepinot, jonka jokaisen kromosomin on sijoitettava, saadaan laskemalla ensin tilausdatan jokaisen tuotteen osalta kokonaistilausmäärä, eli kuinka monta laatikkoa samaa tuotetta kaikissa tilauksissa yhteensä keräillään. Tuotepinojen määrä kullekin tuotteelle saadaan jakamalla tuotteen kokonaistilausmäärä *pinon koko* -parametrin arvolla. Esimerkiksi jos tuotetta A tilataan yhteensä 54 laatikkoa ja *pinon koko* on 10 laatikkoa, luodaan tuotteelle A yhteensä kuusi tuotepinoa, joista viisi on 10 laatikon pinoja ja yksi on neljän laatikon pino. Kun tuotepinojen määrä on selvitetty, algoritmi suorittaa populaation alustuksen. Populaation alustuksessa luodaan *populaation koko* -parametrin verran kromosomeja, joissa tuotepinot ovat satunnaisesti sijoiteltu varastoon.

6.5.2 Vanhempien valinta ja jälkikasvun muodostaminen

Kelpoisuusfunktion tuloksen perusteella lasketaan jokaiselle kromosomille todennäköisyys, jolla kromosomi valitaan vanhemmaksi. Mitä pienempi kelpoi-

suusfunktion tuottama luku, eli kuljettu matka on, sitä suurempi on todennäköisyys. Todennäköisyys lasketaan jakamalla kromosomin kelpoisuusfunktion tuottaman luvun käänteisluku kaikkien kromosomien kelpoisuusfunktion tulosten käänteislukujen summalla. Näin jokaiselle kromosomille saadaan määriteltyä todennäköisyydeksi reaaliluku väliltä 0–1 ja kaikkien kromosomien todennäköisyyksien summa on 1.

Jälkikasvuna syntyvän kromosomin vanhempien valinta tapahtuu geneettisissä algoritmeissa yleisesti käytetyn *roulette wheel selection* -periaatteen mukaan (Lipowski & Lipowska, 2012). Jokaisen kromosomin voidaan ajatella saavan oman sektorinsa ympyrästä ja sektorin koko määräytyy edellisessä kappaleessa mainitun todennäköisyyden mukaan. Tämän jälkeen valitaan satunnaisesti reaaliluku väliltä 0–1 ja valinta vanhemmaksi on se, minkä kromosomin sektorille satunnaisluku osuu, kuten kuviossa 10 on havainnollistettu. Toisen vanhemman valinta tapahtuu samalla periaatteella, mutta ensimmäiseksi valittua vanhempaa ei kuitenkaan voi valita uudestaan toiseksi vanhemmaksi. Sama kromosomi voi siis olla useamman lapsikromosomin vanhempi, mutta yhden lapsikromosomin molempina vanhempina ei voi olla samaa kromosomia.



KUVIO 10: Vanhempien valinta *roulette wheel selection* -periaatteen mukaan

Kun lapsikromosomin vanhemmat ovat valittu, lapsikromosomin genotyyppi muodostetaan yhdistämällä osa vanhempien genotyypeistä käyttäen *order 1 crossover* -algoritmia, joka on yksinkertainen, permutaatioihin perustuviin geneettisiin algoritmeihin soveltuva risteytystapa (Rubicite Interactive, 2019). Tätä risteytystapaa havainnollistaa kuvio 11, jossa esitetään yksinkertaisella esimerkillä lapsikromosomin genotyypin muodostaminen vanhempien genotyypeistä. Aluksi vanhemman A genotyypistä valitaan satunnaisen pituinen osalista tuotepinoja alkaen satunnaisesti valitusta kohdasta. Osalista asetetaan samaan paikkaan lapsikromosomin genotyypissä, jossa se vanhemman A genotyypissä sijaitsee. Tämän jälkeen vanhemman B genotyypistä merkitään käytetyiksi ne tuotepinot, jotka esiintyvät jo lapsikromosomin genotyypissä. Lopuksi kaikki jäljelle jääneet tuotepinot täytetään vanhemman B genotyypistä lapsikromosomin genotyyppiin järjestyksessä vasemmalta oikealle.

Lähtötilanne

Vanhempi A: [0, 1, 2, 3, 4, 5]

Vanhempi B: [5, 2, 4, 0, 3, 1]

Lapsi: [-, -, -, -, -, -]

Kohta 1: Valitse satunnainen osalista vanhemmasta A

Vanhempi A: [0, 1, 2, 3, 4, 5] ⇨ [2, 3, 4]

Kohta 2: Aseta osalista samaan paikkaan lapsikromosomin genotyypissä

Lapsi: [-, -, 2, 3, 4, -]

Kohta 3: Merkitse lapsikromosomissa jo esiintyvät arvot käytetyiksi vanhemmasta B

Vanhempi B: [5, 2, 4, 0, 3, 1] ⇨ [5, X, X, 0, X, 1]

Kohta 4: Täytä loput lapsikromosomin paikoista järjestyksessä vanhemman B arvoilla

Vanhempi B: [5, X, X, 0, X, 1]

Lapsi: [5, 0, 2, 3, 4, 1]

KUVIO 11: Lapsikromosomin muodostaminen *order 1 crossover* -periaatteella

6.5.3 Mutaatiot

Mutaatiossa kaksi tuotepinoa vaihtavat paikkaa keskenään. Mutaatiot toteutetaan lapsikromosomiin aina risteytyksen jälkeen. Jokaisella genotyypin tuotepinolla on *mutaatioiden todennäköisyys* -parametrin määräämä mahdollisuus vaihtaa paikkaa toisen satunnaisesti valitun tuotepinon kanssa. Samassa kromosomissa voi siis tapahtua useampi kuin yksi mutaatio.

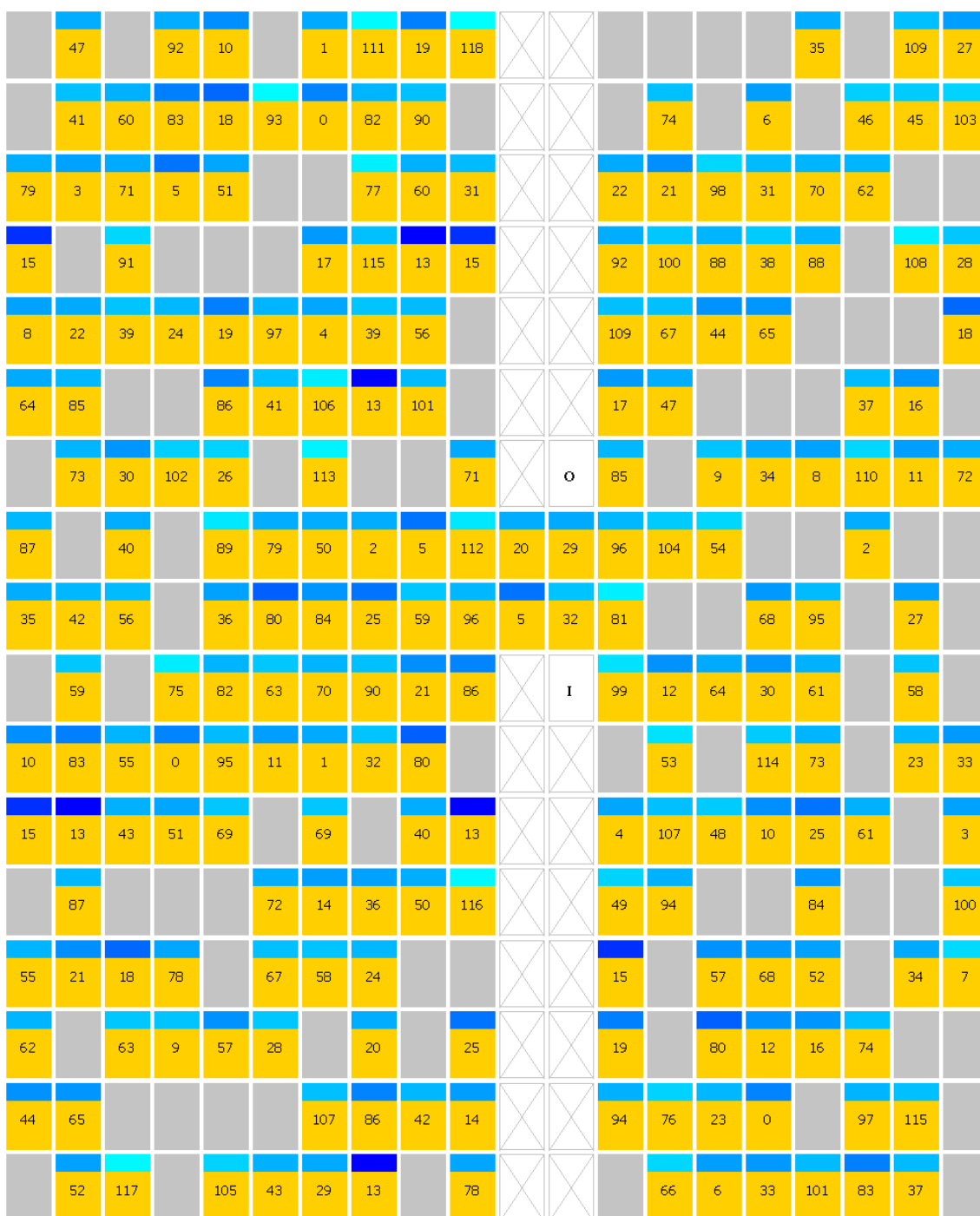
6.5.4 Toteutuksessa käytetyt teknologiat

Artefakti toteutettiin kokonaan Java-ohjelmointikielellä. Geneettisten algoritmien toteutus tehtiin ilman valmista ratkaisua ja ilman kolmannen osapuolen kirjastoja.

6.6 Algoritmin tuottama ratkaisu

Kuviossa 12 on esitetty esimerkkinä algoritmin tuottama satunnainen ratkaisu, eli ratkaisu, jossa tuotepinot ovat sijoiteltu satunnaisessa järjestyksessä varastoon. Kuvassa tuotepinon keskellä oleva numero on tuotteen yksilöivä id-numero, joka kertoo, mitä tuotetta kussakin tuotepinossa on. Tuotepinon yläosassa olevan värikoodin tehtävänä on visualisoida tuotteen kysyntää. Mitä vaa-

leaman sininen väri, sitä vähemmän kyseistä tuotetta tilataan tilausjoukon tilauksissa. Tumman sininen väri kertoo puolestaan usein tilatusta tuotteesta.



KUVIO 12: Algoritmin tuottama satunnainen ratkaisu

7 ARTEFAKTIN DEMONSTROINTI JA ARVIOINTI

Luvussa esitetään ensiksi, kuinka artefaktin tuottamat tulokset saavutettiin. Tämän jälkeen arvioidaan artefaktin kykyä ratkaista tuotteiden sijoitteluongelma simuloitussa portaalirobottivarastossa.

7.1 Demonstrointi

Artefaktin demonstroinnissa algoritmia ajettiin useita kertoja eri parametriyhdistelmillä. Varastona käytettiin kuvion 7 mukaista varastoa. Muuttuvina parametreina olivat *populaation koko* ja *mutaatioiden todennäköisyys*. Näille parametreille käytettiin seuraavia arvoja:

- *Populaation koko*: 15, 25, 50 ja 100.
- *Mutaatioiden todennäköisyys*: 0,05 %; 0,1 %; 0,15 %; 0,2 %; 0,5 % ja 1 %.

Algoritmia ajettiin kymmenen kertaa jokaisella yllä esitettyjen muuttuvien parametrien yhdistelmällä, eli yhteensä 24 eri yhdistelmällä. Parametrien arvot valittiin kokeilemalla, ja yllä oleviin arvoihin päädyttiin, koska niiden koettiin olevan mielekkäitä tuomaan esiin niiden vaikutuksen algoritmin toiminnassa.

Kaikilla ajokerroilla vakioina pysyivät *tilausten määrä*, *pinon koko* ja *tarttujan kapasiteetti*, joille määritettiin seuraavat arvot:

- *Tilausten määrä*: 500.
- *Pinon koko*: 15.
- *Tarttujan kapasiteetti*: 8.

Koko aineistosta valittiin 500 tilauksen joukko, jota algoritmissa käytettiin. Suuremmalla tilausjoukolla algoritmin ajoaika olisi kasvanut selvästi ja valittu 500 tilauksen joukko on kompromissi ajoajan ja tulosten mahdollisesti paremman yleistettävyyden välillä. Valitun 500 tilauksen joukossa yhdessä tilauksessa oli

keskimäärin 4,89 tilausriviä, vähimmillään yhdessä tilauksessa oli yksi tilausrivi ja enimmillään 18. Tilausten joukossa oli 119 eri tuotetta. Pinon koon ja tarttujan kapasiteetin arvot valittiin siten, että ne vastaisivat likimain sitä, mihin oikea järjestelmä kykenisi. Valituilla parametrien arvoilla algoritmi loi tilausjoukon pohjalta 220 tuotepinoa, jotka jokainen kromosomi sijoittelee varastoon.

Algoritmin lopetusehdoksi asetettiin 10 000 sukupolven raja, jonka saavuttuaan algoritmi palauttaa siihen mennessä parhaimman kromosomin ratkaisun. Jokaisesta ajokerrasta tallennettiin algoritmin ajoaika ja kelpoisuusfunktion tulos parhaimmalle kromosomille. Lisäksi jokaisella ajokerralla tiettyjen sukupolvien kohdalla tallennettiin sen hetkisen parhaan kromosomin kelpoisuusfunktion tulos, jotta tuloksen kehittymistä sukupolvien mittaan voitaisiin seurata.

7.2 Tulokset ja arviointi

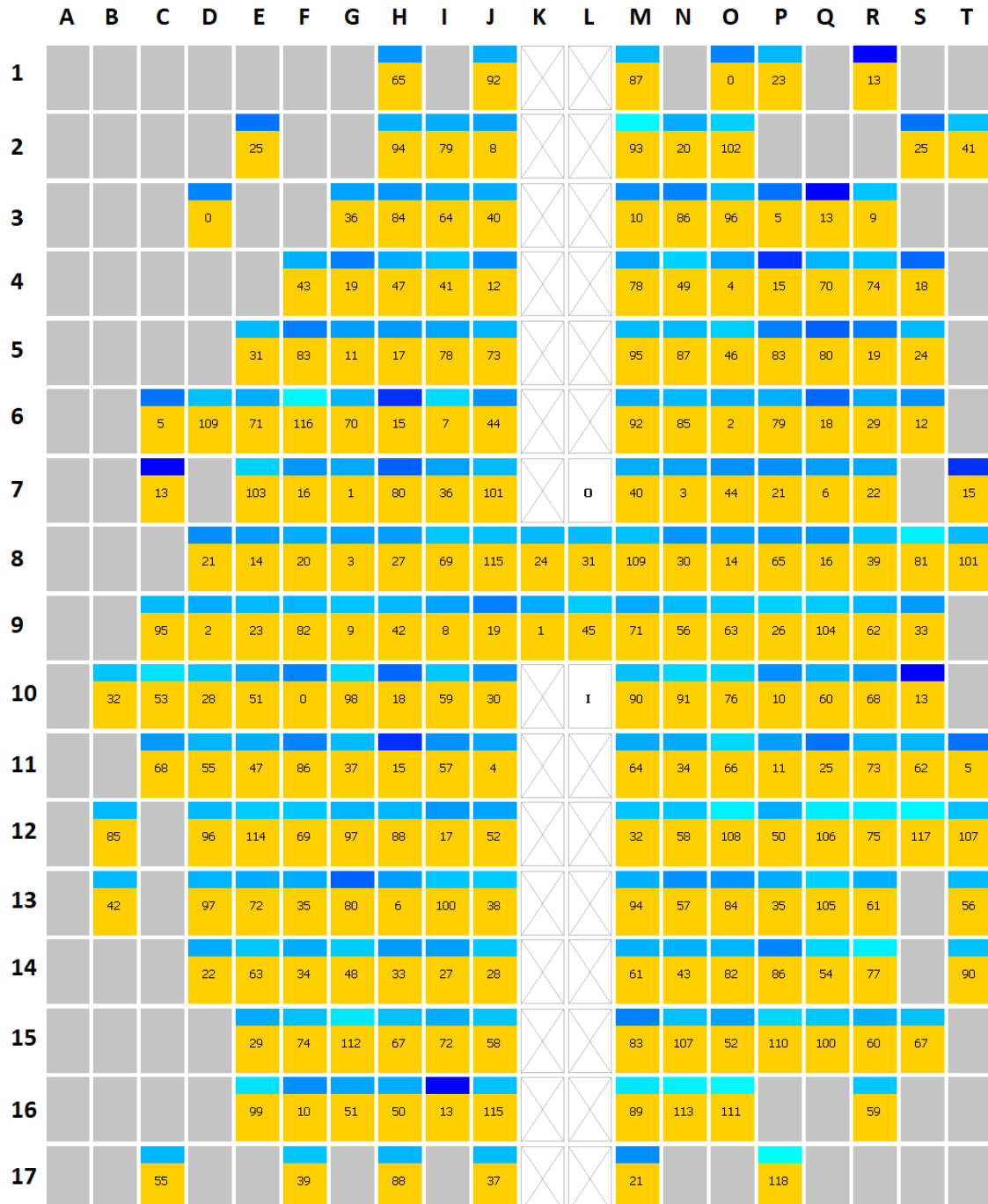
Kaikkien parametriyhdistelmien kymmenen ajokerran parhaimman kromosomin kelpoisuusfunktion keskimääräinen tulos on esitetty taulukossa 2.

TAULUKKO 2: Kelpoisuusfunktion keskimääräinen tulos eri parametriyhdistelmillä

Mutaatioiden todennäköisyys	Populaation koko 15	Populaation koko 25	Populaation koko 50	Populaation koko 100	Keskiarvo
0,05 %	15455	15048	14876	14760	15035
0,10 %	15333	14892	14835	14721	14945
0,15 %	14959	14881	14773	14813	<u>14857</u>
0,20 %	15011	14964	14777	14850	14901
0,50 %	15329	15321	15323	15072	15261
1,00 %	15971	16124	15745	15768	15902
Keskiarvo	15343	15205	15055	<u>14997</u>	

Tulosten perusteella voidaan todeta, että mitä suurempaa kromosomipopulaatiota algoritmissa käytetään, sitä parempaan tulokseen keskimäärin päädytään. Parempi tulos saavutetaan kuitenkin huomattavasti pidentyneen ajoajan kustannuksella (ks. luku 7.2.1). Mutaatioiden osalta parhaimpaan keskimääräiseen tulokseen päädytään, kun *mutaatioiden todennäköisyys* -parametrin arvo on noin 0,15 % paikkeilla. Täysin satunnaiseen ratkaisuun verrattuna algoritmin tuottama ratkaisu on huomattavasti parempi: 100 000 satunnaisen kromosomin kelpoisuusfunktion keskimääräinen tulos oli 24 616 ja algoritmin tuottaman kaikkien parametriyhdistelmien parhain tulos oli 14 005. Parannusta satunnaiseen verrattuna on siis noin 43 %.

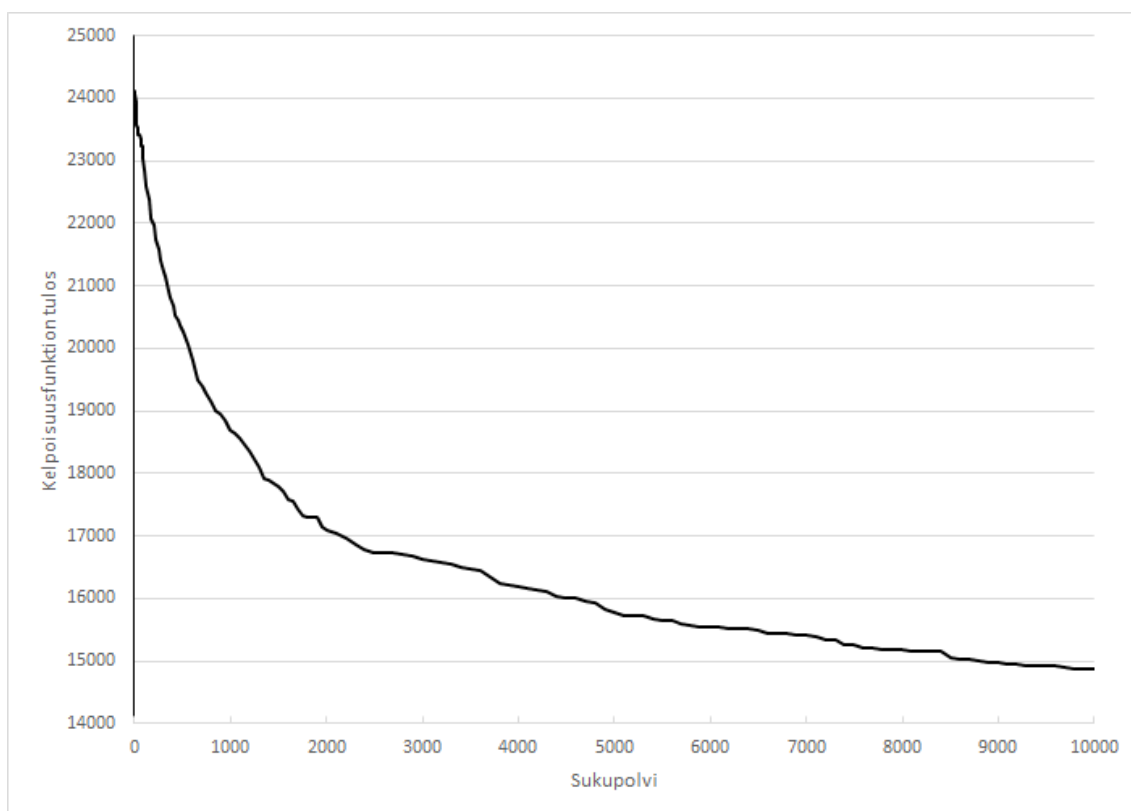
Kuviossa 13 on esimerkki algoritmin tuottamasta ratkaisusta. Selvänä erona kuvion 12 esittämään satunnaiseen ratkaisuun verrattuna on nähtävissä tuotepinojen keskittyminen sisääntulo- ja poisvientipaikkojen ympärille, ja tyhjiä varastopaikkojen jääminen varaston ulkoreunoille. Useassa tapauksessa samojen tuotteiden tuotepinot, joita tilataan usein, ovat sijoiteltuna eri puolille varastoa. Esimerkiksi tuotteen 13, jonka tuotepinot sijaitsevat varastopaikoissa C7, Q3, R1, S10 ja I16 ovat sijoiteltu siten, että tuotteeseen on lyhyt kulkumatka mistä päin varastoa tahansa.



KUVIO 13: Yksi algoritmin tuottamista ratkaisuisista

Lisäksi usein samassa tilauksessa esiintyvät tuotteet ovat useassa tapauksessa lähellä toisiaan. Esimerkiksi tuotetta 5 ja 13 tilataan yhdessä yhteensä 21 kertaa 500 tilauksen joukossa, ja tuotteen 5 kaikki kolme tuotepinoja ovat sijoitettu tuotteen 13 pinojen viereisiin paikkoihin C6, P3 ja T11. Vastaavasti tuotteen 83, joka ei kertaakaan esiinny samassa tilauksessa tuotteen 13 kanssa, tuotepinot sijaitsevat vähintään kahden varastopaikan etäisyydellä tuotteen 13 tuotepinoista paikoissa M15, P5 ja F5.

Tuotepinojen sisääntulo- ja poisvientipaikkojen ympärille keskittämisen lisäksi yhteistä kaikille parametrijohdistelmien tuottamille ratkaisuille oli myös se, kuinka ratkaisut kehittyivät ajan mittaan. Kaikkein suurimmat harppaukset parhaimman kromosomin kelpoisuusfunktion tuloksen kehittymisessä tapahtuivat suunnilleen ensimmäisen 2 500 sukupolven aikana, jonka jälkeen kehitysvauhti hiipui huomattavasti. Tästä esimerkkinä on kuvio 14, jossa esitetään kuvion 13 ratkaisun tuottaneen ajokerran parhaimman ratkaisun kehitys. Kuvion esimerkissä ensimmäisen 2 500 sukupolven aikana parhaimman ratkaisun tulos on parantunut 30,6 % ensimmäisen sukupolven parhaimpaan ratkaisuun verrattuna ja 10 000 sukupolven saavuttaessa tulos on parantunut yhteensä 38,3 % ensimmäiseen sukupolven verrattuna.

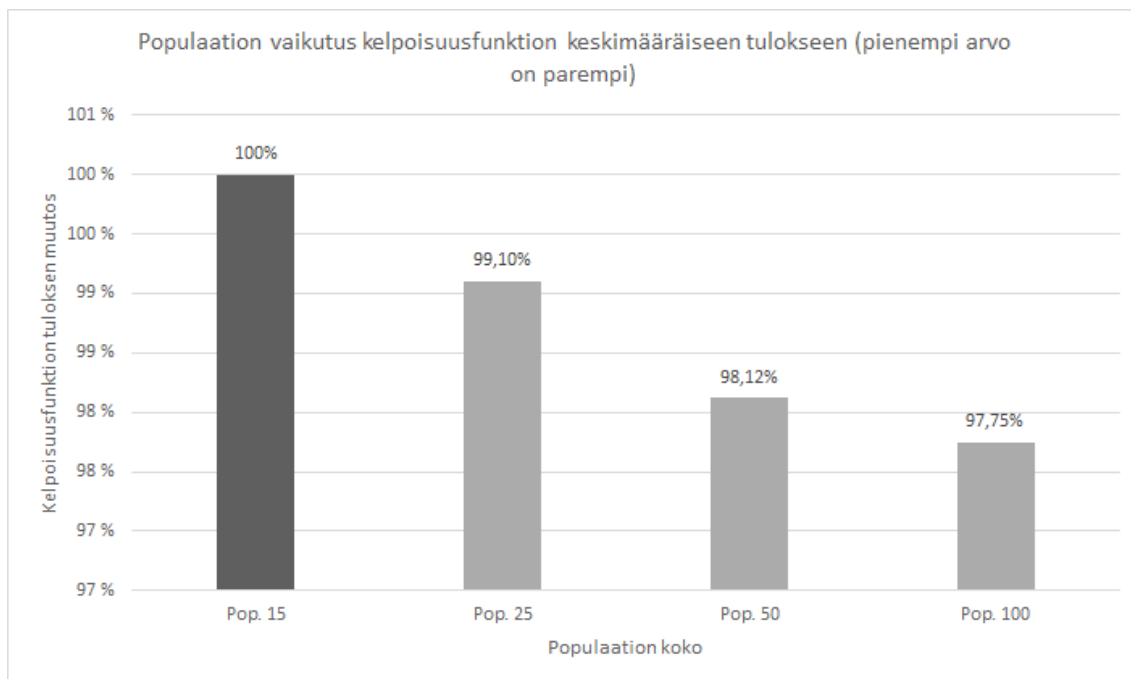


KUVIO 14: Esimerkkiratkaisun kelpoisuusfunktion tuloksen kehittyminen

7.2.1 Populaation koon vaikutus algoritmin toimintaan

Tulokset näyttävät selvästi, että suuremmalla populaation koolla saavutetaan parempia tuloksia (kuvio 15). Populaation kokoa kasvattamalla viidestätoista

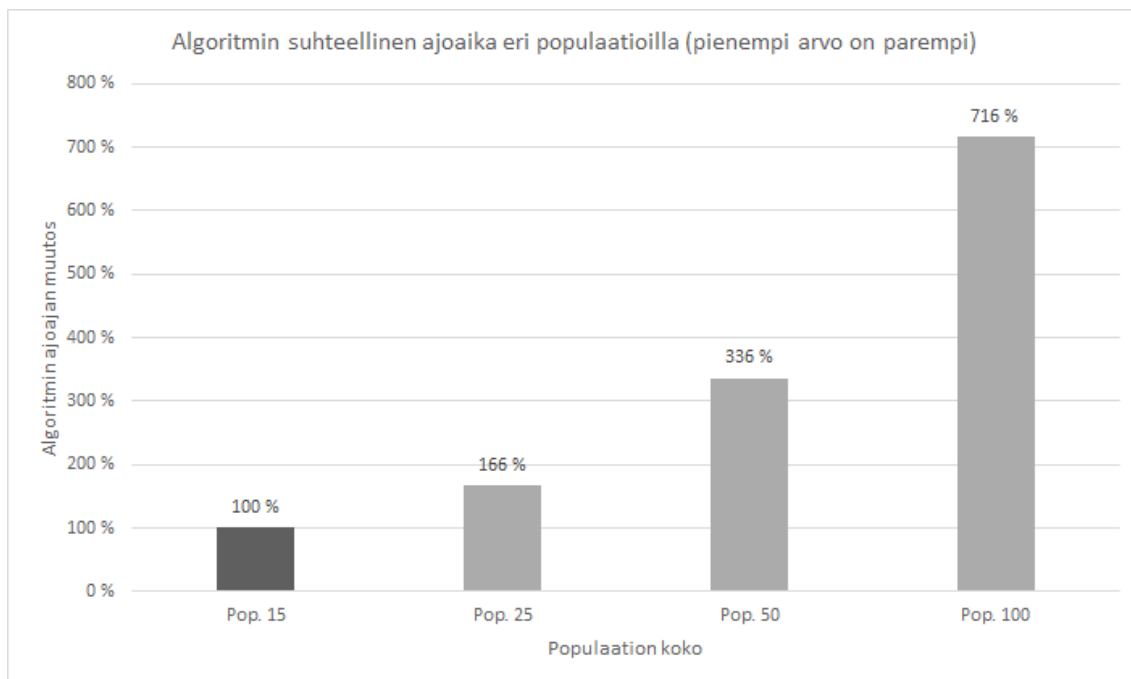
kromosomista sataan paransi keskimääräistä kelpoisuusfunktion parhainta tulosta 2,25 prosentilla. Suurempi populaatiokoko tuo kromosomijoukkoon enemmän vaihtelevuutta, jotka johtavat parempien ratkaisujen saavuttamiseen. Pienemmällä populaatiolla kromosomit eivät todennäköisesti eroa niin paljon toisistaan ja eivät siten mahdollista uusien parempien ratkaisuvaihtoehtojen kehittymistä samaan tahtiin kuin suuremmalla populaatiolla.



KUVIO 15: Populaation koon vaikutus kelpoisuusfunktion keskimääräiseen tulokseen

Kun verrataan algoritmin ajoaikaa eri populaatiomäärillä, käy ilmi, että suuremman populaation hyödyt tuloksessa tulevat ajoajan moninkertaistumisen kustannuksella (kuvio 16). Populaation koon kaksikertaistuessa kaksinkertaistuu likimain myös algoritmin ajoaikakin. Kun populaation koko on 100, algoritmin ajoaika on hieman yli kuusi kertaa pidempi kuin 15 kromosomin populaatiolla. Demonstroinnissa käytetyllä kannettavalla tietokoneella keskimääräinen ajoaika 15 kromosomin populaatiolla 10 000 sukupolven asti oli 7,5 minuuttia ja 100 kromosomin populaatiolla 52 minuuttia. Näiden havaintojen pohjalta kysymykseksi jää, onko suhteellisen pieni tuloksen parannus pidentyneen ajoajan arvoinen?

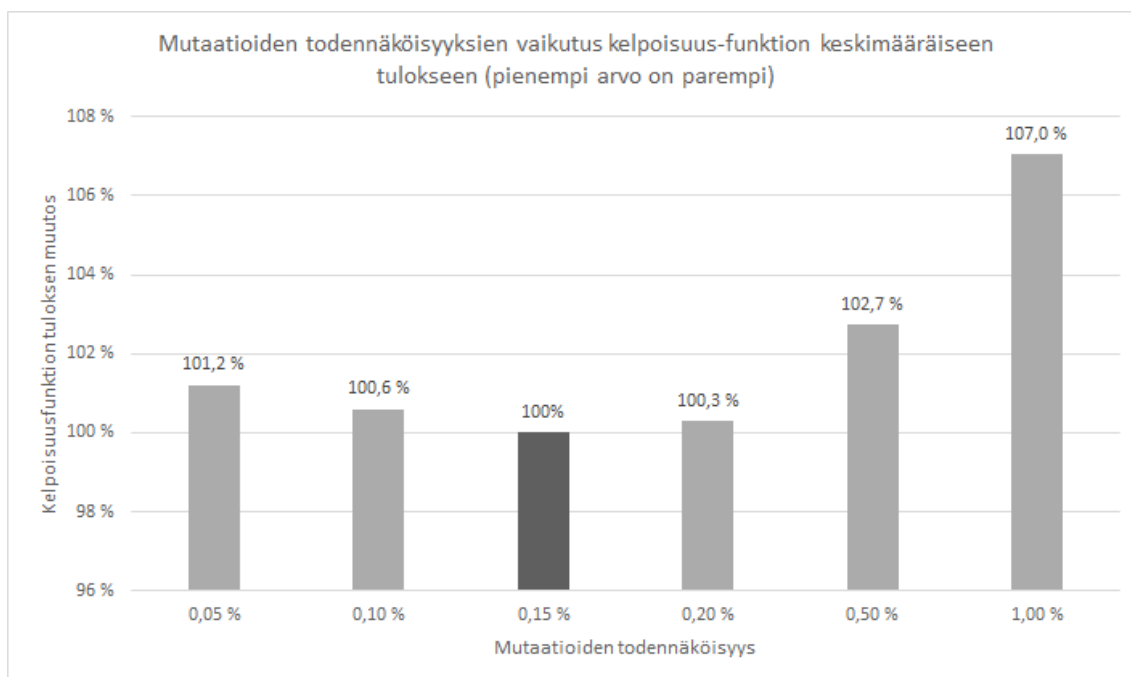
Vastaus riippuu, minkälaiseen automaatiojärjestelmään algoritmia hyödynnettäisiin reaali maailmassa. Jos varastoon tulevat tuotteet ovat päivittäin hyvin erilaisia edellisiin päiviin verrattuna, algoritmia kannattanee ajaa päivittäin, jolloin ajoajan minimoiminen saa tulosta suuremman prioriteetin. Jos taas tuotteet ovat päivä päivältä pitkälti samoja, eikä tilausrakenteeseen poikkea huomattavasti, yhden ajokerran tuottamaa varastojärjestystä voitaneen käyttää verrattain pitkiäkin aikoja. Tällöin algoritmin kannattaa antaa etsiä pidempään parhainta mahdollista tulosta, mahdollisesti 20 000 sukupolven tai pidempäänkin asti, riippuen kuinka paljon yhden sukupolven evoluutio ja arviointi kyseisellä varastokonfiguraatiolla kestää.



KUVIO 16: Populaation koon vaikutus algoritmin ajoaikaan

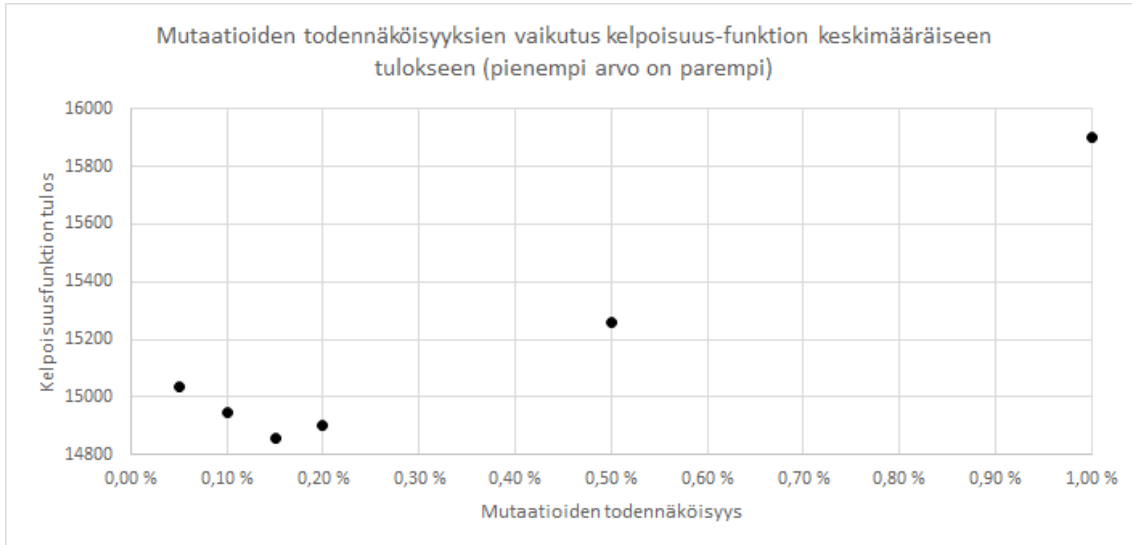
7.2.2 Mutaatioiden todennäköisyyden vaikutus algoritmin toimintaan

Mutaatioiden todennäköisyys -parametrin arvot eivät vaikuta algoritmin ajoaikaan, joten tuloksen parantamisen kannalta ei tarvitse miettiä samanlaista kompromissiratkaisua ajoajan kanssa kuten populaation koolla. Tuloksista selviää, että tiettyyn rajaan asti mutaatioiden todennäköisyyttä kasvattamalla on positiivinen vaikutus kelpoisuusfunktion keskimääräiseen tulokseen (kuvio 17).



KUVIO 17: Mutaatioiden todennäköisyyden suhteellinen vaikutus tulokseen

Demonstroinnissa käytetyllä varastokartalla ja tuotepinoilla optimaalisin mutaatioiden todennäköisyys oli noin 0,15 %. 310 varastopaikan varastolla tämä tarkoittaa keskimäärin vain 0,465 varastopaikan keskinäistä vaihtoa jokaisessa kromosomissa, joista pieni osa on myös merkityksettömiä tyhjien varastopaikkojen välisiä keskinäisiä vaihtoja. Jos kelpoisuusfunktion keskimääräiset tulokset eri mutaatioarvoilla laitetaan pylväsdiagrammin sijasta koordinaatistoon, jossa x-akselin suhde on oikea, asettuvat myös kuvion datapisteet melko hyvin tukemaan 0,15 % olevan tuloksen optimoinnin kannalta lähellä parhainta arvoa kyseisellä konfiguraatiolla (kuvio 18).



KUVIO 18: Mutaatioiden todennäköisyyden vaikutus tulokseen

Sopiva mutaatioiden todennäköisyys jouduttaneen etsimään reaali maailmassa jokaiselle varastokonfiguraatiolle erikseen, koska esimerkiksi suuremmalla varastolla 0,15 % todennäköisyydellä tapahtuu enemmän mutaatioita, kuin demonstroinnissa käytetyllä varastolla. Laskemalla todennäköisyyden, joka halutulla varastokonfiguraatiolla tuottaa noin 0,465 mutaatioita kromosomissa, tarjonnee kuitenkin hyvän lähtökohdan etsinnälle.

7.2.3 Muiden parametrien vaikutukset

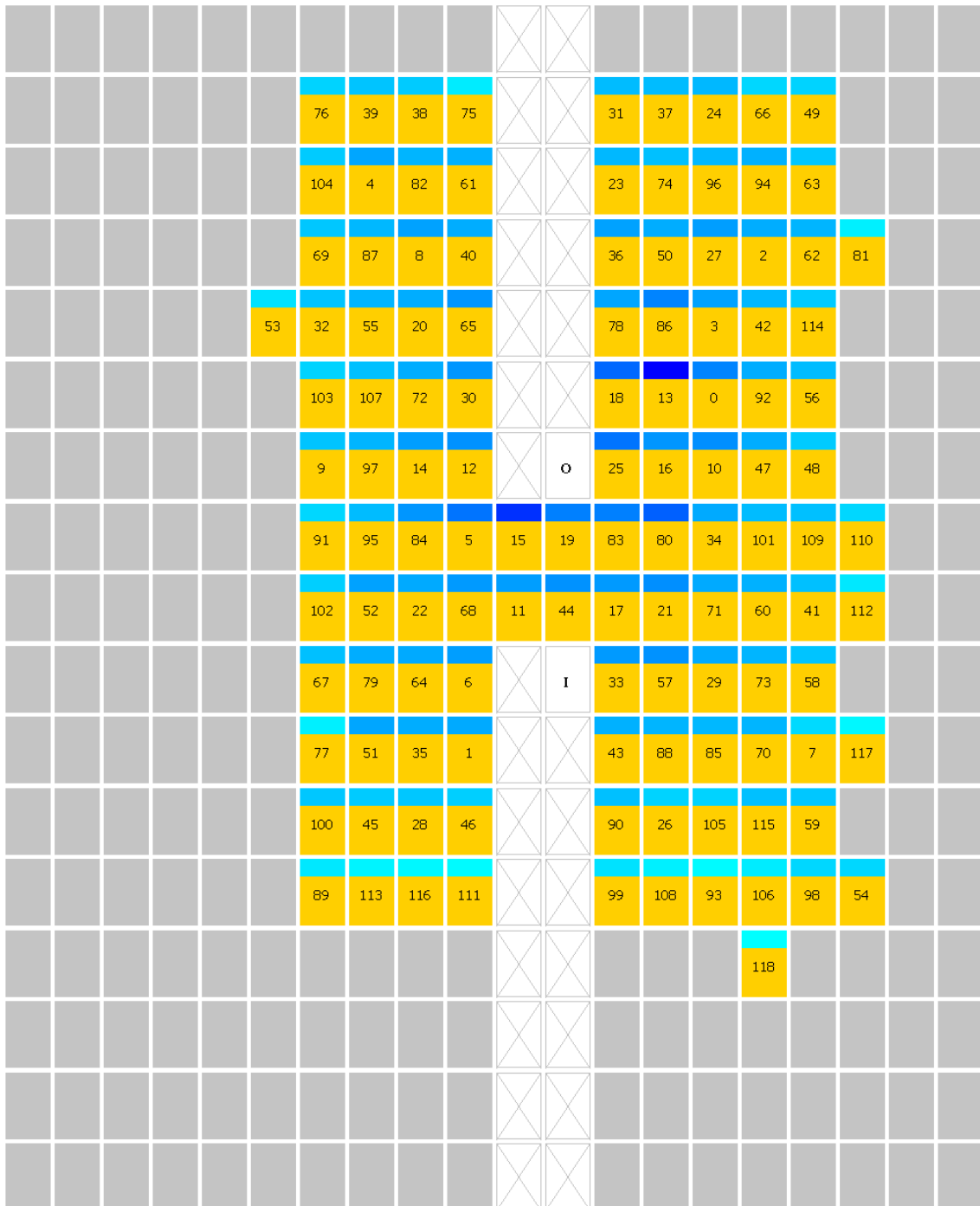
Vaikka demonstroinnissa muuttuvina parametreina käytettiin vain populaation koko ja mutaatioiden todennäköisyyttä, muuttamalla muiden parametrien arvoja voidaan paremmin hahmottaa algoritmin toimintaa yleisesti. Kun *tarttujan kapasiteetti* -parametrin arvon asettaa olemaan 1 (mahdollinen ja käytetty konfiguraatio myös reaali maailmassa), algoritmille ei enää ole niin edullista ryhmitellä usein samoissa tilauksissa esiintyvien tuotteiden tuotepinoja lähelle toisiaan. Tämä tapahtuu siksi, että aina tuotteen poimimisen jälkeen robotti joutuu tyhjentämään täyteen tulleen tarttujan poisvientipaikalla. Lopputuloksena on, että usein tilatut tuotteet kannattaa sijoitella poisvientipaikan lähelle, pienentämään robotin edestakaisin kulkemista.

Esimerkkinä tästä on kuvio 19, jossa esimerkiksi paljon tilattujen tuotteiden 5 (varastopaikat E1, P2 ja O12) ja 13 (M5, Q7, H11, I11 ja O15) tuotepinot eivät ole niin lähellä toisiaan, eivätkä niin paljon hajautettuja varaston eri alueille, kuten kuviossa 13, jossa tarttujan kapasiteetin arvona oli 8. Koska samoilla tuotteilla voi olla useita tuotepinoja, kannattaa samojen tuotteiden tuotepinoja edelleen, mutta pienemmässä määrin, hajauttaa ja sijoitella samoissa tilauksissa esiintyvien tuotteiden lähettyville.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	
1					5	104	11	17	71	27			6	14	46	33	91	52	4		
2					62	102	30	10	48	100			37	50	19	5	25	76	109		
3				117	94	96	21	14	35	109			51	62	83	40	79	66	25		
4					35	105	88	47	70	61			0	72	29	42	71	18	89	107	
5					1	98	80	85	59	33			13	16	92	15	95	3	37	28	
6					32	24	8	57	15	90	107			22	73	64	41	38	110	72	86
7					56	64	60	87	68	12	30		O	4	3	6	31	13	16	2	58
8					9	43	103	78	28	44	15	65	56	36	83	12	86	65	36	20	31
9					73	47	7	9	114	58	43	94	32	82	55	45	60	23	39	19	
10					92	11	84	24	10	80		I	67	8	27	19	86	78	70	0	
11					85	112	25	1	13	13	74			21	17	20	115	57	68	51	90
12					23	101	63	18	97	69			52	44	5	2	18	29	87		
13					55	34	80	49	84	15			0	22	54	34	26	99	53	41	
14					83	42	50	106	82	88			61	96	97	108	40	79	75		
15					95	100	67	116	101	77	21		74	113	13	81	111	115	93		
16						63	59	118		69					10	39					
17																					

KUVIO 19: Esimerkkiratkaisu, kun tarttujan kapasiteetti on 1

Mikäli tarttuvan kapasiteetti on 1 ja lisäksi pinon koko on ääretön (mahdoton konfiguraatio reaali maailmassa), eli yhtä tuotetta on vain ja ainoastaan yksi pino, tällöin optimaalisin ratkaisu olisi sijoitella pinot poisventipaikan ympärille järjestyksessä alkaen kaikkein tilatuimmasta tuotteesta. Esimerkkinä tästä on kuvio 20, jossa on tilanne 10 000 sukupolven jälkeen.



KUVIO 20: Esimerkkiratkaisu, kun tarttuvan kapasiteetti on 1 ja jokaista tuotetta on vain yksi pino

8 POHDINTA

Artefaktin ajokertojen tulokset osoittavat selvästi geneettisten algoritmien potentiaalın tuotteiden sijoitteluongelman ratkaisemisessa. Luotu simulaatio kykenee selvästi parantamaan tilausten tuotteiden hakemiseen kuluva yhteismatkaa verrattuna varastoon, jossa samat tuotteet ovat satunnaisesti sijoiteltu varastoon: keskimäärin parannusta oli parhaimmalla parametrijyhdistelmällä noin 40 % 10 000 sukupolven mennessä. Jos algoritmin annettaisiin ajaa pidempään, parannusta tulisi vielä jonkin verran lisää, mutta silloin pidentynyt ajoaika ei välttämättä olisi enää yhä pienentyvien tulosparannusten arvoinen.

Yksi geneettisten algoritmien suurimmista eduista on se, että algoritmi ei vaadi lainkaan kohdealueen asiantuntemusta tuotteiden sijoittelusta, vaan algoritmi kykenee oppimaan sen itse aiempien tilausten perusteella. Mikäli ihmiselle annettaisiin sijoiteltavaksi varastoon samat tuotepinot kuin artefaktin demonstroinnissa, intuitiivisesti ajateltuna paljon tilatut tuotteet tulisi sijoiteltua kaikkein lähimmäksi poisvientipaikkaa ja vähän tilatut tuotteet jäisivät reunoille, kuvion 20 kaltaisesti. Tuotteet, joita on useampi tuotepino, ihminen todennäköisesti sijoittelisi eri puolille varastoa, muttei todennäköisesti kovinkaan kauaksi poisvientipaikasta.

Geneettinen algoritmi kykenee nopeassa ajassa kokeilemaan lukuisia eri järjestyksiä tuotteille ja arvioimaan niiden suorituskykyä aiempien tilausten avulla. Tämä mahdollistaa sellaisten tehokkaiden ratkaisujen syntymisen, joita ihminen ei pystyisi järkevässä ajassa itse löytämään. Erityisesti kun varastossa on useita kymmeniä eri tuotteita, ihminen ei myöskään kykene itse selvittämään parhainta järjestystä, jolla usein keskenään tilatut tuotteet kannattaisi sijoitella. Geneettisten algoritmien avulla tämä on myös mahdollista ilman tilausdatan manuaalista analysointia ja kohdealueen asiantuntemusta.

Ihminen voi kuitenkin algoritmin parhaimman ratkaisun tuottamisen jälkeen havaita visuaalisesti selvästi näkyvissä olevat optimointimahdollisuudet, joita algoritmi ei välttämättä itse löytäisi niin helposti. Esimerkiksi kuviossa 13 voitaisiin helposti parantaa ratkaisua siirtämällä kaikki erillään olevat tuotepinot pääjoukon viereen, jotta kaikki tuotepinot olisivat tiivisti poisvientipaikan ympärillä. Tämä olisi helppoa visuaalisella käyttöliittymällä, joka näyttäisi heti tuotepinon siirron jälkeen vaikutuksen kelpoisuusfunktion tulokseen. Näiden

korjausten tekeminen vaatisi algoritmilta todennäköisesti useamman tuhannen sukupolven evoluution.

Ideaalitilanteessa reaali maailman portaalirobotteja ohjaavalle järjestelmälle voitaisiin antaa demonstroinnissa käytetyt tuotepinot sijoiteltavaksi samantyyppiseen varastoon ja saada artefaktin ratkaisun kanssa vertailukelpoinen tuotteiden sijoittelujärjestys. Reaali maailman järjestelmän ratkaisu voitaisiin sitten ajaa artefaktin kelpoisuusfunktion läpi, jolloin saataisiin tulos, jonka perusteella voitaisiin arvioida pystyykö geneettisten algoritmien avulla todella parantamaan tuotteiden sijoittelua. Käytännössä yllä olevan vertailun toteuttaminen on äärimmäisen hankalaa. Ensinnäkin tämä edellyttäisi, että kelpoisuusfunktiossa mallinnettaisiin tuotteiden keräily täysin samaksi, kuin mitä se oikeassa järjestelmässä olisi. Toisekseen reaali maailman järjestelmässä on lukuisia muuttujia ja tilannekohtaisia tekijöitä, joilla on vaikutusta niin tuotteiden keräilyyn kuin sijoitteluunkin. Artefaktin toteutuksessa näitä kaikkia muuttujia ei pystytä huomioimaan tai niiden toteutus ei onnistuisi järkevässä ajassa.

Näin ollen artefaktin arviointi reaali maailman tuotantokäyttöön soveltuvana perustuu tämän tutkimuksen puitteissa mitattujen arvojen sijaan enimmäkseen subjektiiviseen näkemykseen. Tarkkoja mittareita voitiin kuitenkin käyttää artefaktin toimintaa kontrolloivien parametrien optimoinnissa, jonka avulla saatiin tietoa millä tavoin geneettisiin algoritmeihin perustuva tuotesijoittelu voitaisiin tulevaisuudessa ottaa tuotantokäyttöön. Artefaktin tuottamia ratkaisuja tarkastelemalla voidaan todeta, että artefakti kykenee riittävän hyvin ratkaisemaan tuotesijoitteluongelma demonstroinnissa käytetyssä rajoitetussa ympäristössä. Sopivilla parametreilla algoritmi kykenee kohtuullisessa ajassa tuottamaan käyttökelpoisen sijoittelujärjestyksen tuotteille.

Yksi geneettisten algoritmien heikkouksista on se, että mikäli kelpoisuusfunktion syötteenä käytetty tilausdata vaihdetaan, myös algoritmin ratkaisun etsinnän on aloitettava alusta. Vanhan ratkaisun käyttäminen pohjana päivittyneelle tilausdatalle johtaa todennäköisesti ongelmiin populaation varianssin suhteen. Kaikki vanhan populaation kromosomit ovat sopeutuneet sijoitteluun vanhan tilausdatan pohjalta, eivätkä ne siten pysty muuttamaan uudelle tilausdatalle sopivaksi muuta kuin satunnaisten mutaatioiden avulla. Aloittamalla evoluution alusta täysin satunnaisella populaatiolla, kromosomit ohjautuvat todennäköisesti nopeammin uudelle tilausdatalle optimaalimpaan ratkaisuun. Uudelleen aloittaminen on kuitenkin vain ongelma, jos ratkaisun etsiminen kestää ajallisesti liian kauan. Demonstroinnissa käytetyn puolet tavanomaisesta pienemmän portaalirobottisolun kohdalla ajoaika ei muodostu ongelmaksi, mutta skaalattaessa isoon tai useamman solun käsittävään järjestelmään tilanne voi olla toinen.

Sovellettavasta ympäristöstä riippuen algoritmin pitkän ajoajan ja uudelleen aloittamisen takia geneettiset algoritmit sopivat parhaiten varastoille, jossa käsitellään pääsääntöisesti samoja tuotteita ja jossa tilauskanta ei vaihtele rajusti. Tällöin algoritmin tuottamaa ratkaisua voitaisiin käyttää pidempiä aikoja ja algoritmin ratkaisun etsintään olisi perusteltua käyttää myös enemmän aikaa. Kelpoisuusfunktiossa voitaisiin käyttää tuhansia tilauksia eri ajanjaksoilta, joiden avulla voitaisiin löytää tilauksissa esiintyvät trendit ja ottamaan ne huomioon ratkaisussa. Esimerkiksi elintarvikkeiden varastot voisivat sopia käyttö-

kohteiksi, sillä niiden kysyntä pysynee ympäri vuoden samankaltaisena kausiluontoisia tuotteita lukuun ottamatta. Algoritmin avulla voitaisiin myös luoda valmiiksi eri sijoittelujärjestyksiä esimerkiksi eri vuodenajoille ja juhlapyhille ottamaan huomioon kausiluontoisen kysynnän vaihtelun.

Luodussa artefaktissa on tällä hetkellä joitakin rajoitteita, jotka täytyy ottaa huomioon, ennen kuin sitä voitaisiin soveltaa tuotantokäyttöön. Suurimpana rajoitteena on se, että kelpoisuusfunktiossa mallinnetaan vain yhden robotin toiminta. Tyypillisesti yhdessä robottisolussa toimii kaksi robottia, jotka keräilevät tuotteita. Toisen robotin toiminnan mallintaminen tuo simulaatioon lisää kompleksisuutta, joka näkyy pienenä lisänä algoritmin ajoajassa. Toisen robotin mallintamisessa joudutaan muun muassa ottamaan huomioon miten tilaukset kannattaa jakaa eri roboteille, sekä toteuttaa törmäyksenesto, eli varmistaa, että robottien x-akselit pysyvät aina vähintään tietyn minimivaroetäisyyden päässä toisistaan.

Toisena rajoitteena on se, että simulaatio käsittää vain yhden robottisolun toiminnan. Useassa tapauksessa esimerkiksi tehtaassa tai jakelukeskuksessa on useita robottisoluja, jotka voivat osallistua saman tilauksen keräilyyn. Robottisolujen lisäksi osana automaatiojärjestelmää voi olla korkeavarasto, josta kuljetetaan täydennyksinä tuotteita robottisoluille. Yksittäinen robottisolu ei siis ole täysin erillinen entiteetti, vaan osa suurempaa järjestelmää. Tämän seurauksena myös artefaktissa täytyisi tunnistaa kokonaisjärjestelmän vaikutukset riittävällä tasolla ja ottaa ne huomioon.

Kolmantena rajoitteena voidaan pitää sitä, että artefaktissa kaikkien tuotepinojen oletetaan tulevan samalla kertaa varastoon ennen kelpoisuusfunktion tilausten keräilyä. Oikeassa tuotantoympäristössä näin ei kuitenkaan tapahdu, vaan tuotepinoja tulee myös varastoon tilausten keräilyn ollessa jo käynnissä. Tämän seurauksena algoritmin tulee kyetä käsittelemään tilanteita, jossa uudelle varastoon tulevalle tuotepinolle pitää löytää varastopaikka. Mahdollisena ratkaisuna tähän on, että algoritmin alustuksessa luodaan ylimääräisiä tuotepinoja, joilla täytetään kaikki tyhjät varastopaikat. Ongelmaksi kuitenkin tässä tapauksessa jää, miten toimitaan täysin uuden tuotteen tullessa varastoon, tai jos yhden tuotteen pinoja tulee varastoon enemmän kuin sille on algoritmin alustuksessa varattu paikkoja.

9 YHTEENVETO

Tässä tutkielmassa tutkittiin tekoälyn hyödyntämistä sisälogistiikan automaatiojärjestelmien ohjauksessa. Tutkielma tehtiin toimeksiantona tämänkaltaisia automaatiojärjestelmiä valmistavalle Cimcorp Oy:lle. Aihe on mielekäs tutkimuksen kannalta, koska viime vuosina tekoälyteknologiat ovat kehittyneet huomattavasti ja tämän seurauksena niiden avulla voidaan toteuttaa tehokkaita oppimiseen ja autonomisuuteen perustuvia ratkaisuja, jotka eivät aiemmin olleet mahdollisia. Sisälogistiikka, sekä yleisesti logistiikan ja toimitusketjun hallinta ovat keskeinen osa nykypäivän organisaatioita, ja yhä globalisoituvamassa maailmassa niiden merkitys ei todennäköisesti tule ainakaan väheneään.

Tutkielman tavoitteena oli vastata tutkimuskysymykseen:

- Miten tekoälyä voi soveltaa sisälogistiikan automaatiojärjestelmien ohjauksessa?

Tutkielma toteutettiin DSRM-prosessia käyttäen, jonka tuloksena syntyi simuloitun portaalirobotivaraston tuotesijoittelun ratkaiseva artefakti, joka hyödyntää geneettisiä algoritmeja. Kirjallisuuskatsauksessa selvisi, että alaan liittyvä tutkimus on toistaiseksi ollut vähäistä. Aiempien tutkimusten pohjalta tekoälyä on hyödynnetty seuraavissa sisälogistiikan automaatiojärjestelmien alueilla:

- keräilyongelman ratkaisemiseen AGV-vaunujen tai korkeavarastojen tapauksissa
- sisälogistiikan automaatiojärjestelmien ja niihin liittyvien tietojärjestelmien, kuten WMS:n suunnitteluun

Huomattavaa oli kiinnostus erityisesti AGV-vaunuihin. Tekoälyyn liittyvistä teknologioista hyödynnettiin seuraavia:

- geneettiset algoritmit
- moniagenttijärjestelmät

- parviäly ja -optimointi
- sumea logiikka
- vahvistusoppiminen

Näistä geneettiset algoritmit ja moniagenttijärjestelmät olivat eniten esillä tutkimuksissa.

Luotu artefakti osoitti geneettisten algoritmien potentiaalin portaalirobotivaraston tuotesijoitteluongelman ratkaisemisessa. Simuloidussa portaalirobotivarastossa geneettisten algoritmien avulla saatiin sopivilla parametreilla kohutuullisessa ajassa luotua toimivia tuotesijoitteluratkaisuja varaston tuotepinoille. Satunnaiseen ratkaisuun verrattuna algoritmi kykeni optimoimaan kelpoisuusfunktion tulosta keskimäärin 40 % parhaimmalla parametrijohdistelmällä. Lisäksi arvioimalla algoritmin eri parametrien arvojen vaikutusta algoritmin tulokseen saatiin käytännön tietoa, miten geneettisiä algoritmeja kannattaisi ottaa käyttöön reaali maailman automaatiojärjestelmissä.

Suurimpina rajoitteena tutkimuksen kannalta olivat simulaation ja reaali maailman automaatiojärjestelmien eroavaisuudet. Tarkkojen tulosten saavuttamiseksi simulaation olisi käyttäydyttävä täysin samalla tavalla kuin reaali maailman automaatiojärjestelmän. Artefaktin merkittävimpiä puutteita olivat vain yhden robotin simuloiminen, kaikkien tuotteiden saapuminen samalla kertaa varastoon ja vain yhden portaalirobotisolun simuloiminen. Tästä huolimatta saavutetut tulokset vaikuttivat lupaavilta ja kannustavat jatkamaan tutkimuksia geneettisten algoritmien hyödyntämiseen reaali maailman automaatiojärjestelmissä.

Eri tekoälyteknologioiden ja niiden hyödyntämiskohteiden kartoittamisen kannalta tutkimuksen rajoitteena on mahdollisesti se, että aiempien tutkimusten hakuun käytetyt hakusanat olivat liian laajoja. Jos tutkimus keskittyy hyvin spesifiseen tekoälyn teknologiaan, se ei välttämättä nouse hakutuloksissa esille, koska tutkimuksessa ei välttämättä mainita sanaa "tekoäly" kertaakaan. Vastava tilanne saattaa olla myös sisälogistiikan osalta. Käyttämällä sisälogistiikkaa tarkempia hakusanoja, joitain relevantteja tutkimuksia olisi mahdollisesti ollut vielä löydettävissä. Näitä ongelmia pyrittiin kirjallisuuskatsauksessa välttämään käyttämällä myös joitain tarkempia hakusanoja, mutta useita eri hakusanojen yhdistelmiä jäi tämän tutkimuksen puitteissa vielä kokeilematta.

Tutkimuksia, joissa tekoälyn käyttöä olisi laajasti tarkasteltu sisälogistiikan automaatiojärjestelmissä, ei tiettävästi ole ennen tätä tutkimusta tehty. Tämä tutkimus tarjoaa siis yleiskuvan aiheesta, josta selviää kuinka tekoälyä on tällä hetkellä hyödynnetty ja toisaalta millä alueilla tutkimuksessa on puutteita. Tutkimuksia, jossa tekoälyn käyttöä olisi hyödynnetty portaalirobotivarastossa on myös erittäin harvakseltaan. Luotu artefakti osoittaa, että muissa tutkimuksissa käytetyt geneettiset algoritmit soveltuvat myös portaalirobotivarastoihin. Näin ollen tutkimus laajentaa siis tietämystä tekoälyn hyödyntämisestä portaalirobotipohjaisissa automaatiojärjestelmissä.

Jatkotutkimuksissa voitaisiin esimerkiksi soveltaa aiemmissä tutkimuksissa käytettyjä tekniikoita portaalirobotteihin, koska tämän hetkinen kiinnostus on lähes kokonaan kohdistunut toisenlaisiin automaatiojärjestelmiin. Tässä tutkimuksessa portaalirobotteihin sovellettiin geneettisiä algoritmeja, mutta kiin-

nostavaa olisi myös tietää, olisiko muilla yllä listatuilla teknologioilla ollut mahdollista saada parempia tuloksia samassa ajassa. Lisäksi aiemmissä tutkimuksissa on käsitelty vain yhdenlaista automaatiojärjestelmien teknologiaa, mutta usein automaatiojärjestelmissä on käytetty useita eri teknologioita, kuten AGV-vaunuja ja korkeavarastoja. Toisena jatkotutkimusaiheena voisi siis olla, kuinka tekoälyn avulla voitaisiin tehostaa useiden eri automaatiojärjestelmien teknologioiden yhteistoimintaa kokonaisuutena.

LÄHTEET

- Bainbridge, L. (1983). Ironies of automation. In *Analysis, Design and Evaluation of Man-Machine Systems 1982* (pp. 129-135).
- Battini, D., Calzavara, M., Persona, A., & Sgarbossa, F. (2015). A comparative analysis of different paperless picking systems. *Industrial Management & Data Systems*, 115(3), 483-503.
- Beyer, T., Yousefifar, R., Abele, S., Bordasch, M., Göhner, P., & Wehking, K. H. (2015). Flexible agent-based planning and adaptation of material handling systems. In *2015 IEEE International Conference on Automation Science and Engineering (CASE)* (pp. 1060-1065). IEEE.
- Borana, J. (2016). Applications of Artificial Intelligence & Associated Technologies. Department of Electrical Engineering, Jodhpur National University. Proceeding of International Conference on Emerging Technologies in Engineering, Biomedical, Management and Science.
- Brady, M. (1985). Artificial intelligence and robotics. *Artificial intelligence*, 26(1), 79-121.
- Brezovnik, S., Gotlih, J., Balič, J., Gotlih, K., & Brezočnik, M. (2015). Optimization of an automated storage and retrieval systems by swarm intelligence. *Procedia Engineering*, 100, 1309-1318.
- Bringsjord, S., & Schimanski, B. (2003). What is artificial intelligence? Psychometric AI as an answer. In *IJCAI* (pp. 887-893).
- Bukova, B., Brumerčikova, E., Cerna, L., & Drozdziel, P. (2018). The Position of Industry 4.0 in the Worldwide Logistics Chains. *LOGI-Scientific Journal on Transport and Logistics*, 9(1), 18-23.
- Cárdenas, J. J., Garcia, A., Romeral, J. L., & Andrade, F. (2009). A genetic algorithm approach to optimization of power peaks in an automated warehouse. In *2009 35th Annual Conference of IEEE Industrial Electronics* (pp. 3297-3302). IEEE.
- Chowdhury, G. G. (2003). Natural language processing. *Annual review of information science and technology*, 37(1), 51-89.
- Christopher, M. (2011). *Logistics & supply chain management*. Pearson UK.
- Cimcorp. (2019a). Cimcorp-Konserni. Haettu 29.1.2019 osoitteesta <https://www.cimcorp.com/fi/cimcorp/cimcorp-konserni>
- Cimcorp. (2019b). Kuva haettu 14.3.2019 osoitteesta <https://www.cimcorp.com/en/logistics-automation/products-and-solutions/robotics>
- De Koster, R., Le-Duc, T., & Roodbergen, K. J. (2007). Design and control of warehouse order picking: A literature review. *European journal of operational research*, 182(2), 481-501.
- Delfmann, W., Ten Hompel, M., Kersten, W., Schmidt, T., & Stölzle, W. (2018). Logistics as a science: Central research questions in the era of the fourth industrial revolution. *Logistics Research*, 11(9), 1-13.

- Domingos, P. M. (2012). A few useful things to know about machine learning. *Commun. Acm*, 55(10), 78-87.
- Dou, J., Chen, C., & Yang, P. (2015). Genetic scheduling and reinforcement learning in multirobot systems for intelligent warehouses. *Mathematical Problems in Engineering*, 2015.
- Drakaki, M., & Tzionas, P. (2017). Manufacturing scheduling using colored petri nets and reinforcement learning. *Applied Sciences*, 7(2), 136.
- Elements of AI. (2018). Haettu 28.2.2019 osoitteesta <https://course.elementsofai.com/1/1>
- Ene, S., & Öztürk, N. (2012). Storage location assignment and order picking optimization in the automotive industry. *The international journal of advanced manufacturing technology*, 60(5-8), 787-797.
- Envista. (2019). WMS vs. WCS vs. WES. Haettu 15.3.2019 osoitteesta <http://www.mhi.org/media/members/16455/131129732041154697.pdf>
- Faber, N., de Koster, R. M. B., & van de VELDE, S. L. (2002). Linking warehouse complexity to warehouse planning and control structure: an exploratory study of the use of warehouse management information systems. *International Journal of Physical Distribution & Logistics Management*, 32(5), 381-395.
- Graves, A., Mohamed, A. R., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In 2013 IEEE international conference on acoustics, speech and signal processing (pp. 6645-6649). IEEE.
- Güller, M., & Hegmanns, T. (2014). Simulation-based performance analysis of a miniload multishuttle order picking system. *Procedia CIRP*, 17, 475-480.
- Haupt, R. L. (1995). An introduction to genetic algorithms for electromagnetics. *IEEE Antennas and Propagation Magazine*, 37(2), 7-15.
- Hevner, A. R., March T. M., Park J., Ram S. (2004). Design science in information systems research. *MIS quarterly*, 28(1), 75-105.
- Hirschberg, J., & Manning, C. D. (2015). Advances in natural language processing. *Science*, 349(6245), 261-266.
- Hitomi, K. (1994). Automation – its concept and a short history. *Technovation*, 14(2), 121-128.
- IFR, International Federation of Robotics. (2018). Artificial Intelligence in Robotics. Haettu 5.3.2019 osoitteesta https://ifr.org/downloads/papers/Media_Backgrounder_on_Artificial_Intelligence_in_Robotics_May_2018.pdf
- Jain, A. K., Mao, J., & Mohiuddin, K. M. (1996). Artificial neural networks: A tutorial. *Computer*, (3), 31-44.
- Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245), 255-260.
- Jurczak, M. (2018). Logistics 4.0 in practice. Miniload technology uses maximum of warehouse space. Haettu 30.3.2019 osoitteesta <https://trans.info/en/logistics-4-0-in-practice-miniload-technology-uses-maximum-of-warehouse-space-92520>
- Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4, 237-285.

- Kaplan, J. (2016). *Artificial Intelligence: What everyone needs to know*. Oxford University Press.
- Kartnig, G., Grösel, B., & Zrnić, N. (2012). Past, state-of-the-art and future of intralogistics in relation to megatrends. *FME Transactions*, 40(4), 193-200.
- Kłosowski, G., Gola, A., & Amila, T. (2018). Computational intelligence in control of AGV multimodal systems. *Ifac-papersonline*, 51(11), 1421-1427.
- Konak, A., Coit, D. W., & Smith, A. E. (2006). Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety*, 91(9), 992-1007.
- Kuo, R. J., Kuo, P. H., Chen, Y. R., & Zulvia, F. E. (2016). Application of metaheuristics-based clustering algorithm to item assignment in a synchronized zone order picking system. *Applied Soft Computing*, 46, 143-150.
- Kurhinen, J. (2010). TIEA301 Kandidaattiseminaari. Haettu 31.1.2019 osoitteesta <http://users.jyu.fi/~kurhinen/tiea301/tutkimusmenetelma.html>
- Lambert, D. M., & Cooper, M. C. (2000). Issues in supply chain management. *Industrial marketing management*, 29(1), 65-83.
- Liao, S. H. (2005). Expert system methodologies and applications—a decade review from 1995 to 2004. *Expert systems with applications*, 28(1), 93-103.
- Lipowski, A., & Lipowska, D. (2012). Roulette-wheel selection via stochastic acceptance. *Physica A: Statistical Mechanics and its Applications*, 391(6), 2193-2196.
- Lee, C. K. M., Lv, Y., Ng, K. K. H., Ho, W., & Choy, K. L. (2018). Design and application of Internet of things-based warehouse management system for smart logistics. *International Journal of Production Research*, 56(8), 2753-2768.
- Lukka, K. (2014). Kari Lukka: Konstruktiivinen tutkimusote. Haettu 31.1.2019 osoitteesta <https://metodix.fi/2014/05/19/lukka-konstruktiivinen-tutkimusote/>
- Lummus, R. R., & Vokurka, R. J. (1999). Defining supply chain management: a historical perspective and practical guidelines. *Industrial Management & Data Systems*, 99(1), 11-17.
- Mansouri, M., Andreasson, H., Pecora, F. (2015) Towards Hybrid Reasoning for Automated Industrial Fleet Management. In: 24th International Joint Conference on Artificial Intelligence, Workshop on Hybrid Reasoning
- Muratec USA. (2019). Kuva haettu 14.3.2019 osoitteesta https://www.muratec-usa.com/wp-content/uploads/2017/09/AGV-Unit-Load_Magnetic-Tape-Guided-machine-only.jpg
- Nagel, L., Roidl, M., & Follert, G. (2008). The Internet of Things: On Standardisation in the Domain of Intralogistics. In *Proceedings of the First International Conference on The Internet of Things*, Zürich (pp. 16-21).
- Nau, D. S. (2007). Current trends in automated planning. *AI magazine*, 28(4), 43-43.
- Nau, D. S. (2010). Automated Planning. Haettu 6.3.2019 osoitteesta <https://www.cs.umd.edu/~nau/cmsc421/planning.pdf>
- Obermeier, P. (2018). Scalable Robotic Intra-Logistics with Answer Set Programming. In *Technical Communications of the 34th International*

- Conference on Logic Programming (ICLP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
- Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). A Model for Types and Levels of Human Interaction with Automation. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 30(3), 286-297.
- Peffer, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of management information systems*, 24(3), 45-77.
- PwC. (2018). *The Macroeconomic Impact of Artificial Intelligence*. Haettu 24.2.2019 osoitteesta <https://www.pwc.co.uk/economic-services/assets/macro-economic-impact-of-ai-technical-report-feb-18.pdf>
- Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), 257-286.
- Ratliff, H. D., & Rosenthal, A. S. (1983). Order-picking in a rectangular warehouse: a solvable case of the traveling salesman problem. *Operations Research*, 31(3), 507-521.
- Rubicite Interactive. (2019). Order 1 Crossover. Haettu 2.7.2019 osoitteesta <http://www.rubicite.com/Tutorials/GeneticAlgorithms/CrossoverOperators/Order1CrossoverOperator.aspx>
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Berg, A. C. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3), 211-252.
- Russell, S., Norvig, P., & Intelligence, A. (1995). A modern approach. *Artificial Intelligence*. Prentice-Hall, Englewood Cliffs, 25(27), 79-80.
- Salinas, E. (2017). Katie Malone on Machine Learning. *IEEE Software*, 34(4), 92-96.
- Savant Automation. (2019). AGV Configurations. Haettu 14.3.2019 osoitteesta <http://www.agvsystems.com/agvs/>
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61, 85-117.
- Searle, J. R. (1980). Minds, brains, and programs. *Behavioral and brain sciences*, 3(3), 417-424.
- Seidmann, A. (1988). Intelligent control schemes for automated storage and retrieval systems. *The International Journal Of Production Research*, 26(5), 931-952.
- Shen, W., Hao, Q., Yoon, H. J., & Norrie, D. H. (2006). Applications of agent-based systems in intelligent manufacturing: An updated review. *Advanced engineering INFORMATICS*, 20(4), 415-431.
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Stergiou, C. & Siganos, D. (1997). Neural Networks. Haettu 7.3.2019 osoitteesta https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html
- Stone, P., & Veloso, M. (2000). Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3), 345-383.

- Tutorialspoint (2019). Genetic Algorithms - Quick Guide. Haettu 20.5.2019 osoitteesta https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_quick_guide.htm
- Villanueva, B. M., & Salenga, M. L. M. (2018). Bitter Melon Crop Yield Prediction using Machine Learning Algorithm. *Int. J. Adv. Comput. Sci. Appl*, 9, 1-6.
- VTT. (2017). *Tuottoa ja tehokkuutta Suomeen tekoälyllä*. Haettu 24.2.2019 osoitteesta <https://www.vtt.fi/inf/pdf/policybrief/2017/PB1-2017.pdf>
- Wang, K. (2016, November). Logistics 4.0 Solution-New Challenges and Opportunities. In 6th International Workshop of Advanced Manufacturing and Automation. Atlantis Press.
- Weber, F., & Schütte, R. (2019). A Domain-Oriented Analysis of the Impact of Machine Learning—The Case of Retailing. *Big Data and Cognitive Computing*, 3(1), 11.
- Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and computing*, 4(2), 65-85.
- Wurman, P. R., D'Andrea, R., & Mountz, M. (2008). Coordinating hundreds of cooperative, autonomous vehicles in warehouses. *AI magazine*, 29(1), 9-9.
- Yu, D., & Deng, L. (2012). Tutorial 9: Deep Learning and Its Applications in Signal Processing. Haettu 7.3.2019 osoitteesta https://web.archive.org/web/20120718043709/http://www.icassp2012.com/Tutorial_09.asp
- Zhu, X. J. (2005). Semi-supervised learning literature survey. University of Wisconsin-Madison Department of Computer Sciences.