



TAMPEREEN TEKNILLINEN YLIOPISTO

VILLE TOIVONEN
HEURISTINEN OPTIMOINTIMENETELMÄ TILAUSOHJATUN
TUOTANNON KARKEASUUNNITTELUUN

Diplomityö

Tarkastaja: professori Minna Lanz
Tarkastaja ja aihe hyväksytty
Talouden ja rakentamisen
tiedekuntaneuvoston kokouksessa
9. joulukuuta 2015

TIIVISTELMÄ

VILLE TOIVONEN: Heuristinen optimointimenetelmä tilausohjatun tuotannon karkeasuunnitteluun
Tampereen teknillinen yliopisto
Diplomityö, 60 sivua
Syyskuu 2016
Tuotantotalouden DI-tutkinto-ohjelma
Pääaine: Teollisuustalous
Tarkastaja: professori Minna Lanz

Avainsanat: Lineaarinen optimointi, projektinohjaus

Tilausohjatussa tuotannossa lopputuotteita valmistetaan ainoastaan asiakastilausten perusteella. Valmiita tuotteita varastoidaan korkeintaan lyhytaikaisesti, joten kysynnän vaihdellessa tuotanto toimii usein tiukkojen aikataulupaineiden alaisena. Toisaalta pyrkimys taloudelliseen tehokkuuteen rajoittaa korkeaan kuormitustilanteeseen varatun vapaan kapasiteetin määrää. Tuotannosuunnittelun keskeisin tehtävä onkin löytää hyväksyttävä valmistusohjelma kustannusten ja aikataulun ristipaineessa.

Tutkimuksessa tätä ongelmaa on lähestytty matemaattisen optimoinnin keinoin. Suunnitteluongelmalle esitetään ensin formaali muotoilu, minkä jälkeen se pyritään ratkaisemaan asetettujen tavoitteiden suhteen. Tuotannosuunnitteluongelman matemaattista mallintamista ja ratkaisua vaikeuttavat erilaisten reunaehtojen ja muuttujien runsas lukumäärä, sekä tavoitteiden ristiriitaisuus. Laajojen ja vaikeasti ratkaistavien ongelmien käsittelyyn käytetään usein metaheuristisia menetelmiä, joilla pyritään saavuttamaan hyvä ratkaisu järkevässä ajassa.

Suunnitteluongelman muotoilussa aikataulujen noudattaminen on valittu reunaehdoksi, jonka puitteissa eri resurssiryhmien kuormitusta pyritään tasaamaan. Tasainen resurssikuormitus pienentää välivarastojen tarvetta, helpottaa tuotannonohjausta ja mahdollistaa järkevän investointisuunnittelun pitkällä aikavälillä. Työssä esitetään monivaiheinen heuristinen menetelmä tuotannon karkeasuunnittelun toteuttamiseen. Menetelmässä minimoidaan sekä resurssien yksittäisiä kuormitushuippuja että kuormituksen vaihtelua suunnitteluhorisontissa.

Työkaluksi on valittu Microsoft Excel sen yleisyyden ja lähestyttävyyden vuoksi. MS Exceliin on toteutettu käyttöliittymä, jolla hallitaan tuotantotietoja ja optimointisovellusta. Suunnittelualgoritmi on ohjelmoitu VBA-makroina (Visual Basic for Applications). Excel-tiedosto toimii siis itsenäisenä suunnittelutyökaluna sisältäen tietokannan, käyttöliittymän ja optimointialgoritmin.

Kaupalliseen optimointityökaluun verrattuna sovellettava menetelmä on huomattavan nopea ja tuottaa käytännön kannalta relevantteja tuloksia. Ratkaisumenetelmän nopeutta voidaan hyödyntää toistamalla ratkaisuproseduuri useaan kertaan, jolloin vaikeissakin ongelmissa lähestytään optimiratkaisua.

ABSTRACT

VILLE TOIVONEN: Heuristic optimization method for aggregate production planning in make-to-order production
Tampere University of Technology
Master of Science Thesis, 60 pages
September 2016
Master's Degree Programme in Industrial Engineering and Management
Major: Industrial Management
Examiner: Professor Minna Lanz

Keywords: Linear programming, Project scheduling

Customer orders are guiding actions in make-to-order production. Finished goods are stored only for brief periods of time and manufacturing has to cope with tight schedules under fluctuating demand. On the other hand, the pursuit of economic efficiency limits the availability of reserve capacity during peak demand periods. The key mission of aggregate production planning is to find a feasible schedule under the existing cost and time constraints.

This problem is approached with means of mathematical programming. The production planning problem is formulated and then solved regarding set criteria. The main difficulty of production planning problems comes from excessive number of variables and constraints, as well as conflicting objectives. Meta heuristic methods are often used to solve large and mathematically difficult problems. These are aimed to produce good results in short period of time, while not guaranteeing a global optimum.

The planning problem is modelled as a time constrained resource levelling problem. Levelled workload in different resource groups reduces the amount of WIP, facilitates short term production control and allows sensible long-term investment planning. A multi-stage heuristic planning method is presented for aggregate production planning. The method minimizes peak loads of individual resources as well as variation of workload over time.

The planning tool has been implemented in Microsoft Excel because of its prevalence among production managers. Production data and optimization parameters are managed through user interface implemented in an Excel file, while the algorithm is programmed in VBA (Visual Basic for Applications) macros. Therefore, the Excel file serves as an independent planning tool containing the database, user interface and the planning algorithms.

The presented planning method is considerably fast compared to a commercial general purpose optimization tool. The achieved results are applicable and relevant to practice. The speed of the optimization method allows several optimization runs in order to improve the solution in very difficult problems.

ALKUSANAT

Tämä diplomityö on kirjoitettu loppuun keväällä 2016. Se sitoo yhteen ison nipun vanhoja langanpätkiä ja pöytälaatikkomuistiinpanoja, mutta samalla tulee mieleen, ettei maailma ole ihan valmis tämänkään tutkimusalan osalta.

Suuret kiitokset tuesta kuuluvat perheelleni, professori Minna Lanzille, sekä koko Kone- ja tuotantotekniikan laitokselle. Lisäksi haluan erityisesti kiittää Aalto yliopiston professori Esko Niemeä neuvoista, ohjauksesta ja herkeämättömästä mielenkiinnosta tuotannon optimointia kohtaan.

Tampereella, 5.8.2016

Ville Toivonen

SISÄLLYSLUETTELO

1.	JOHDANTO	1
1.1	Tuotannosuunnittelu konepajateollisuudessa	1
1.2	Tutkimusote ja tutkimusongelma	2
1.3	Työn menetelmä ja rajaukset.....	2
1.4	Tutkimuksen sisältö ja rakenne	3
2.	TUOTANNON TILAUSOHJAUS	4
2.1	Tuotantoympäristö	4
2.2	Kysynnän vaihtelu	5
2.3	Tilausohjattu tuotanto	7
2.4	Suunnittelujärjestelmän vaatimukset tilausohjatussa tuotannossa	9
2.5	Imu- ja työntöohjaus	10
3.	TUOTANNONSUUNNITTELU	11
3.1	Suunnitteluprosessi	11
3.2	Tietojärjestelmät	13
3.3	Kapasiteetinsuunnittelun perusstrategiat.....	14
4.	PROJEKTINOHJAUSTEORIAT TUOTANNONSUUNNITTELUSSA.....	17
4.1	Ongelman muotoilu	17
4.2	Kuormitusvaihtelun tasoittamisongelma (RLP).....	18
4.3	Lineaarisen optimointiongelman ratkaiseminen.....	22
5.	TUOTANNON KARKEASUUNNITELUN OPTIMOINTIONGELMA.....	26
5.1	Jatkuvasti jakautuneet resurssit	26
5.2	Tavoitefunktion arvojen laskeminen	28
5.3	Ongelman matemaattinen muotoilu.....	32
5.4	Esimerkki optimointiongelman koosta	35
6.	OHJELMAN KUVAUS	37
6.1	Hierarkkinen ratkaisualgoritmi.....	37
6.2	Alihankinnan käsittely optimointisovelluksessa	41
6.3	Työnvaiheiden dynaaminen lyhentäminen	41
6.4	Ohjelmakoodi	42
6.5	Käyttöliittymän yleiskuvaus.....	44
6.6	Suunnittelutyökalun parametrit	46
6.7	Optimointitulokset	48
7.	TULOKSET	49
7.1	Koejärjestely.....	49
7.2	Optimointitulokset	50
8.	PÄÄTELMÄT	54
8.1	Suunnittelumenetelmä.....	54
8.2	Saavutetut tulokset.....	55
8.3	Kehitysmahdollisuudet	55
	LÄHTEET	57

LYHENTEET JA MERKINNÄT

AMP	engl. Adaptive Memory Programming, optimointimenetelmäluokka
APP	engl. Aggregate Production Planning, tuotannon karkeasuunnittelu
APS	engl. Advanced Planning and Scheduling, tuotannonsuunnitteluohjelma
ATO	engl. Assemble-To-Order, tilausohjattu kokoonpano
Brute force	matemaattisen todistuksen menetelmä
ConWIP	engl. Constant WIP, tuotannonohjausmenetelmä
DBR	engl. Drum-Buffer-Rope, tuotannonohjausmenetelmä
ERP	engl. Enterprise Resource Planning, toiminnanohjausjärjestelmä
ISA-95	tuotantotiedon tiedonsiirtostandardi
JIT	engl. Just-In-Time, tuotantofilosofia
LP	engl. Linear Programming, lineaarinen optimointi
MRP	engl. Material Requirements Planning, materiaalien tarvelaskentajärjestelmä
MRP II	engl. Manufacturing Resource Planning, tarvelaskenta ja kapasiteettisuunnittelujärjestelmä
NP-ongelma	engl. Nondeterministic Polynomial, matemaattinen ongelmaluokka
NP-vaikea	laskennallisen kompleksisuuden luokka
PPC	engl. Production Planning and Control, tuotannonsuunnittelu ja -ohjaus
RCSPSP	engl. Resource Constrained Project Scheduling Problem, kapasiteettirajoitteinen projektin aikataulutusergelma
RLP	engl. Resource Levelling Problem, kuormitusvaihtelun tasausongelma
SCM	engl. Supply Chain Management, toimitusketjunhallinta
MILP	engl. Mixed-Integer LP, LP-luokan alaluokka
MPS	engl. Master Production Schedule, päätuotantosuunnitelma
MTO	engl. Make-To-Order, tilausohjattu tuotanto
NPV	engl. Net Present Value, projektin nettonykyarvo
SIMPLEX	LP-ongelmien ratkaisuun kehitetty algoritmi
VBA	Visual Basic for Applications, makro-ohjelmointikieli
WIP	engl. Work In Process, keskeneräinen tuotanto

i	työnvaiheen indeksi
k	resurssin indeksi
t	ajanhetken indeksi
P_k	resurssin k kuormitushuippu
r_{ikt}	työnvaiheen i kuormitus resurssille k ajanhetkellä t
p_i	työnvaiheen i kesto
C_i	työnvaiheen i suunniteltu valmistumishetki
$O(i)$	työnvaiheen i sisältävän tilauksen työnvaiheiden joukko
x_{it}	vaiheen i ajanhetkellä t meneillään olon kertova apumuuttuja
a_i	työnvaiheen i sisältävän tilauksen aikaisin aloitushetki
b_i	työnvaiheen i sisältävän tilauksen maksimiläpäisy aika
d_i	työnvaiheen i sisältävän tilauksen deadline
W_{ik}	työnvaiheen i työkuorma resurssilla k
HZ	ajoitushorisontin pituus
M	suuri luku
n	työnvaiheiden lukumäärä
m	tilauksen ajoitusvara
z	tilausten lukumäärä

1. JOHDANTO

1.1 Tuotannonsuunnittelu konepajateollisuudessa

Konepajateollisuus kattaa merkittävän osan Suomen teollisuustuotannosta. Erityisen kilpailukykyisiksi ovat osoittautuneet pitkälti erikoistuneet ja asiakasräätälöidyt investointituotteet, joita valmistetaan teollisuuden tarpeisiin. Tämän tyyppistä tuotantoa ohjataan yleensä asiakastilausten perusteella, koska tuotteet ovat kalliita ja tilauskohtaisesti varioituvia. Kustannuspaineet tekevät tilausohjatun tuotannon suunnittelusta haastavaa, koska ”ylimääräistä” valmistuskapasiteettia ei ole varaa ylläpitää.

Tilausohjatussa tuotannossa kysynnän vaihteluun ei voida varautua varastoimalla lopputuotteita, mutta samalla toimitusvarmuus on kuitenkin merkittävä kilpailutekijä. Tuotannon karkeasuunnittelun tavoitteena on varmistaa, että yritys pystyy tyydyttämään tulevaisuuden kysynnän mahdollisimman taloudellisesti. Päätöksenteon kohteena ovat tuotanto- ja henkilöstömäärät, varastotasot, sekä alihankintaan liittyvät päätökset (Wang & Fang, 2001). Tyypillinen karkeasuunnittelun aikajänne on kahdesta kahteentoista kuukautta.

Tuotannonsuunnittelua voidaan lähestyä matemaattisena ongelmana, jossa tuotannon tavoitteet pyritään toteuttamaan tuotantojärjestelmän ja ympäristön asettamien reunaehtojen puitteissa. Tuotannonsuunnittelun tehtävänä on luoda tuotantoaikataulu, jonka perusteella kysyntä pystytään tyydyttämään yksittäisten tilausten aikatauluvaateiden mukaisesti. Suunnitelma vaikuttaa tuotannon kustannuksiin esimerkiksi valmistuksen kapasiteettisuunnittelun kautta. Tuotannon vaihtelu aiheuttaa tarpeen kapasiteetin säätelylle, mikä yleensä kasvattaa kustannuksia. Tuotannonohjaukselle suunnitelma antaa vertailukohdan, jonka perusteella tuotannon edistymistä voidaan seurata (Ballard, 2000).

Sekä tuotannolle asetetut tavoitteet että tuotantoa kuvaavat tekijät vaihtelevat toimiala- ja yrityskohtaisesti. Tuotannon tavoitteet liittyvät ensisijaisesti aikataulujen pitämiseen ja kustannusten minimointiin, mutta tarkemmassa tarkastelussa näihin voidaan pyrkiä lukuisilla eri tavoilla. Matemaattinen malli on aina yksinkertaistus todellisuudesta, jolloin haasteena onkin valita tuotantoympäristöä kuvaavat muuttujat, joilla on mahdollisimman suora vaikutus asetettuihin tavoitteisiin. Käytännössä kysymys ei ole ainoastaan valinnan vaikeudesta, vaan myös mittaamisen helppoudesta. Tuotannonsuunnittelun, -ohjauksen, sekä -seurannan tulee perustua tuotantojärjestelmästä saatavaan todelliseen informaatioon, joten tuotannon mittaus- ja raportointijärjestelmien rakenne vaikuttaa käytäntöön soveltuvaan mallinnustapaan.

1.2 Tutkimusote ja tutkimusongelma

Tutkimus noudattaa lähinnä liiketaloustieteissä esiintyvää konstruktivistista tutkimusotetta (Olkkonen, 1993). Työssä ei ole asetettu tutkimuskysymysten muotoon puettavaa tutkimusongelmaa, vaan tavoitteena on kehittää tilausohjatun tuotannon karkeasuunnitteluun soveltuvaa työkalua. Konstrukttiivinen tutkimusote on luonteeltaan empiirinen ja normatiivinen, ja kuuluu siis soveltavan tutkimuksen alueelle. Työn lopussa arvioidaan kehitetyn työkalun soveltuvuutta suunniteltuun käyttökohteeseen. Keskeisin tavoiteltava kontribuutio on työkalun algoritmin nopeuttaminen ja käyttöliittymän suunnittelu.

Tutkimuksen kohteena olevan suunnitteluongelman matemaattinen muotoilu on esitetty aiemmin (Niemi, 2004), joten menetelmän osalta tavoitteena on lähinnä ratkaisuheuristiikan parantaminen. Ongelma on matemaattisesti muotoilu kuvaamaan kapasiteettirajoittamaton kuormituksen tasausongelma (engl. Resource Levelling Problem, RLP). Heuristiikan käyttökelpoisuutta voidaan arvioida ratkaisunopeuden ja löydetyn ratkaisun hyvyyden perusteella.

1.3 Työn menetelmä ja rajaukset

Tutkimuksen kehityskohteena on tuotannonsuunnittelutyökalu, jonka käyttöliittymää ja ajoitusalgoritmia parannetaan. Työ sijoittuu vahvasti soveltavan tutkimuksen alueelle, minkä vuoksi käyttökelpoisuus teollisessa ympäristössä on tärkeänä kriteerinä eri ratkaisuvaihtoehtoja arvioitaessa. Työkalun kehitys perustuu deduktiiviseen päättelyyn sekä eri ratkaisuvaihtoehtojen systemaattiseen kokeiluun. Työkalun kehityksessä on hyödynnetty yrityksiltä saatua todellista tuotantodataa, mutta yksittäisten yritysten tilauskannalla tehtyjä optimointiajoja ei ole sisällytetty esitettäviin tuloksiin. Heurististen menetelmien tuottamien ratkaisujen epävarmuudesta johtuen menetelmän tehokkuutta arvioidaan satunnaisesti generoiduilla ongelmilla, joiden optimiratkaisut tunnetaan.

Ajoitusalgoritmin potentiaalisina hyödyntäjinä nähdään tyypilliset suomalaisen kone- ja metalliteollisuuden koneistavaa valmistusta ja kokoonpanoa harjoittavat yritykset. Tätä taustaa vasten on tehty myös suunnitteluongelman muotoilu, jossa on hyödynnetty sekä teollisuudesta saatua tuotantodataa sekä kokemusperäistä harkintaa. Suunnittelumenetelmää ei ole testattu muissa ympäristöissä, eikä sen sovellettavuutta kohderyhmän ulkopuolella arvioida. Yleisesti voidaan todeta, että usein tietyn ongelmatyyppin ratkaisuun suunnitellut menetelmät toimivat korkeintaan keskinkertaisesti muussa ympäristössä (Wolpert & Macready, 1997).

Tämä työ on rajattu käsittelemään konepajateollisuudessa tyypillistä yksittäiskappale- tai piensarjatuotantoa. Tällöin tuotannonsuunnittelun matemaattisessa mallinnuksessa

on luontevaa soveltaa projektinohjausteorioiden perusrakennetta, joissa tavoitteena on yhden tai useamman työvaiheisiin jaetun tuotteen valmistus aika- ja kapasiteettirajoitteiden puitteissa. Tarkastelu on rajoitettu lähinnä tehtaan sisäiseen toimintaan ja esimerkiksi alihankinta jätetään vähälle huomiolle. Tämä on normaali toimintatapa myös teollisuudessa silloin, kun valmistusketjut muodostuvat kohtalaisen itsenäisistä toimijoista.

Työssä ei keskitytä tuotannosuunnitteluprosessiin. Tämän alle kuuluvia aiheita ovat esimerkiksi suunnittelun aikajänne ja suunnittelutiheys, suunnittelun tarkkuustaso eli aikaresoluutio, päätökset mahdollisesta uudelleensuunnittelusta, sekä ohjausjärjestelmän vapausasteet. Osittain myös töiden vaiheistus ja resurssiryhmien muodostus voivat olla suunnitteluprosessiin kuuluvia päätöksiä. Suunnitteluprosessi vaikuttaa siis suunnitteluongelman haastavuuteen ja muotoiluun, mutta sisältää myös oman laajan ongelmakenttänsä, jonka analysointi rajataan työn ulkopuolelle.

Tuotannosuunnittelusta puhuttaessa ei voi olla törmäämättä lyhenteeseen MRP-järjestelmä (engl. Material Requirements Planning), joka oli alun perin tietokonepohjaiseen materiaalarvelaskentaan kehitetty työkalu. Tässä työssä MRP-lyhenteellä viitataan myöhempään MRP/MRP II (engl. Manufacturing Resource Planning) järjestelmään, joka kattaa myös valmistuksen kapasiteettisuunnittelun.

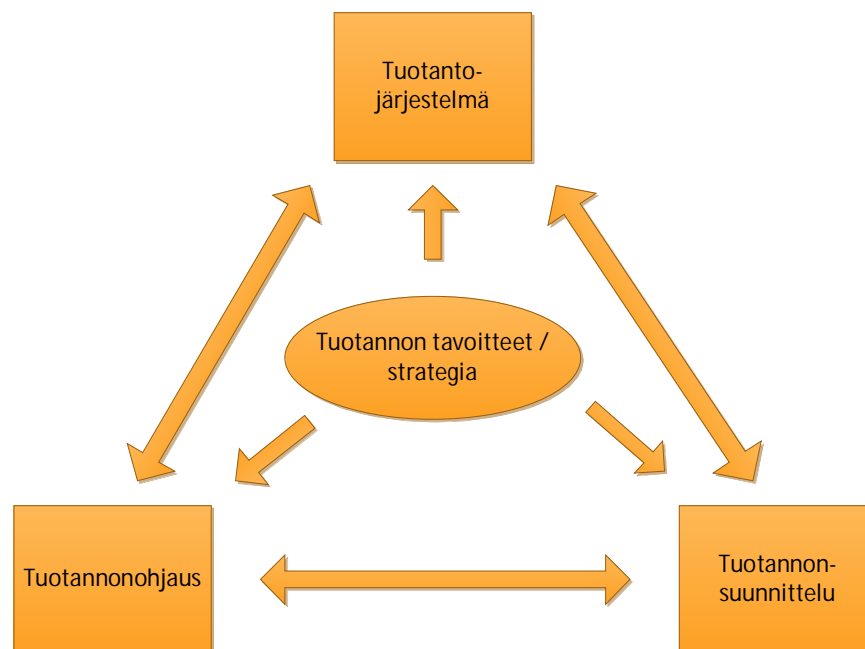
1.4 Tutkimuksen sisältö ja rakenne

Luvuissa 2 ja 3 tutustutaan tuotannon tilausohjaukseen ja tuotannosuunnittelu-järjestelmiin yleisellä tasolla. Luvussa 4 lähestytään suunnitteluongelman muotoilua projektinsuunnittelunäkökulmasta, ja työssä käytetty suunnitteluongelman muotoilu esitetään Luvussa 5. Luvuissa 6 ja 7 on kuvattu suunnittelutyökaluun kehitetty käyttöliittymä sekä ajoitusalgoritmi yksityiskohtaisesti. Luvussa 8 on esitetty tulokset suunnittelumenetelmän tehokkuuden testauksesta, minkä jälkeen työn loppuun on jäsennetty päätelmät ja arviointi kehityskohteena olevan työkalun sovellettavuudesta teolliseen tuotantoympäristöön, sekä potentiaaliset jatkotutkimusaiheet.

2. TUOTANNON TILAUSOHJAUS

2.1 Tuotantoympäristö

Tuotannon tavoitteiden tulisi olla linjassa niiden kilpailutekijöiden kanssa, jotka yritys on määrittänyt tärkeiksi yleisessä strategiassaan (Roth & Miller, 1992). Tuotantostrategiasta voidaan alkaa puhua, kun nämä kilpailutekijät on tunnistettu ja muutettu selkeästi priorisoiduiksi tavoitteiksi, joita hyödynnetään päätöksenteossa. Esimerkiksi Chen & Liao (2003) listaavat tuotannolle kuusi eri tavoitetta, joita voidaan käyttää karkeasuunnittelussa. Monissa yrityksissä tuotantoon ei oteta strategista näkökulmaa (Demeter, 2003), vaan päätöksenteko ristiriitaisten tavoitteiden välillä on tapauskohtaista. Kuva 1 on esitetty toimintaympäristö, johon tuotantostrategialla vaikutetaan.



Kuva 1. Tuotantostrategia ohjaa suunnittelua ja toimintaa

Tuotantojärjestelmä koostuu tehtaan työntekijöistä, koneista, tietojärjestelmistä ja muista valmistuksessa ja logistiikassa hyödynnettävistä resursseista, sekä prosesseista jotka kuvaavat näiden toimintoja. Näiden toimintojen aikataulutuksessa tarvitaan tuotannonohjausta. Tuotannonohjauksen tehtäviä ovat esimerkiksi töiden vapauttaminen tuotantoon, työjonojen järjestäminen ja työntekijöiden ohjaaminen (Hopp & Spearman, 2008). Ballard (2000) määrittää ohjausjärjestelmän keskeiseksi tehtäväksi toteutuneiden kustannusten tai aikataulun monitoroinnin ja vertaamisen suunnitelmaan, sekä negatiivisten poikkeamien havainnoinnin. Tuotannonsuunnittelun tulos on siis tavoite, jonka tuotannonohjaus pyrkii toteuttamaan.

2.2 Kysynnän vaihtelu

Tarve tuotannosuunnittelulle juontuu asiakaskysynnän vaihtelusta. Pitkällä aikavälillä tuotantomäärien täytyy vastata tyydytettävää kysyntää. Lyhyellä aikavälillä tuotannon ja lopputuotevarastojen tulisi vastata kysyntään, sillä vapaassa kilpailutilanteessa tyydyttämättä jäävää kysyntää ei yleensä voida siirtää myöhemmäksi. Asiakkaat siis valitsevat ne toimittajat, jotka pystyvät luotettavasti vastaamaan tarpeeseen. Tuotannon karkeasuunnittelun tavoitteena on varmistaa riittävä tuotantomäärä ja varastotasot lyhyen aikavälin kysynnän kattamiseen niin, että tuotanto on pitkällä aikavälillä mahdollisimman taloudellista.

Holt & al. (1960, ss. 48-49) esittävät kolme perusstrategiaa, joilla vaihtelevaan kysyntään voidaan sopeutua:

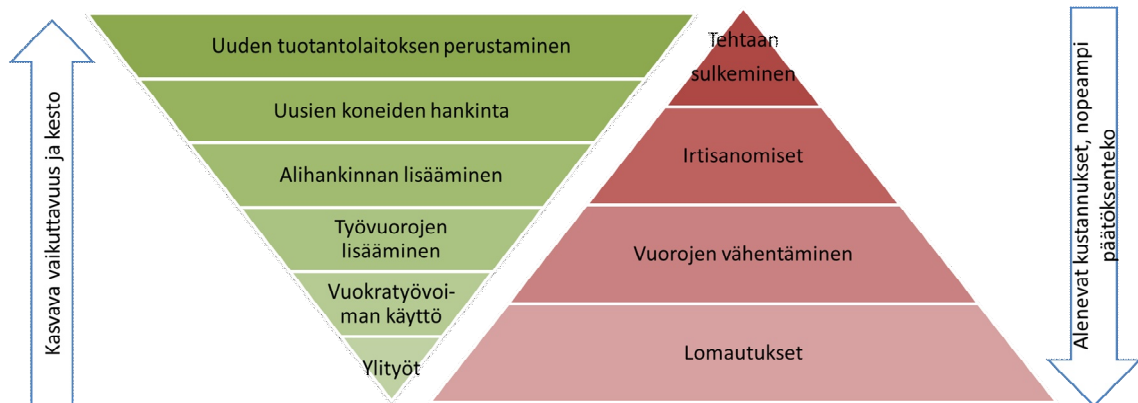
1. Työvoimaa palkataan ja irtisanotaan vaihtelevan kysynnän mukaan.
2. Tuotantomääriä sopeutetaan kysyntään vaihtelemalla työtuntien määrää.
3. Työvoima ja tuotantomäärät säilytetään vakiona ja kysynnän vaihtelua hallitaan lopputuotevarastojen avulla.

Listan kahdessa ensimmäisessä vaihtoehdossa kapasiteettia pyritään sopeuttamaan kysyntään. Tätä lähestymistapaa kutsutaan yleisesti kysynnän seuraamiseksi (engl. *chase strategy*). Kolmannessa vaihtoehdossa pyrkimyksenä on sen sijaan tasaisen tuotantovirran ylläpitäminen (engl. *level strategy*). Käytännössä kustannustehokkaimmaksi vaihtoehdoksi muodostuu jokin näiden kolmen kombinaatio, eikä niinkään yksittäinen strategia. Näiden edullisin painotus vaihtelee myös yksittäisessä yrityksessä kysynnän vaihtelun taajuuden ja suuruuden mukaan (Holt & al. 1960, s. 50).

Kapasiteetin sopeuttamiskeinot ovat muuttuneet 1960-luvun jälkeen, mutta pääperiaatteet ovat yhä samat; tuotanto joko seuraa tai tasoittaa kysyntää. Vakituisen työvoiman irtosanominen ei ole sosiaalisessa tai edes kustannustehokkaassa mielessä järkevä tapa reagoida lyhytaikaisen kysynnän vaihtelun aiheuttamiin muutoksiin. Sen sijaan vuokratyövoiman ja etenkin alihankinnan käyttö ovat lisänneet yritysten kapasiteettijoustavuutta. Konepajateollisuudessa päämiesyritykset ovat usein keskittyneet loppukokoonpanoon ja ulkoistaneet komponenttivalmistuksen alihankkijoille.

Vaihteleva kysyntä aiheuttaa paineita kapasiteetin säätämiseen varsinkin tilausohjatussa tuotannossa. Kysynnän kasvu johtaa helposti toimitusaikojen pitenemiseen, ja toisaalta laskeva kysyntä aiheuttaa kapasiteetin vajaakäyttöä ja lisää tuotannon

yksikkökustannuksia. Tämän vuoksi kapasiteettia pyritään aika ajoin säätämään kysyntää vastaavaksi. Kuva 2 on listattu yleisimpiä keinoja kapasiteetin sopeuttamiseen.



Kuva 2. Kapasiteetin sopeuttamiskeinoja

Uuden tuotantolaitoksen perustaminen on pitkäaikainen investointi, joka vaatii huolellista valmistelua sekä luotettavan ennusteen kysynnän pitkäaikaisesta kasvusta. Myös uusiin koneisiin investoiminen on pitkän vaikutusajan päätös. Sen sijaan listan neljä viimeistä kohtaa ovat keinoja, joita voidaan käyttää aktiivisesti kapasiteetin lisäämiseen. Alihankinnassa huomio kiinnittyy oikeiden yhteistyökumppaneiden sekä ulkoistettavien toimintojen valintaan. Työvuorojen lisääminen vaatii usein myös uuden työvoiman rekrytointia, mikäli työtehtäviä ei voida automatisoida. Vuokratyövoiman käyttö ja ylityöt ovat keinoja, joita käytetään usein lyhytaikaisen kapasiteettivajeen kompensoimiseen. Kapasiteetin vähentämisessä pääosin samat keinot toimivat toiseen suuntaan. Tosin tuotannossa käytettävien koneiden jälkimarkkinat ovat usein varsin kehnot, jolloin kapasiteettia rajoitetusti vähentävät päätökset kohdistuvat yleensä ensisijaisesti työvoimaan.

Kaikki edellä esitetyt kapasiteetin muuttamiseen käytetyt keinot aiheuttavat ylimääräisiä kustannuksia, jotka riippuvat eri tekijöistä. Kustannusten käyttäytyminen eri tilanteissa on myös paljon yritys- ja toimialakohtaista. Suorien kustannustekijöiden lisäksi kapasiteettimuutoksissa aiheutuu kustannuksia myös tuottavuuden muutosten kautta. Syitä tähän ovat esimerkiksi uusien työntekijöiden heikompi osaaminen ja alhaisempi tuottavuus, alihankinnassa tapahtuvat laaturvirheet ja liiallisen ylityömäärän aiheuttama työtehon lasku. Piper ja Vachon (2001) toteavat, että tuottavuuden heikkenemistä huomioimattomat kapasiteettisuunnittelumallit eivät tuota realistisia tuloksia. Samalla he ehdottavat, että lisätutkimusta tulisi suunnata kapasiteettimuutoksista aiheutuvan tuottavuuden heikkenemiseen vaikuttavien tekijöiden tarkempaan selvittämiseen.

2.3 Tilausohjattu tuotanto

Lopputuotteiden varastointi kasvattaa sitoutuneen pääoman kustannuksia, sekä sisältää muun muassa teknologisesti kehityksestä johtuvan epäkuranttiusriskin. Nämä tekijät aiheuttavat paineita lopputuotevarastojen supistamiseen varsinkin korkean jalostusarvon omaavien tuotteiden osalta. Muita oleellisesti tilausohjatun tuotannon suosiota lisänneitä tekijöitä ovat JIT-filosofiaan (engl. Just-In-Time) liittyvä pyrkimys varastojen vähentämiseen, sekä yritysten kiinnostus palvelutason parantamiseen asiakaskohtaisella lopputuotteiden räätälöinnillä.

Tilausohjautuvuus-termi toimii vastinparina varasto-ohjaukselle, mutta viittaa usein ainoastaan lopputuotevaraston hallintaan. Käytännössä tuotannosuunnittelu ja -ohjausjärjestelmissä eri nimikkeitä ja toimintoja hallitaan eri tavoin ja molempia ohjaustapoja käytetään rinnakkain. Tämän vuoksi usein käytetään työn järjestämiseen perustuvaa tarkempaa jaottelua. Todennäköisesti yleisimmin käytetty jaottelu tehdään tuotannon toimintojen perusteella. Tämän kolmiosaisen luokittelun ovat esittäneet muiden muassa Amaro et al. (1999, s.351) seuraavasti:

Tilausohjattu suunnittelu (Engineer-To-Order, ETO)

Tilausohjattua suunnittelua tehdään tuotteille, jotka vastaavat yksilöityyn asiakastarpeeseen, tai vaativat merkittävää asiakaskohtaista räätälöintiä. Tuotteen osat, tuoterakenne ja reititys tuotantojärjestelmässä voivat näin ollen olla yksilöllisiä jokaisella tilauksella.

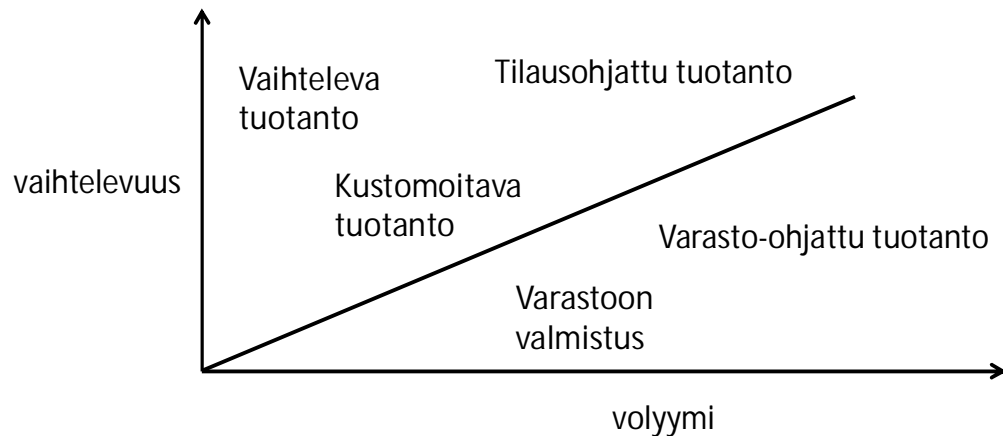
Tilausohjattu kokoonpano (Assemble-To-Order, ATO)

Tilausohjatussa kokoonpanossa lopputuotteet kootaan pääasiassa standardoiduista komponenteista, tai moduuleista. Näitä varioimalla voidaan tuottaa erilaisia tuotteita ja tilaukset voivat myös sisältää asiakaskohtaista kustomointia. Komponentit voidaan valmistaa sisäisesti yrityksessä tai tilata alihankintana, mutta ainakin osaa niistä ohjataan ennusteiden perusteella.

Tilausohjattu valmistus (Make-To-Order, MTO)

Tilausohjatussa valmistuksessa ainakin suurinta osaa tuotteen valmistukseen tarvittavista toiminnoista ohjataan asiakastilausten perusteella. Joissain tapauksissa myös osa materiaaleista ja raaka-aineista voidaan hankkia tilauskohtaisesti. Tilausohjattu valmistus mahdollistaa korkeamman asiakaskohtaisen räätälöintiasteen, kuin tilausohjattu kokoonpano.

Toinen mahdollisuus on luokitella tuotteiden varioituvuuden perusteella. Amaro et al. (1999), sekä Stevenson et al. (2005) ovat esittäneet Kuva 3 mukaisen luokittelun. Tuoterakenteeseen perustuva luokittelu on jossain määrin harhaanjohtava yleistys, koska lopputuotteen varioituvuus ei ole sama asia kuin tuotannon työn vaihtelevuus. Esimerkiksi sulautetuissa järjestelmissä varioituvuus saatetaan toteuttaa pääosin ohjelmistolla, jolloin kokoonpanotyön näkökulmasta tuotteiden vaihtelevuus on pientä.



Kuva 3. Tuotannon luokittelu volyymin ja vaihtelun suhteen, muokattu lähteestä Stevenson et al. (2005)

Tuotteiden varioituvuuden kasvaessa tuotantomäärät yleensä laskevat. Tällöin myös tuotannon skaalaedut menetetään tai niiden ylläpitäminen ainakin hankaloituu. Vaihtelun kasvaessa tuotannon nimikkeiden lukumäärä kasvaa ja osien kiertonopeus heikkenee. Tuottavuuden ja kustannustason näkökulmasta tämä aiheuttaa painetta nimikekohtaisten varastotasojen alentamiseen ja varastoitavien nimikkeiden määrän karsimiseen. Tehokas varastonhallinta puolestaan vaatii, että tiedetään etukäteen mitä valmistetaan ja milloin. Tämä on siis yksi merkittävä motivaatiotekijä tuotannonsuunnittelutoiminnon kehittämiseksi.

Toimitusketjunhallinnan (SCM, Supply Chain Management) yleistyttyä tuotannonsuunnittelun merkitys on kasvanut myös yrityksen toimittajille ja asiakkaille jaettavan informaation vuoksi. Toimitusketjunhallinnalla pyritään parantamaan tehokkuutta lisäämällä ketjun yli tapahtuvaa toimintojen koordinoitua ja ennustamista (Zhou & Benton, 2007). Yksittäisten toimijoiden tuotantoinformaation laatu on silloin keskeisessä osassa prosessin toimivuuden kannalta.

2.4 Suunnittelujärjestelmän vaatimukset tilausohjatussa tuotannossa

Stevenson (2005) argumentoi, että kilpailukykyisen toimitusaikavaateen vuoksi tilausohjatun tuotannon tuotannosuunnittelu- ja ohjausjärjestelmän (engl. PPC, Production Planning and Control) vaatimusmäärittely on haastavampi kuin varasto-ohjatussa tuotannossa. Tilausohjatun tuotannosuunnittelun yleisiksi järjestelmävaatimuksiksi ehdotetaan seuraavaa ominaisuuslistaa:

1. Tarjouspyyntövaiheen sisällyttäminen järjestelmään toimituspäivä- ja kapasiteettisuunnittelua varten
2. Erilliset vaiheet työn kirjaamiselle ja tuotantoon vapauttamiselle, tavoitteena toimituspäivien noudattaminen
3. Mahdollisuus käsitellä kustomoituja tuotteita ja yksittäiskappaleita suunnittelussa
4. Mahdollisuus suunnitella ja ohjata tuotantoa vaihtelevilla reitityksillä
5. Soveltuvuus myös pienille ja keskisuurille yrityksille.

Stevensonin lista ei ole kattava vaatimusmäärittely, vaan sisältää lähinnä yksittäiskappaleen ohjaukseen liittyviä toiminnallisuuksia. Suunnittelujärjestelmässä tämä vaatimus voidaan toteuttaa mallintamalla yksittäiset tilaukset toisistaan riippumattomina projekteina. Luvussa 4 käsitellään tarkemmin projektinhallintanäkökulmaa tuotannosuunnittelussa.

Tuotannon tilausohjautuvuus voi lyhentää suunnittelujärjestelmän aikajännettä, koska ennustamisen osuus vähenee. Myös tässä työssä karkeasuunnittelulla tarkoitetaan lähtökohtaisesti muutaman kuukauden pituisen tuotantosuunnitelman tekemistä, mitä voi seurata töiden allokointi yksittäisille resursseille hienosuunnittelulla. Lyhyempi aikajänne asettaa vaatimuksia myös suunnittelujärjestelmässä käytettävän informaation laadulle. Suunnittelussa käytettävän informaation täytyy olla tarkempaa kuin suunnittelun aikaresoluutio, joten suunnitteluhorisontti, aikaresoluutio ja suunnitteluinformaation laatu vaikuttavat toisiinsa ketjussa.

2.5 Imu- ja työntöohjaus

Goldrattin (1984) 1980-luvulla lanseeraama TOC-johtamismenetelmä (engl. Theory-Of-Constraints) korostaa tuotannonohjauksen näkökulmana imuohjausta. Sen perusajatuksena on pullonkaularesurssin varastopaikoista lähtevä valmistusimpulssi, joka kertoo aiemmille vaiheille, milloin uusi työ pitää aloittaa. Vastaavasti Ohno (1988) pitää imuohjausta JIT-filosofian keskeisenä komponenttina TPS-järjestelmässä (Toyota Production System). Imuohjauksen yleistymisen ja sillä saavutetut hyvät tulokset siirsivät varastonhallintavastuuta tuotannosuunnittelusta ohjaukselle. Samalla alkoi pitkään kestänyt tieteellinen keskustelu imuohjauksen suhteesta MRP/MRPII järjestelmiin (Benton & Shin, 1998).

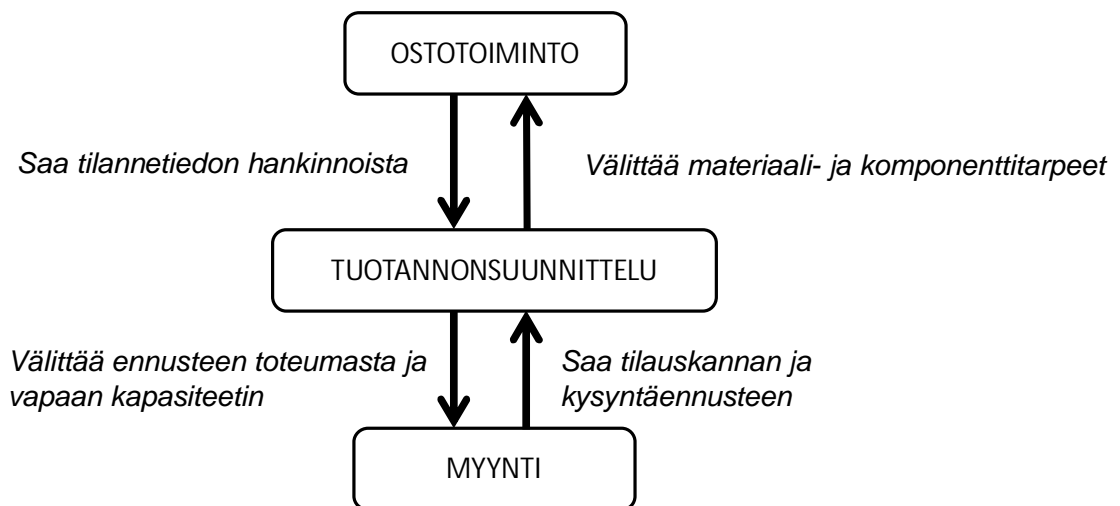
Bentonin & Shinin (1998) mukaan MRP:tä JIT- tai TOC-metodiin vertaileva kirjallisuus keskittyy nimenomaan imu- ja työntöohjauksen eroihin. Työntöohjaus perustuu keskitettyyn suunnitelmaan, kun taas imuohjaus on paikalliseen päätöksentekoon perustuva hajautettu ohjausjärjestelmä. MRP:tä on kehitetty esimerkiksi ottamaan huomioon suunnitteluun liittyvät epävarmuudet (Mula et al., 2006), mutta se on silti työntöohjattu järjestelmä. Nykyisin käytössä on myös hybridijärjestelmiä, joissa MRP on suunnittelujärjestelmä, ja ohjausjärjestelmä perustuu esimerkiksi JIT periaatteille.

Imuohjauksen edut tunnustetaan laajalti ja sitä sovelletaan etenkin tilausohjatussa tuotannossa. On kuitenkin hyvä huomata, että käytännössä tuotantojärjestelmissä esiintyy piirteitä molemmista ohjaustavoista. Esimerkiksi etukäteen määritetyt valmistuksen eräkoot ovat määritelmällisesti työntöohjausta. Spearman & Zazanis (1992) argumentoivat, että imuohjauksen hyödyt johtuvat ennemmin keskeneräisen tuotannon rajoittamisesta kuin prosessin alkua kohti virtaavan informaation suunnasta. Plenert (1999) puolestaan korostaa MRP-järjestelmän etuna joustavuutta tuotevarianttien ja valmistuksen reititysten suhteen.

3. TUOTANNONSUUNNITTELU

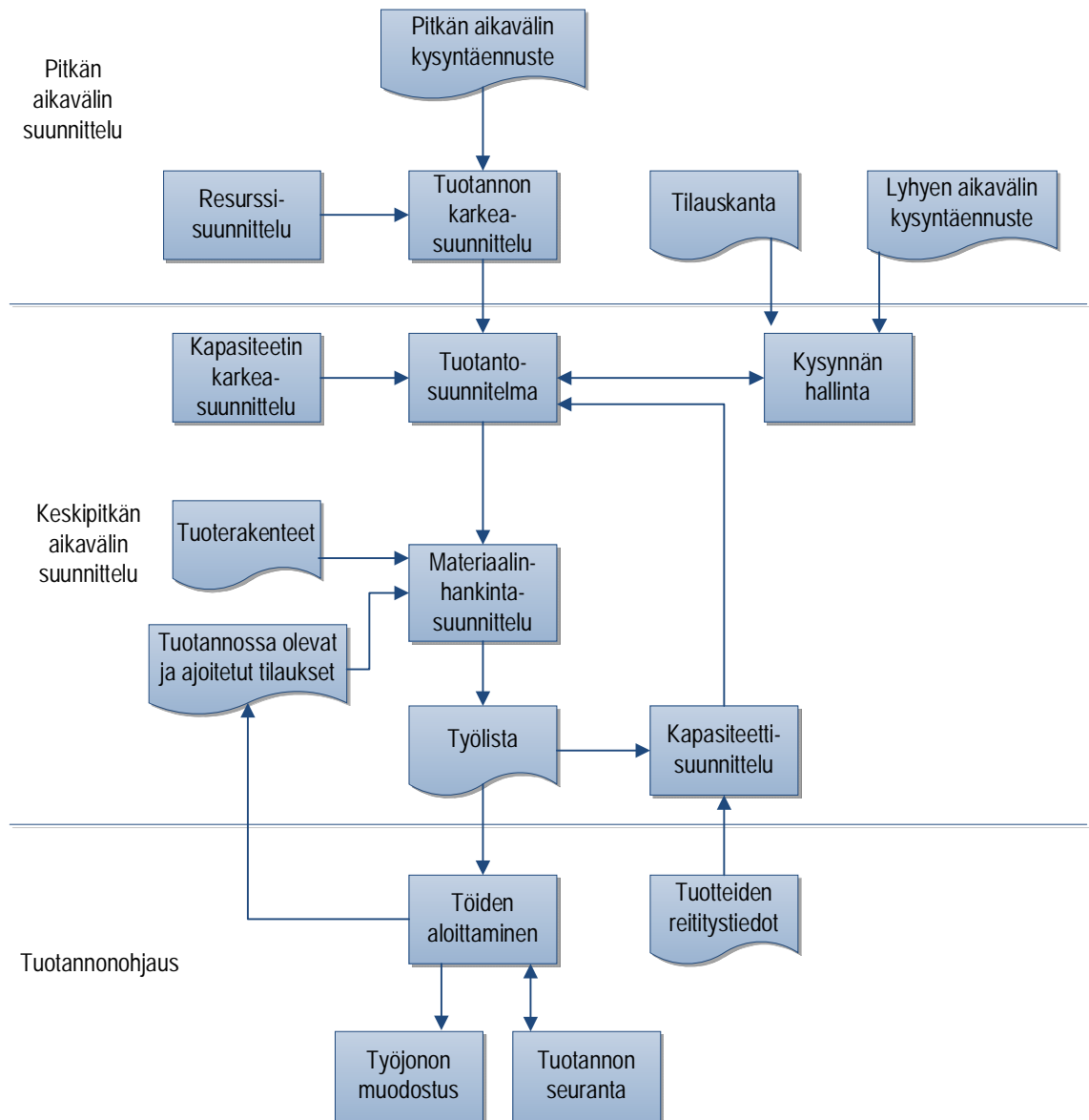
3.1 Suunnitteluprosessi

Tuotannon karkeasuunnittelun tavoitteena on varmistaa, että suunnitelman mukaiseen tuotantomäärään varataan riittävästi resursseja (Chen & Liao, 2003). Se on yrityksen operatiivisen toiminnan kannalta keskeinen toiminto, joka kerää ja välittää informaatiota yrityksen myynnin, ostojen ja tuotannon välillä (Kuva 4). Tuotannonsuunnittelu tuottaa ostotoiminnolle informaation tulevista materiaali- ja komponenttitarpeista. Toiseen suuntaan kulkevaa informaatiota ovat vahvistukset hankinnoista, sekä mahdolliset materiaali puutteet tai poikkeustilanteet. Vastaavasti tuotannonsuunnittelun kommunikointi myynnin kanssa kattaa tuotannon tavoitteet, aikataulut ja ennusteen toteumasta.



Kuva 4. Tuotannonsuunnittelun tietovirrat ostoon ja myyntiin

Hopp & Spearman (2008) esittävät yksityiskohtaisemman tuotannonsuunnittelun skeeman, johon sisältyvät suunnittelutehtävät eri aikajäniteillä, sekä niiden väliset yhteydet. Kuva 5 esitetään tärkeimmät tuotannonsuunnitteluun liittyvät tehtävät ja dokumentit eri aikajäniteillä. Suorakaiteen muotoiset laatikot kuvaavat toimintoja, jotka tuottavat jalostettua informaatiota, ja aaltoviivan sisältävät laatikot kuvaavat dokumentteja. Nuolet osoittavat missä eri suunnittelutehtävien tuottamaa informaatiota tarvitaan.



Kuva 5. Yrityksen suunnittelutoiminnot tuotannossa (muokattu lähteestä Hopp & Spearman, 2008).

Suunnittelujärjestelmä voidaan jakaa ajallisesti kolmeen luokkaan, joista käytetään usein myös nimityksiä strateginen, taktinen ja operatiivinen taso.

Pitkän aikavälin suunnittelu

Pitkän aikavälin suunnittelu perustuu tulevaisuuden markkina- ja kysyntäennusteisiin. Kysyntäennusteen perusteella yritys tekee resurssisuunnitelman ja arvioi tulevaisuuden investointitarpeet. Suunnittelun tavoitteena on luoda käsitys tulevaisuuden tuotantomääristä, sekä vastaavan tuotantomäärän tuottamiseen tarvittavasta

kapasiteetista. Tuotannon karkeasuunnitelman perusteella tehdään päätökset pitkän vaikutusajan kapasiteetti-investoinneista.

Keskipitkän aikavälin suunnittelu

Keskipitkän aikavälin suunnittelussa tavoitteena on varmistaa, että tuotannossa on saatavilla oikeat materiaalit, sekä tilausten toimittamiseen vaadittava kapasiteetti ja työvoima. ERP- (engl. Enterprise Resource Planning) ja MRP-järjestelmien tiedonkäsittelykapasiteettia hyödynnetään erityisesti keskipitkällä aikavälillä. Keskipitkän aikavälin tärkein suunnitteludokumentti on päätuotantosuunnitelma (engl. Master Production Schedule, MPS), minkä perusteella ajoitetaan materiaalihankinnat, suunnitellaan mainoskampanjat, ja tehdään työvuorolistat.

Keskipitkän aikavälin suunnittelu voidaan edelleen jakaa karkea- ja hienosuunnitteluun (Buxey, 2003). Kaksivaiheisessa suunnitteluprosessissa karkeasuunnittelu tuottaa yleisen tason tuotantosuunnitelman eri suunnittelutoiminnoille ja hienosuunnittelu puolestaan toimittaa tarkan aikataulutuksen tuotannonohjaukseen. Suunnittelu ei ole aina kaksivaiheista, vaan tuotantoympäristöstä riippuen erillinen hienosuunnittelu voidaan myös jättää tekemättä. Tällöin tuotannonohjauksella on merkittävämpi rooli käytännön toteutuksen organisoinnissa.

Operatiivinen ohjaus

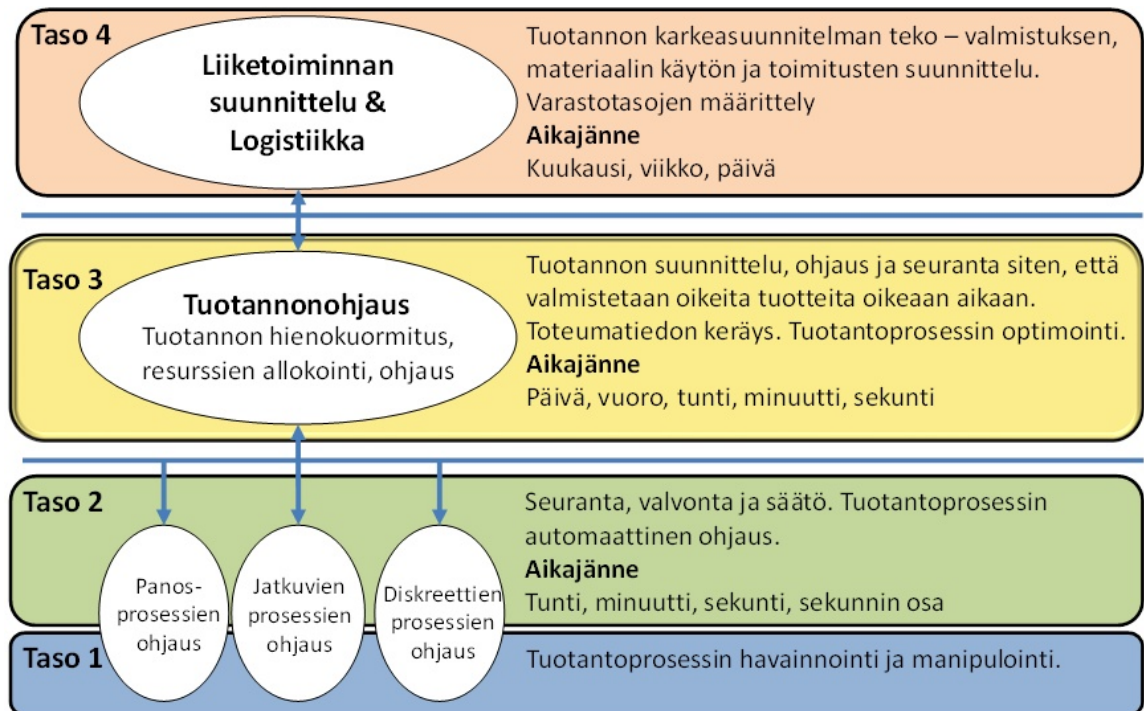
Lyhyen aikavälin suunnittelutehtäviin kuuluu työjonon muodostus, tuotannonseuranta, sekä tilausten toimittaminen. Työjono muodostetaan tuotantosuunnitelmasta johdettavan työlistan perusteella. Tuotannon kustannusrakenteen kannalta merkittävimmät päätökset tehdään pitkällä ja keskipitkällä aikavälillä. Lyhyellä aikajänteellä suunnittelun pääasiallisena tavoitteena on mahdollisimman hyvän toimitusvarmuuden saavuttaminen sekä tähän liittyen reagointi poikkeustilanteisiin.

3.2 Tietojärjestelmät

Tuotannosuunnittelu on pitkään ollut tutkijoita kiinnostava aihe. Sen ongelmat ovat matemaattisesti haastavia ja eri tuotantoympäristöissä on omia erityispiirteitään. Stevenson (2005) toteaa, että yleisen tason suunnittelutyökalut eivät täytä toimialakohtaisia vaatimuksia riittävän hyvin.

ISA-95 standardi määrittelee tuotannosuunnittelun ja -ohjauksen (MES/APS) työkalut tasolle 3 Kuva 6 mukaisesti. Standardi määrittelee tuotannon tietojärjestelmien roolit ja keskinäiset suhteet. Käytännössä tilanne ei ole aina yhtä selkeä, sillä toiminnanohjausjärjestelmillä saatetaan tehdä tuotannon karkeasuunnittelua ja muutkin

tuotannon tietojärjestelmät sisältävät usein toimintoja standardin mukaisen määrittelyn ulkopuolelta.



Kuva 6. ISA-95 standardin määrittelemät yrityksen toiminnalliset hierarkiatasot (muokattu ANSI/ISA-95.00.03-2005 standardista).

Tuotannosuunnittelutyökalut toimitetaan yleensä osana MES-järjestelmää suunnittelun ja ohjauksen kiinteästä yhteydestä johtuen. Nämä ovat yhteydessä toiminnanohjausjärjestelmään, jossa käsitellään ainakin asiakastilaukset ja transaktiodata. MES-järjestelmässä ylläpidetään tuotantodata. Yleensä tuotantodata hallitaan ERP-järjestelmässä, jos erillistä tuotannonohjausjärjestelmää ei ole käytössä.

3.3 Kapasiteetinsuunnittelun perusstrategiat

Kysynnän vaihtelu johtuu useista eri tekijöistä, eikä näiden vaikutusten arvioiminen ole aina helppoa. Kausivaihtelua aiheuttavia tekijöitä ovat esimerkiksi vuodenajat, lomakaudet ja sää. Monille teollisuustuotteille kausiluonteista vaihtelua voi aiheuttaa yritysten investointimäärärahojen ajoittuminen tietyille osalle vuotta. Lisäksi yleinen talouden kehitys ja toimialan kilpailutilanne aiheuttavat kysynnän vaihtelua pidemmällä aikavälillä. Buxey (2003) on analysoinut kausivaihtelua kohtaavilla teollisuudenaloilla käytettäviä tuotantostrategioita 42 yrityksen kyselytutkimuksella.

Tuotantokapasiteetin sovittaminen kysynnän muutoksiin

Tuotantokapasiteettia voidaan säädellä useilla eri keinoilla, joiden edullisuus vaihtelee eri tilanteissa ja tuotantoympäristöissä. Yleisesti ottaen kapasiteetin kasvattaminen on huomattavasti helpompaa ja edullisempaa kuin sen vähentäminen. Tuotannossa käytettävien koneiden jälleenmyyntiarvo ei vastaa niiden arvoa yritykselle, vapautuvaa tilaa ei välttämättä saada hyödynnettyä kustannustehokkaasti, ja irtisanomiset sekä lomauttamiset aiheuttavat ylimääräisiä kustannuksia.

Helpoin tapa lyhytaikaisen ylimääräisen kapasiteettitarpeen täyttämiseen on teettää ylitöitä. Jatkuvasti käytettynä pitkällä aikavälillä ylityöt muodostuvat kalliiksi ja voivat aiheuttaa työtehon heikkenemistä sekä tyytymättömyyttä työntekijöissä. Suurempi kapasiteettilisäys saavutetaan, mikäli käyttöön voidaan ottaa ylimääräinen työvuoro.

Mikäli kohtuullinen lisäkapasiteetin tarve nähdään jatkuvaksi, on järkevämpää palkata lisää työvoimaa. Työvoiman lisääminen aiheuttaa koulutuskustannuksia ja riskin korkeammasta kustannustasosta myös kysynnän vähentyessä. Tämän vuoksi kapasiteettitarve voidaan usein täyttää käyttämällä vuokratyövoimaa.

Edelliset keinot keskittyvät työvoimamuutoksilla tehtävään kapasiteetin lisäämiseen. Alihankinnan lisäämisellä voidaan lisätä myös konekapasiteettia ilman, että tuotantokapasiteettiin sitoutuneen pääoman riski kasvaa. Alihankinnan käyttäminen satunnaisen kapasiteettitarpeen täyttämiseen vaatii perusteellista suunnittelua jo etukäteen, sillä alihankintapäätökset ovat vaikutuksiltaan varsin laajoja ja niissä huomioitavia tekijöitä on useita.

Pitkän aikavälin kapasiteettimuutoksiin liittyvät lähinnä kone-, tuotantolinja- ja tuotantolaitosinvestoinnit. Nämä ovat pitkän vaikutusajan päätöksiä, joita ei tehdä kausivaihtelun tasoittamiseksi, vaan suurempien toimintaympäristön muutosten vuoksi.

Vaihtelun tasaaminen varastojen avulla

Tuotannonohjauksen kannalta helpoin ratkaisu vaihtelevan kysynnän tyydyttämiseen on mitoittaa tuotantotahti pitkän aikavälin kysynnän mukaan ja tasoittaa vaihtelua lopputuotevarastojen avulla. Tämä edellyttää, että tulevaisuuden kysyntä voidaan ennustaa kohtuullisella varmuudella myös yksittäisten tuotteiden osalta. Varastointi aiheuttaa kustannuksia varastoihin sitoutuneen pääoman sekä varastotilan vuokrien kautta. Lisäksi varastot sisältävät epäkuranttiusriskin, mikä realisoituessaan johtaa siihen, että tuotteet joudutaan myymään suunniteltua alhaisemmalla hinnalla tai niitä ei saada myytyä ollenkaan.

Varsinkin tilausohjatussa tuotannossa mahdollisuudet varastointiin ovat rajalliset. Lopputuotteiden varastoinnilla ja joissain tapauksissa etukäteistoimituksilla voidaan tasoittaa tuotantotahtia myös tilausohjatussa tuotantoympäristössä, mikäli toimitusajat ovat riittävän pitkiä suhteessa tuotteiden läpäisy aikaan. Toimitusajoilla kilpailu kuitenkin vähentää tätä pelivaraa.

Kysynnän tasoittaminen tuotevalikoiman avulla

Mikäli tuotteiden kausivaihtelu tunnetaan, voidaan tuotesortimentti pyrkiä valitsemaan niin, että tuotantoyksikön kokonaiskysyntä säilyy tasaisena. Tällöin tarkoituksena on tuottaa tuotteita, joiden kysyntä ajoittuu selvästi eri ajanjaksoille. Edellytyksenä toisiaan subventoivien tuotteiden valinnalle on luonnollisesti, että niitä voidaan tuottaa tehokkaasti olemassa olevalla tuotantokapasiteetilla. Yleinen esimerkki tästä on grillien ja lämmittimien valmistaminen samassa tehtaassa. Kyseiset tuotteet ovat teknisesti varsin samankaltaisia, mutta niiden kysyntä vuodenajan mukaan poikkeaa toisistaan huomattavasti.

Kysyntään vaikuttaminen

Tuotantomääriä ja kysyntää voidaan tasapainottaa keskenään myös vaikuttamalla kysyntään. Esimerkiksi alennuksilla ja markkinointipanostuksilla voidaan pyrkiä siirtämään kysyntää niille ajanjaksoille, jossa se on tavallista alhaisempaa. Tyypillisesti kulutushyödykkeiden hintajousto on suurempi kuin investointihyödykkeiden, joten myös edellytykset kysynnän tasoittamiseen hinnan alennuksilla ovat suurempia kuluttajasektorilla.

MRP/MRPII-järjestelmien perintönä karkeasuunnittelua on pidetty toimintona, jossa pyritään arvaamaan tulevaisuuden tuotantomäärät. Tilausohjatussa järjestelmässä tämä ei kuitenkaan ole keskeinen tavoite, vaan karkeasuunnittelu tulisi nähdä luotettavaan informaatioon perustuvana toimintojen aikatauluttamisena. Myös Buxey (2003) toteaa enemmistön yrityksistä käyttävän kysyntää seuraavia tuotantostrategioita ja tekee tästä päätelmän, että perinteiseen karkeasuunnitteluun suuntautunut tutkimus on heikosti sovellettavissa käytäntöön. Yritykset ovat siis siirtyneet tilausohjattuihin toimintatapoihin, joissa suunnittelun rooli muuttuu.

4. PROJEKTINOHJAUSTEORIAT TUOTANNOSUUNNITTELUSSA

4.1 Ongelman muotoilu

Perinteisissä MRP-järjestelmissä tuotannon kuormituksen arviointiin käytettiin usein kiinteitä tuotemalleja, mutta niihin on sittemmin tuotu useita menetelmiä epävarmuuden hallintaan (Mula, 2005). Kiinteällä tuote- tai kuormitusmallilla tarkoitetaan tässä sitä, että tuotteen valmistamiseksi tehtävien työvaiheiden ajoitus pysyy vakiona suhteessa toisiinsa. Tällöin tuotteiden valmistuksen ajoitus määritellään ainoastaan tuotannon aloitus- tai valmistumishetken perusteella ja resurssitarpeet seuraavat suoraviivaisesta tarvelaskennasta. Käytännössä malli soveltuu parhaiten tuotantolinjan kapasiteettitarkasteluun.

Tilausohjatun tuotannon suunnittelussa voidaan sen sijaan käyttää hyväksi projektinsuunnittelun tutkimusalueella saavutettuja tuloksia. Eräs tutkituimpia projektinsuunnitteluongelmia on kapasiteettirajoitettu projektin ajoitus -ongelma (Resource Constrained Project Scheduling Problem, RCPSp). Muita paljon tutkittuja ongelmia ovat resurssikuormituksen taseus -ongelma (Resource Leveling Problem, RLP), sekä projektin nettonykyarvo -ongelma (Net Present Value, NPV). Projektinohjausnäkökulmassa korostetaan tilausten riippumattomuutta toisistaan, eikä esimerkiksi asetusaikojen minimointi ole silloin tyypillinen tavoite ongelman ratkaisussa.

Edellä mainituista ongelmatyypeistä varsinkin ensimmäistä on sovellettu myös teolliseen tuotantoympäristöön, sillä tuotantoresursseilla ajatellaan usein olevan kiinteä kapasiteettirajoitus. Tällöin tuotantoon tulevat tilaukset kuvataan projekteina, jotka pyritään ajoittamaan kapasiteettirajoitusten puitteissa niin, että myöhästymisiä tulee mahdollisimman vähän. Työvaiheilla on tietyt edeltävyysreunaehdot, jotka määrävää niiden suoritusjärjestyksen. De Reyck ja Herroelen (1998) ovat esittäneet optimiratkaisun löytävän proseduurin sekä yhden että useamman projektin RCPSp-ongelmaan.

Resurssikuormituksen taseus -ongelmassa tavoitteet ovat osittain samoja kuin edellisessä, mutta lähestymistapa on hieman erilainen. Tässä tavoitteena on ajoittaa projektit niin, että ajoitusreunaehdot täyttyvät ja eri resurssien kuormituksen vaihtelu minimoituu. RCPSp -ongelmaan verrattuna tavoite ja reunaehdot ovat siis vaihtaneet paikkaa. RLP -ongelmassa huomioidaan yleensä kuormitusvaihtelu myös kapasiteettirajojen alapuolella, minkä vuoksi se on RCPSp -ongelmaa monipuolisempi. Näiden kahden ongelmaluokan ero ei ole aina ilmiselvä, sillä esimerkiksi Kis (2005) esittää laajennuksen RCPSp -ongelmaan, jossa minimoidaan oman kapasiteetin

ylittävää kapasiteettitarvetta. Tällöin ajoitusreunaehtoja ei rikota, mutta tavoite ei myöskään ole kuormitusvaihtelun tasaaminen.

Nettonykyarvo -ongelmassa tavoitteena on maksimoida projektin nettonykyarvo (Neumann ja Zimmermann, 2000). Nettonykyarvon laskenta perustuu siihen, että työvaiheille tai niiden joukolla määritetään joko positiivinen (tuotto) tai negatiivinen (kustannus) kassavirta. Tämän lisäksi määritellään diskonttaustekijä, minkä perusteella kassavirtojen nykyarvot voidaan laskea. Optimointiproseduurin tavoitteena on ajoittaa työvaiheet ajoitus- ja kapasiteettireunaehtojen puitteissa niin, että positiiviset kassavirrat ajoittuvat mahdollisimman aikaiseksi ja negatiiviset kassavirrat mahdollisimman myöhäiseksi ajankohdaksi. Kassavirtojen nettonykyarvon maksimointi voi olla järkevä tavoite varsinkin suurissa ja pitkäkestoissa projekteissa.

Kappaletavaruotantoon sovellettaessa nettonykyarvo-ongelman positiiviset kassavirrat määräytyvät yleensä valmiiden tuotteiden perusteella. Jatkuva tuotannossa asiakaslähtöiset tai tuotannolliset tavoitteet, kuten toimitusvarmuus ja tasainen kuormitusaste ovat useimmiten tärkeämpiä tavoitteita kuin käyttöpääoman hetkellinen optimointi. Toinen ero yksittäisiin projektitoimituksiin on uusiutumattomien resurssien käsittely. Jatkuva tuotannossa resurssit ovat koneita ja työtunteja, joiden ajatellaan olevan käytettävissä suunnitteluhorisontin jokaisella ajanjaksolla. Kertaluonteiset projektit sen sijaan sisältävät useammin resursseja, jotka ovat käytettävissä ainoastaan tietyillä suunnitteluajanjaksoilla. Tällaisia resursseja kutsutaan uusiutumattomiksi resursseiksi (Brucker et al., 1999).

4.2 Kuormitusvaihtelun tasoittamisongelma (RLP)

Tilausohjatun tuotannon suunnittelussa asiakastilaukset muutetaan tuotannon töiksi, jotka täytyy saada tehdyksi usein tiukan aikarajoituksen puitteissa. Taulukko 1 on kerätty tyypillisiä suunnitteluongelman matemaattisessa muotoilussa käytettäviä reunaehtoja ja muuttujia sekä suunnittelun tavoitteita. Listaus perustuu pääosin kohdassa 4.1. viitattuihin malleihin (De Reyck & Herroelen (1998), Neumann & Zimmermann (2000), sekä Kís (2005)), mutta sisältää myös muita kirjallisuudessa esiintyviä tekijöitä. Reunaehtojen ja muuttujien välinen jako ei ole selkeä, sillä sama tekijä voidaan tilanteesta riippuen esittää joko reunaehtona tai muuttujana.

Taulukko 1. Tyypillisiä tuotannosuunnitteluongelman osatekijöitä

Reunaehdot	Muuttujat	Tavoite
Kapasiteettirajoitteet	Resurssien saatavuus tietyllä ajanhetkellä	Toimitusvarmuuden maksimointi
Aikaisimmat aloitusajat	Työmäärä tietyllä ajanhetkellä	Palvelukyvyyn maksimointi
Toimitusajat	Työnvaiheiden aloitushetket	Toimitusaikojen keskiarvon minimointi
Työnvaiheiden järjestys	Työjonon järjestys	Toimitusmäärän maksimointi
Töiden maksimikesto	Eräkoot	Kapasiteetin käyttöasteen maksimointi
Komponenttien toimitusajat	Kuormitusintensiteetti	Kapasiteetin käyttöasteen vakiointi
Varastopaikat	Vuorojärjestelyt	Kapasiteettimuutosten minimointi
Uusiutumattomat resurssit	Tilaukoot	Tuotantokustannusten minimointi
Tilarajoitteet	Investoinnit	Läpäisyajan minimointi
Investointi- ja pääomarajoitukset		Läpäisyajan vakiointi
Työjonon sekvenssiehdot		Työn keskeytyksien minimointi
Prosessikohtaiset erityisehdot		Asetusaikojen minimointi
		Tuotannon arvon maksimointi
		Osapuutteiden minimointi

Taulukko 1 listaus ei ole kattava, vaan se esittää suunnitteluongelman muotoiluun liittyvien valintatilanteiden monimuotoisuutta yleisellä tasolla. Mallinnuksessa on otettava kantaa siihen, mitkä tekijät ovat tärkeimpiä ja millä tarkkuudella suunnittelua kannattaa tehdä. Muuttujiksi tulisi valita tekijöitä, joihin voidaan aidosti vaikuttaa suunnitteluhorisontin aikana, ja pidemmän päätöksentekoprosessin vaativat tekijät kannattaa mallintaa reunaehtoina.

Kapasiteettirajoite

Kuormituksen tasaamiseen pyrkivissä ajoitusmenetelmissä on selkeä jako kapasiteettirajoitteisiin ja -rajoittamattomiin menetelmiin. Kummankin näkökulman puolesta löytyy loogisia perusteluita. Kapasiteettirajoitteisten menetelmien puolesta puhuu kapasiteetin sovittamisen jäykkyys. Esimerkiksi koneinvestointeja ei ole mahdollista tehdä hetkellisen ylikuormitustilanteen perusteella. Toisaalta monissa tilanteissa kapasiteetin kasvattaminen voi tapahtua varsin joustavasti. Yleisiä tapoja tähän ovat vuokratyövoiman ja alihankinnan lisääminen, ylityöt, sekä ylimääräisten työvuorojen käyttöönotto.

Keskeinen ero näiden kahden lähestymistavan välillä havaitaan tilanteessa, jossa järjestelmän kapasiteetti ei ole riittävä tuottamaan tavoiteltua tuotantomäärää aikarajoitusten puitteissa. Kapasiteettirajoitteiset menetelmät rikkovat ajoitusreunaehtoja joko aikaistamalla tai myöhästyttämällä tiettyjä vaiheita tai tilauksia. Rajoittamattomissa menetelmissä puolestaan hyväksytään nimelliskapasiteetin ylittäminen, mutta työvaiheiden myöhästymisiä tai aikaistamisia ei sallita.

Kapasiteettirajoitusten merkitys riippuu siis pitkälti tuotantojärjestelmän joustavuudesta ja tuotantofilosofiasta. Mikäli kapasiteetin sovittamiseen on mahdollisuuksia, ei kapasiteettirajoitusten tiukka määrittäminen ole välttämättä järkevää. Tämä korostuu etenkin tilausohjatussa tuotannossa, jossa toimitusvarmuus on usein ensisijaisen tärkeä kilpailutekijä. Tilausohjatussa tuotannossa resurssikuormituksen vaihtelu on usein melko suurta. Tämä on johtanut siihen, että kapasiteetin joustavuutta on usein korostettu tuotantojärjestelmän suunnittelussa.

Tavoitteen valinta

Tilauskohtaisesti tilattavat komponentit asettavat usein rajoituksia tuotannon aikaistamiselle. Nämä ovat usein arvokkaita ja vaativat pitkän toimitusajan. Tämä asettaa tuotannon aloittamiselle rajoituksia, joita ei ole mahdollista rikkoa edes kustannuksia kasvattamalla. Toimitusvarmuuden korostuminen kilpailutekijänä on kuitenkin ehkä kaikkein tärkein yksittäinen syy kapasiteettirajoittamattoman menetelmän käyttämiseen tuotannosuunnittelussa. Tilausohjatussa tuotannossa toimitusvarmuus on usein tuotannon näkökulmasta kustannuksia tärkeämpi tekijä, eikä suunniteltuja myöhästymisiä voida hyväksyä.

Resurssikuormituksen vakiointi tai sitä välillisesti kuvaavat tavoitteet soveltuvat tuotannosuunnittelun päätavoitteeksi kahdesta syystä. Tasainen kuormitus helpottaa tehtaan kapasiteettisuunnittelua, koska järjestelmän kokonaiskapasiteettia rajoittavat tekijät on helppo tunnistaa. Tämä alentaa pitkällä aikavälillä valmistuksen kustannuksia. Operatiivisesta näkökulmasta tasainen kuormitus tarkoittaa Littlen lain (Little, 1961)

perusteella myös ennustettavaa läpäisyaikaa. Keskenräisen tuotannon ja samalla läpäisyajan vakiointiin tietyllä tuotantotahdilla perustuvat myös ConWIP (engl. Constant WIP) ohjausmenetelmä (Hopp & Spearman, 2008), sekä ainakin osittain myös imuohjausmenetelmät.

Matemaattisesta näkökulmasta on myös huomioitava, että tavoite on vähemmän tärkeä kuin reunaehto. Ajoitusreunaehdoilla rajoitettu kuormituksen tasausongelma sisältää siis erittäin asiakasorientoituneen ongelmanmuotoilun, vaikka sen nimestä voisi päättyä toiseenkin johtopäätökseen.

Suunnitteluongelman yksinkertaistaminen

Joissain tilanteissa tuotantojärjestelmällä on luonnollinen pullonkaula, joka määrittää järjestelmän kokonaiskapasiteetin. Tyypillisesti esimerkiksi lämpökäsittely voi olla tällainen prosessi. Uunin kapasiteettia ei voi käyttötuntien maksimoinnin lisäksi kasvattaa muuten kuin investoimalla toiseen uuniin. Tällöin tuotannonsuunnittelussa voidaan lähteä siitä, että kannattavuusnäkökulmasta tärkeintä on uunin käyttöasteen maksimointi ja kaikki muut tavoitteet ovat sekundäärisiä. Tästä lähtökohdasta tehtävää suunnittelua kutsutaan pullonkaulaohjaukseksi.

Pullonkaulaohjausta käytetään usein, vaikka järjestelmässä ei olisi luonnollista rajoitetta. Syynä tähän on yksinkertaisesti se, että yksittäisen kuormituspisteen tuotannonsuunnittelu on helppoa. Suunnittelun yksinkertaistaminen tässä tilanteessa on kuitenkin haitallista, koska silloin lähtökohtaisesti hyväksytään kapasiteetin käyttöasteen aleneminen pullonkaulaa edeltävissä resursseissa. Tyypillinen esimerkki pullonkaulaohjauksesta on Goldrattin (1988) DBR-ohjaustapa (engl. drum-buffer-rope).

4.3 Lineaarisen optimointiongelman ratkaiseminen

Erityyppisten matemaattisten ongelmien ratkaisuun on kehitetty useita optimointimenetelmiä, joista jotkut tuottavat riittävässä ajassa aina optimiratkaisun, ja toiset pyrkivät tuottamaan nopeasti hyvän ratkaisun. Tässä luvussa esitellään menetelmiä, joita voisi soveltaa RLP-optimointiongelmiin ratkaisuun. Optimointimenetelmien tehokkuutta arvioidaan matemaattisesti erittäin vaikeiden ongelmien ratkaisuaajan perusteella. Seuraavassa on esitelty joidenkin hyvin tunnettujen menetelmien peruseriaatteet, mutta huomattavasti kattavamman yleiskatsauksen esittävät esimerkiksi Harjunkoski et al. (2014).

Eksakti optimiratkaisu

Branch and Bound

Branch-and-bound metodia voidaan soveltaa tilanteissa, joissa optimointiongelma sisältää diskreettejä muuttujia. Perusajatuksena on jakaa ratkaistava ongelma osaongelmiin, joista suljetaan pois niitä ratkaisujen joukkoja, joiden tiedetään sisältävän vain jo löydettyä ratkaisua huonompia vaihtoehtoja. Ratkaisujen osajoukkoja tutkitaan muuttamalla diskreettejä muuttujia jatkuva-arvoisiksi, jolloin osaongelma voidaan muotoilla lineaarisena optimointiongelmana. LP-ongelmien ratkaisuun on olemassa tehokkaita algoritmeja, kuten usein käytetty optimiratkaisun tuottava SIMPLEX algoritmi (esimerkiksi Nelder & Mead, 1965). Osaongelman ratkaisuna saadaan ääriarvo, josta tiedetään, että tutkitusta osajoukosta ei löydy parempaa ratkaisua. Tämän jälkeen osajoukko voidaan jakaa uudelleen pienempiin osiin ja toistaa sama prosessi, tai sulkea kokonaan pois tarkasteltavien joukkojen listalta.

Tämän metodin rinnalle on kehitetty myös hybridimenetelmä Branch-and-cut, jossa ongelman pilkkomiseen käytetään lineaariepäyhtälöitä (ns. leikkaustasoja, engl. cutting plane). Menetelmä soveltuu erityisesti MILP-ongelmiin (engl. mixed-integer linear programming), joissa on sekä jatkuvia että diskreettejä muuttujia.

Metaheuristiset menetelmät

Metaheuristiikalla tarkoitetaan ylätasoa prosessia, joka ohjaa yksityiskohtaisen heuristiikan toimintaa matemaattisen ongelman ratkaisussa (Parejo et al., 2012). Usein näiden strategiat jäljittelevät luonnonilmiöitä tai eläinten käyttäytymistä. Metaheuristisia menetelmiä käytetään usein, kun ratkaistava ongelma on kokoluokaltaan suuri ja laskennallisesti vaikea. Menetelmät eivät varmista globaalin optimiratkaisun löytämistä (Kallrath, 2002), mutta niiden tavoitteena onkin löytää ”riittävän hyvä” ratkaisu kohtuullisessa ajassa.

Eri menetelmien tehokkuus riippuu paljolti ratkaistavan ongelman luonteesta, eikä yksiselitteisesti voida sanoa, että jokin olisi aina toista parempi (Yang, 2013). Usein pyritään myös hyödyntämään eri menetelmien hyviä puolia soveltamalla eri menetelmiä eri osaongelmiin. Esimerkiksi Ganesh ja Punniyamoorthy (2005) esittelevät geneettisen algoritmin ja simuloidun jäähtymisen yhdistävän hybridimenetelmän jatkuva-aikaisen tuotannonsuunnitteluongelman ratkaisuun. Tässä geneettinen algoritmi toimii koko ratkaisualuetta luotaavana menetelmänä, ja simuloidulla jäähtymisellä etsitään lokaalia optimia rajatulta ratkaisualueelta. Esiteltyjä menetelmiä voidaan käyttää myös hierarkkisen rakenteena, jossa ratkaistaan ensin yleisen tason ongelma, ja sitten yksityiskohtaisempi tai toissijaisen tavoitteen sisältävä ongelma rajatulla alueella.

Tabu Search

Tabu search -menetelmässä käytetään paikallista heuristiikkaa etsimään valitun ratkaisun lähiympäristöstä parempia ratkaisuja, kunnes uutta ratkaisua ei enää löydetä tai muu lopetusehto täyttyy. Tämän jälkeen valitaan toinen alue, jonka lähiympäristöä aletaan jälleen tutkia (Glover, 1990). Menetelmän nimitys tulee listasta, jossa ylläpidetään ratkaisuja joita ei haluta enää tutkia. Nämä voivat olla esimerkiksi jo tarkastettuja ratkaisuja tai alueita. Tabu-listan käytön vuoksi tämä menetelmä soveltuu ainoastaan ongelmiin, joissa algoritmia ohjaavat muuttujat ovat diskreettejä. Jatkuva-arvoisilla muuttujilla on äärettömän monta ratkaisua, joten jo tarkastettujen ratkaisujen listaaminen olisi hyödytöntä.

Tabu search -menetelmässä voidaan käyttää useamman tason ehtoja, joilla ratkaisun etsimistä ohjataan (Glover, 1990). Korkean tason ehdoilla pyritään varmistamaan, että ratkaisuvaihto käydään riittävän kattavasti läpi. Keskitason ehdoilla pyritään valitsemaan alue, josta on todennäköistä löytää aiempaa parempi ratkaisu. Alimman tason ehtona ylläpidetään listaa tarkastetuista ratkaisuksista, jotta samaa pistettä ei lasketa kahdesti. Tabu search -menetelmä soveltuu hyvin myös monitavoiteoptimointiin, koska siinä voidaan tarkastella ratkaisujoukkoja yksittäisten ratkaisujen sijaan (Baykasoglu, 2001).

Geneettiset algoritmit

Geneettiset algoritmit jäljittelevät biologista evoluutiota optimiratkaisun etsimisessä. Algoritmien keskeisiä termejä ovat populaatio, periytyminen, risteytys ja mutaatio. Geneettisissä algoritmissa ratkaisu kuvataan kromosomina, joka koostuu geeneistä. Geeni puolestaan on vakiopituinen bittijono. Yksiselitteisellä kuvaustavalla ratkaisun osista saadaan risteytysprosessissa vaihtokelpoisia. (Man et al., 1996)

Ensimmäinen populaatio tuotetaan yleensä satunnaisesti. Tämän jälkeen algoritmi arvioi populaation ratkaisujen tai niistä valitun osajoukon hyvyyttä, ja muodostaa pareja, jotka

tuottavat seuraavan populaation. Hyvillä ratkaisuilla on suurempi todennäköisyys valikoitua tähän joukkoon, kuin huonoilla. Uusi ratkaisu tuotetaan ”risteyttämällä”, eli valikoimalla jälkeläiseen osia kahdesta edellisen sukupolven ratkaisusta. Tämän lisäksi prosessiin voi kuulua mutaatio, jossa uuden ratkaisun jotain osaa muutetaan satunnaisesti. Mutaatio vähentää löydettävän ratkaisun riippuvuutta alkuperäisestä peruspopulaatiosta, sekä parantaa edellytyksiä olla hakeutumatta paikalliseen optimiratkaisuun. Algoritmi jatkaa uusien populaatioiden tuottamista samalla periaatteella, kunnes lopetusehto täyttyy ja paras löydetty ratkaisu valitaan ongelman ratkaisuksi. (Man et al., 1996)

Simuloitu jäähditys

Simuloitu jäähditys (engl. Simulated Annealing, SA) lainaa analogiansa fysiikasta, ja jäljittelee esimerkiksi metallin tai nesteen jäähtymistä (Kirkpatrick et al., 1983). Esimerkiksi minimointiongelmassa systeemin kaikki mahdolliset tilat kuvaavat ratkaisujen joukkoa, ja tavoitefunktion arvo kuvaa järjestelmän lämpötilaa. Tavoitteena on löytää optimiratkaisua kuvaava energiaminimi.

Jäähtyminen on stokastinen prosessi, jossa järjestelmän seuraavaa tilaa ei voida tietää varmasti edellisen tilan perusteella. Lisäksi prosessin luonne riippuu kappaleen lämpötilasta niin, että kuumassa lämpötilassa tilamuutokset ovat suurempia tai nopeampia kuin jäähtyneessä kappaleessa. Liian nopeassa jäähditysprosessissa kappaleen kiderakenteeseen jää virheitä, jotka heikentävät sen rakennetta. Energiaminimin saavuttaminen vaatii, että järjestelmän annetaan hakeutua termiseen tasapainotilaan aina ennen lämpötilan alentamista. (Kirkpatrick et al., 1983)

Optimointimenetelmä jäljittelee näitä ominaisuuksia valitsemalla ympäristömuuttujaksi lämpötilan, mitä lasketaan optimoinnin edistyessä. Prosessin alussa algoritmi kulkee laajasti mahdollisten ratkaisujen alueella, eli todennäköisyys hyväksyä uusi ratkaisu järjestelmän tilaksi on suuri. Algoritmin edetessä todennäköisyys siirtyä aiempaa tilaa huonompaan ratkaisuun laskee, mikä johtaa myös pienempiin yksittäisiin muutoksiin tilojen välillä. Lämpötilan ollessa nolla etsitään ahneelle algoritmilta tyypillisesti ainoastaan aiempaa parempia ratkaisuja. Menetelmä yhdistää ongelman pilkkomisen ja iteratiivisen parantamisen lähestymistavat (Kirkpatrick et al., 1983).

Viimeaikainen kehitys

2000-luvulla erilaisten metaheuristiikkojen kehitys on jatkunut nopealla tahdilla ja viime aikoina erityisesti ns. parviäly eli eläinjoukkojen käyttäytymisen jäljittely on ollut yleinen lähestymistapa. Uusien luontoaiheisten algoritmien esikuvina ja nimien antajana ovat saaneet toimia esimerkiksi mehiläiset, muurahaiset, lepakot, tulikärpäset ja käkien loisiminen (Yang, 2013). Geem & al. (2001) julkaisivat muusikon improvisointia

matkivan harmoninen haku -algoritimin (engl. harmony search, HS), jonka tehokkuuden arviointiin on sovellettu esimerkiksi algoritmin Sudokun ratkaisukykyä (Geem, 2007). Pääpiirteiltään kyseinen menetelmä muistuttaa geneettisiä algoritmeja.

Metaheuristisia menetelmiä pidetään tehokkaina monien erityyppisten ongelmien ratkaisemiseen, mutta niiden keskinäinen vertailu on hankalaa. Wolpert ja Macready (1997) esittävät, että ilmaisia lounaita ei ole ja yleisen tason algoritmien tehokkuuserot riippuvat ainoastaan testauksessa käytetystä ongelmatyypistä. Teollisten sovellusten kannalta tämä ei ole merkittävä ongelma, koska yleensä ratkaistavan ongelman tyyppi ja muotoilu tunnetaan ratkaisumenetelmää valitessa. Esimerkiksi Paraskevopoulos et al. (2016) ovat testanneet AMP-luokan (engl. Adaptive Memory Programming) metodeja RCPSP-ongelman ratkaisuun ja saavuttaneet hyviä tuloksia. AMP-luokkaan kuuluvat muun muassa Tabu Search ja Ant Colony algoritmi sekä geneettiset algoritmit.

5. TUOTANNON KARKEASUUNNITELUN OPTIMOINTIONGELMA

5.1 Jatkuvasti jakautuneet resurssit

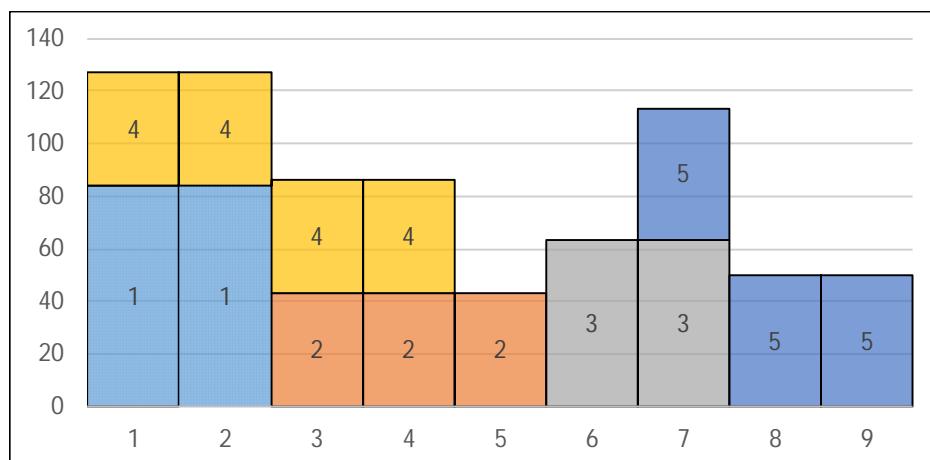
Tässä työssä esitettävä ajoitusmenetelmä perustuu oletukseen jatkuvasti jakautuvista resursseista. Tämä tarkoittaa sitä, että yksittäistä työvaihetta voidaan tehdä vaihtelevalla intensiteetillä eri ajanhetkillä. Ominaisuus on tyypillinen etenkin henkilöresursseille, sillä työntekijät voivat usein vaihtaa työtä ilman pitkiä asetusajoja.

Työvaiheiden kuormitusintensiteetin vaihtelu mahdollistaa kuormituksen tasaamisen työvaiheiden ajoitusta muuttamatta. Kuva 7 on esitetty joukko yksittäistä resurssia kuormittavia työvaiheita. Vaakasuorat palkit kuvaavat työvaiheen ajoitusta, ja niiden sisällä olevat luvut kuvaavat kunkin työvaiheen työkuormaa.

Työ	Ajanhetki								
	1	2	3	4	5	6	7	8	9
1	168								
2			130						
3						127			
4	172								
5							150		

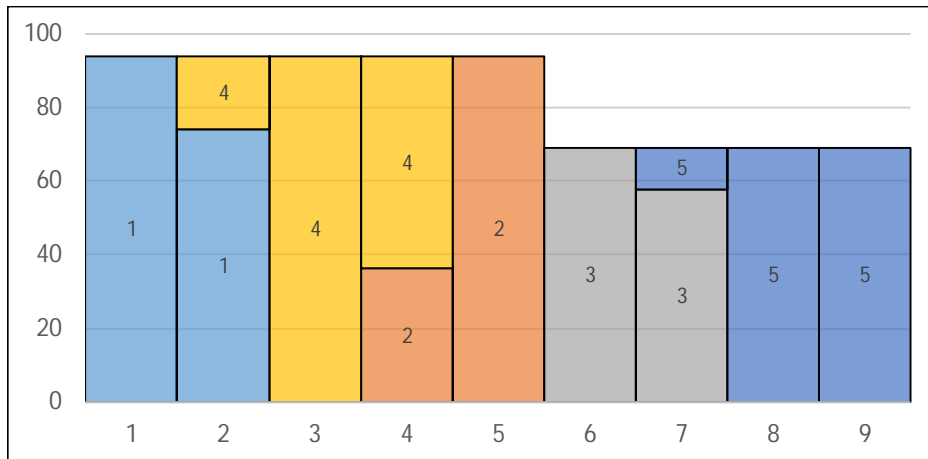
Kuva 7. Yksittäiselle työpisteelle ajoitettujen töiden aikataulu.

Kuva 8 on esitetty edeltävää aikataulua vastaava resurssin kuormitusprofiili, kun kuormitusintensiteetti pidetään vakiona. Numerot palkkien sisällä vastaavat Kuva 7 työvaiheiden indeksejä. Esitetyssä profiilissa resurssin kuormitushuippu yhdellä ajanhetkellä on 127 yksikköä.



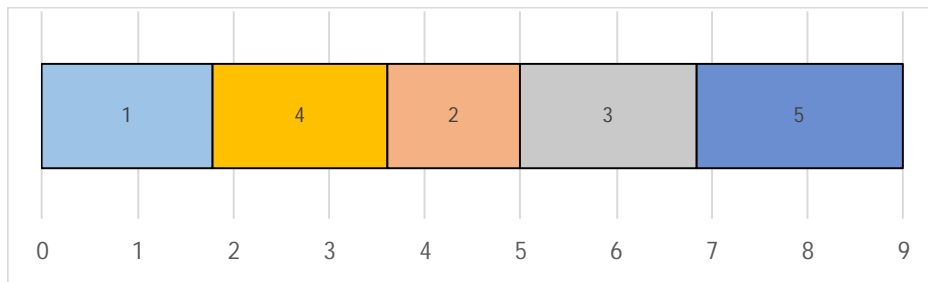
Kuva 8. Aikataulua vastaava kuormitusprofiili vakiointensiteetillä.

Mikäli työnvaiheiden kuormitusintensiteetin vaihtelu eri ajanhetkillä sallitaan, voidaan sama työmäärä ajoittaa huomattavasti tasaisemmin. Tämä on esitetty Kuva 9. Kunkin työnvaiheen ajoitus ja työmäärä on pidetty ennallaan, mutta resurssin kuormitushuippu on ainoastaan 94 yksikköä. Yhdenkään työnvaiheen ajoitusreunaehtoja tai kokonaiskuormaa ei ole muutettu.



Kuva 9. Resurssin kuormitusprofiili vaihtelevalla kuormitusintensiteetillä

Edeltävässä esimerkissä aika on diskreetti ja työnvaiheen kuormitus eri ajanhetkillä on jatkuva-arvoinen muuttuja. Tilanne muuttuu, mikäli myös aika asetetaan jatkuva-arvoiseksi muuttujaksi. Tällöin vastaava hienosuunnitelma voidaan esittää Kuva 10 mukaisesti.



Kuva 10. Töiden lopullinen ajoitus hienosuunnitelmassa

Oleellista esimerkissä on, että jokainen työ voidaan tehdä yhtäjaksoisesti tasaisella kuormitusintensiteetillä, vaikka karkeasuunnittelussa kuormitusintensiteetin vaihtelu sallittiin. Tämä pätee aina yksittäiselle koneresurssille, mikäli yksikään työ ei ole ajoitettu niin, että se alkaa myöhemmin ja päättyy aikaisemmin kuin jokin toinen samaa resurssia käyttävä työ (Niemi et al., 2006). Tulos mahdollistaa vaihtelevan kuormitusintensiteetin käyttämisen suunnittelumallissa, vaikka kuormitettavat resurssit olisivat tasaisella kuormitusintensiteetillä käytettäviä koneita. Useamman koneen muodostamille resursseille kyseinen muunnos hienosuunnitelmaksi ei välttämättä ole

aina ajoitusreunaehtoien puitteissa mahdollinen. Tätä tilannetta on tarkasteltu artikkelissa Niemi et al., (2006).

5.2 Tavoitefunktion arvojen laskeminen

Seuraavassa esitetään esimerkin avulla, kuinka tavoitefunktioiden arvot lasketaan ohjelmassa. Tarkastellaan kolmea työvaihetta, jotka kuormittavat samaa resurssia. Työvaiheiden valmistusajat ja työkuormat on esitetty Ganttin kaaviona Kuva 11.

Työvaihe	Ajanhetki										
	1	2	3	4	5	6	7	8	9	10	
1		75									
2		90									
3							50				

Kuva 11. Esimerkki kolmen työvaiheen suunnitelluista valmistusajoista ja työkuormista

Kuva 11 mukainen kuormitustilanne voidaan esittää Taulukko 2 mukaisesti. Taulukkoon tallennetaan kullakin ajanhetkiparilla $\{t_1, t_2\}$ alkavien ja päättyvien vaiheiden työkuormien summa.

Taulukko 2. Työvaiheiden työkuormat eri ajanhetkillä
Työvaiheen päättymishetki t_2

		1	2	3	4	5	6	7	8	9	10
Työvaiheen alkamishetki t_1	1										
	2				75	90					
	3										
	4										
	5										
	6										
	7										
	8										
	9									50	
	10										

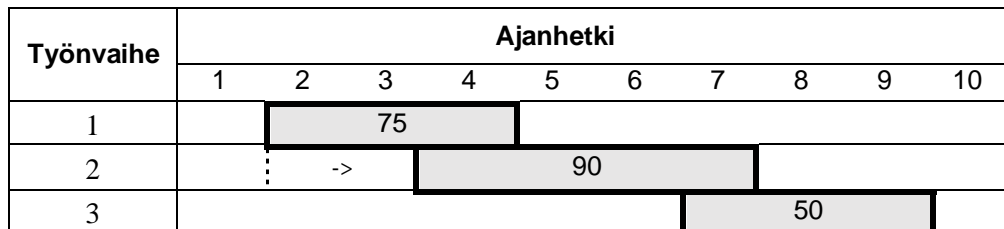
Seuraavassa vaiheessa lasketaan jokaisen ajanhetkiparin $\{t_1, t_2\}$ välisenä aikana alkavien ja päättyvien työvaiheiden kuormien summa jaettuna aikavälin pituudella. Kuva 11 ajoitusta vastaavat arvot on esitetty Taulukko 3. Yksittäinen Taulukko 3 arvo vastaa pienintä mahdollista kyseiselle aikavälille sijoitettavaa kuormaa. Tämän taulukon suurin arvo kuvaa siis yksittäisen resurssin pienintä mahdollista kuormahuippua.

Taulukko 3. Eri aikavälien minimikuormitus

		Ajanhetki t_2									
		1	2	3	4	5	6	7	8	9	10
Ajanhetki t_1	1	0	0	0	18,8	33	27,5	23,6	20,6	23,9	21,5
	2		0	0	25	41,3	33	27,5	23,6	26,9	23,9
	3			0	0	0	0	0	0	7,1	6,3
	4				0	0	0	0	0	8,3	7,1
	5					0	0	0	0	10	8,3
	6						0	0	0	12,5	10
	7							0	0	16,7	12,5
	8								0	0	0
	9									0	0
	10										0

Taulukossa on korostettu solu, jossa on suurin arvo. Resurssin pienin mahdollinen kuormitushuippu on siis 41,3 yksikköä aikavälillä {2,5}.

Seuraavaksi esimerkin työvaihe 2 on siirretty alkamaan vasta ajanhetkellä neljä (Kuva 12). Vaiheen siirtämisen jälkeen koko taulukkoa ei tarvitse laskea uudelleen, vaan muutokset voidaan päivittää ainoastaan niihin soluihin, joihin vaiheen siirtäminen vaikuttaa. Ensimmäisen taulukon päivittäminen on suoraviivaista, mutta vaiheen siirtyminen aiheuttaa enemmän muutoksia Taulukko 3.



Kuva 12. Työvaiheiden ajoitus, jossa toinen vaihe on siirretty alkamaan ajanhetkellä neljä.

Siirron aiheuttamat muutokset keskimääräisiin työkuormiin eri aikaväleillä on esitetty harmaalla pohjalla Taulukko 4. Voidaan havaita, että yksittäisen vaiheen siirtäminen vaikuttaa varsin rajalliselle alueelle etenkin suuremmissa taulukoissa. Tämä on keskeinen asia laskentanopeuden kannalta, sillä algoritmin suoritusaikaa saadaan merkittävästi vähennettyä päivittämällä ainoastaan muuttuvat solut koko taulukon uudelleenlaskemisen sijaan.

Taulukko 4. Ajoitusmuutoksen aiheuttamat kuormitusmuutokset

		Ajanhetki t ₂									
		1	2	3	4	5	6	7	8	9	10
Ajanhetki t ₁	1	0	0	0	18,8	15	12,5	23,6	20,6	23,9	21,5
	2		0	0	25	18,8	15	27,5	23,6	26,9	23,9
	3			0	0	0	0	18	15	20	17,5
	4				0	0	0	22,5	18	23,3	20
	5					0	0	0	0	10	8,3
	6						0	0	0	12,5	10
	7							0	0	16,7	12,5
	8								0	0	0
	9									0	0
	10										0

Esimerkissä on tilanne, jossa vaiheen ajoitusta siirrettiin myöhäisemmäksi. Tällöin kuormitushuippu voi pienentyä niillä aikaväleillä, joissa t_1 on pienempi tai yhtä suuri kuin vaiheen aikaisempi aloitushetki ($t_1 < 3$) ja t_2 on pienempi kuin uusi valmistumishetki ($t_2 < 7$). Vastaavasti kuormitushuippu voi kasvaa aikaväleillä, joissa t_1 on suurempi kuin aikaisempi aloitushetki ($t_1 > 2$) ja t_2 on suurempi tai yhtä suuri kuin uusi valmistumishetki ($t_2 > 6$). Mikäli vaihe ajoitettaisiin aiempaa aikaisemmaksi, tilanne olisi luonnollisesti päinvastainen. Esimerkissä laskettu kuormitushuipun minimi vastaa seuraavassa kohdassa 5.3. esitettävän matemaattisen muotoilun yhtälöä (1).

Kuormituksen tasaaminen huipun ulkopuolella

Kuormitushuipun minimointi ei riitä takaamaan tasaista kuormitusprofiilia, vaan resurssikuormituksen tasoittamiseen kuormitushuipun ulkopuolella tarvitaan neliöllistä kustannusfunktiota. Neliösummaa voidaan käyttää kuvaamaan vaihtelua aikasarjoissa, koska se on läheistä sukua varianssille, mutta hiukan sitä nopeampi laskea. Taulukko 5 on laskettu kullekin ajanhetkelle sijoitettavissa olevan kuormituksen maksimiarvo. Minimoimalla tämän neliösummaa koko ajoitushorisontissa, voidaan kuormituksen tasauksen edellytyksiä parantaa ilman, että lasketaan todellista yksittäiselle ajanhetkelle ajoitettavaa työkuormaa.

Taulukko 5. Esimerkin aikataulujen maksimikuormituksen neliö**Kuva 11 mukainen ajoitus:**

Ajanhetki	1	2	3	4	5	6	7	8	9	10	
Resurssin maksimikuormitus	0	165	165	165	90	0	50	50	50	0	Yhteensä
Maksimikuormi- tuksen neliö	0	27225	27225	27225	8100	0	2500	2500	2500	0	97275

Kuva 12 mukainen ajoitus:

Ajanhetki	1	2	3	4	5	6	7	8	9	10	
Resurssin maksimikuormitus	0	75	75	165	90	90	140	50	50	0	Yhteensä
Maksimikuormi- tuksen neliö	0	5625	5625	27225	8100	8100	19600	2500	2500	0	79275

Neliösumman luonteeseen kuuluu, että sen arvo minimoituu, kun yksittäiset arvopisteet lähestyvät keskiarvoa. Kohdan 5.3. matemaattisessa muotoilussa yhtälö (4) kuvaa maksimikuormituksen neliösumman minimointia.

Esimerkin mukaisessa siirrossa siis myös tämän kustannusfunktion arvo paranee. Tämä on tietysti myös odotettua, sillä vaiheen siirrolla parannetaan selvästi mahdollisuutta kuormittaa aiemmin täysin ilman kuormaa ollutta ajanhetkeä ($t = 6$). Mielenkiintoisempi havainto voidaan sen sijaan tehdä, mikäli työ 2 siirretään alkamaan vielä yhden aikayksikön myöhemmin. Tällöin kuormitushuippu muodostuu uudelle aikavälille {5,9} Taulukko 6 mukaisesti.

Taulukko 6. Muutetun ajoituksen kuormitushuippu

		Ajanhetki t2									
		1	2	3	4	5	6	7	8	9	10
Ajanhetki t1	1	0	0	0	18,8	15	12,5	10,7	20,6	23,9	21,5
	2		0	0	25	18,8	15	12,5	23,6	26,9	23,9
	3			0	0	0	0	0	15	20	17,5
	4				0	0	0	0	18	23,3	20
	5					0	0	0	22,5	28	23,3
	6						0	0	0	12,5	10
	7							0	0	16,7	12,5
	8								0	0	0
	9									0	0
	10										0

Havaitaan, että minimikuormitushuippu kasvaa hiukan (27,5 -> 28) siirron seurauksena.

Toisen kustannusfunktion osalta tilanne on Taulukko 7 mukainen:

Taulukko 7. Muutetun ajoituksen maksimikuormituksen neliösumma

Ajanhetki	1	2	3	4	5	6	7	8	9	10	
Resurssin maksimikuormitus	0	75	75	75	90	90	140	140	50	0	Yhteensä
Maksimikuormi- tuksen neliö	0	5625	5625	5625	8100	8100	19600	19600	2500	0	74775

Esimerkissä toisen kustannusfunktion arvo siis paranee (pienenee), vaikka minimikuormitushuippu kasvaa. Tämän vuoksi kustannusfunktiot joudutaan asettamaan tärkeysjärjestykseen algoritmia suunniteltaessa. Kuormitushuipun minimoimiseen käytetty kustannusfunktio valitaan tärkeämmäksi, eikä sen heikkenemistä sallita, vaikka toisen funktion arvo paranisi. Tätä valintaa voi perustella sillä, että kuormitusasteen vaikutukset tuotannon kustannuksiin eivät käyttäydy lineaarisesti. Yleensä tavanomaisen kuormitusasteen ylittyessä kapasiteetin säätöön liittyvien kustannusten kasvu kiihtyy.

5.3 Ongelman matemaattinen muotoilu

Tässä työssä esitettävän suunnittelumethodin tavoitteena on luoda tuotantoaikataulu, jossa kaikki ajoitusreunaehdot toteutuvat ja resursseja kuormitetaan mahdollisimman tasaisesti. Niemi (2004) on aiemmin esittänyt optimointiongelman matemaattisen muotoilun lähes vastaavana, kuin se tässä esitetään. Ongelmassa käytetään kahta eri tavoitefunktiota, joiden perusteella ajoituksen hyvyttä arvioidaan. Ensimmäisellä tavoitefunktiolla pyritään minimoimaan resurssien kuormitushuippujen summaa. Tämä voidaan ilmaista seuraavasti:

$$\text{Minimoi } \sum_{k=1}^m P_k, \text{ niin että} \quad (1)$$

$$P_k - \sum_i r_{ikt} \geq 0, \text{ kaikille } k, t, \quad (2)$$

missä k on resurssin indeksi, m resurssien lukumäärä, t ajanhetken indeksi ja r_{ikt} työvaiheen i kuormitus resurssille k ajanhetkellä t . Tässä yhtälö (1) on tavoitefunktio ja epäyhtälö (2) on reunaehto, joka asettaa kunkin kuormitushuipun P_k vähintään yhtä suureksi kuin resurssin k suurin kuormitus millä tahansa ajanhetkellä (vrt kohdan 5.2. Taulukko 3).

Suurien kuormitushuippujen lisäksi myös kuormituksen vaihtelu eri ajanhetkinä on erittäin epätoivottavaa. Kuormitusvaihtelun vähentämiseksi toisella tavoitefunktiolla

pyritään minimoimaan kunkin resurssin kuormituksen neliösummaa suunnitteluhorisontissa. Kuormituksen neliösumma on varsin yleisesti käytetty kuormitusvaihtelun indikaattori, sillä kokonaiskuorman pysyessä vakiona tämän arvo minimoituu täysin tasaisessa kuormituksessa ja kasvaa neliöinnin vaikutuksesta nopeasti vaihtelun lisääntyessä. Tavoitefunktio voidaan muotoilla seuraavasti:

$$\text{Minimoi } \sum_{k=1}^m \sum_{t=1}^{HZ} \sum_i \frac{r_{ikt}^2}{\emptyset}, \quad (3)$$

missä HZ on aikaperiodien lukumäärä ja muut parametrit ovat vastaavia kuin edellä.

Yhtälöstä (3) voidaan havaita, että kuormituksen neliösumman laskemiseen tarvitaan tieto kunkin työvaiheen kuormituksesta myös yksittäisellä ajanhetkellä. Tämän vuoksi optimointisovelluksessa käytetään neliösumman sijasta helpommin ratkaistavaa yhtälöä, joka pyrkii minimoimaan yksittäiselle ajanhetkelle ajoitettavissa olevan työkuorman neliösummaa.

$$\text{Minimoi } \sum_{k=1}^m \sum_t \sum_{u_1=1}^t \sum_{u_2=t}^i \frac{C_{i,u_1+p_i-1} C_{iu_2} W_{ik}^2}{\emptyset} \quad (4)$$

, missä $C_{it} = 1$, jos työvaihe i valmistuu ajanhetkellä t ja 0 muutoin, W_{ik} on työvaiheen i resurssille k aiheuttama työkuorma, ja p_i on työvaiheen i kesto. Yhtälössä sulkulausekkeen sisällä oleva kolmoissumma varmistaa, että vaiheen työkuorma lasketaan neliösummaan jokaiselle ajanhetkelle t , jolloin vaihe on käynnissä (vrt. kohdan 5.2 Taulukko 5).

Muut muuttujat ja reunaehdot

Kunkin työvaiheen suunniteltu valmistumisaika C_i :

$$C_i = \sum_{t=1}^{HZ} t C_{it} \quad (5)$$

, missä $C_{it} = 1$, jos työvaihe i valmistuu ajanhetkellä t ja 0 muutoin.

Jokainen työvaihe valmistuu täsmälleen kerran, joten:

$$\sum_{t=1}^{HZ} C_{it} = 1 \quad (6)$$

Työnvaihe voidaan aloittaa aikaisintaan sitä edeltävien työnvaiheiden valmistumista seuraavana ajanhetkenä. Toisaalta mikään työnvaihe ei voi päättyä tilauksen viimeistä mahdollista valmistumishetkeä (d_i) myöhemmin. Tämän vuoksi:

$$d_i \geq C_i + p_i \quad \text{kaikille } i \in P(i) \text{ ja kaikille } i \quad (7)$$

$P(i)$ on vaihetta i edeltävien työnvaiheiden joukko ja p_i on työnvaiheen i kesto. Työnvaiheen aikaisinta aloittamishetkeä voi rajoittaa tilauksen aikaisin aloitushetki a_i . Yksikään työnvaihe ei voi alkaa ennen sen sisältävän tilauksen aloittamista, joten:

$$C_i - p_i \geq a_i \quad (8)$$

Lisäksi tilauksen tai sen haaran maksimiläpäisyaikaa voidaan rajoittaa parametrilla b_i . Määritellään $O(i)$ niiden työnvaiheiden joukoksi, jotka tehdään samaa tilausta varten kuin työnvaihe i . Tällöin

$$\max_{g \in O(i), i} (C_g - C_i + p_i) \leq b_i \quad (9)$$

Binaarimuuttuja x_{it} on yksi, jos työnvaihe i on ajoitettu suoritettavaksi ajanhetkenä t ja nolla muulloin. Tällöin

$$x_{it} = \sum_{u=t}^{t+p_i-1} C_{iu} \quad (10)$$

Yksittäisen työnvaiheen tietylle resurssille tietyllä ajanhetkellä aiheuttamaa kuormitusta kuvaavan muuttujan r_{ikt} täytyy olla nolla vaiheen ajoituksen ulkopuolella. Tämä voidaan määrittää seuraavasti, kun M on suuri luku:

$$r_{ikt} \leq Mx_{it} \quad (11)$$

Työnvaiheen i kuorma resurssille k , W_{ik} , täytyy allokoida vaiheen suoritusajalle, joten:

$$\sum_{t=1}^{H_Z} r_{ikt} = W_{ik} \quad (12)$$

Neumann ja Zimmermann (1999, ss. 592 - 595) osoittavat, että lähes vastaavasti muotoiltu yksittäisen projektin RLP -ongelma kuuluu NP-vaikeiden¹ luokkaan sekä

¹ NP-vaikeille (*Non-deterministic Polynomial-time hard*) ongelmille ei ole löydetty ongelman varmasti polynomiaalisessa ajassa optimaalisesti ratkaisevaa algoritmia. Matemaattisia ongelmaluokkia ja niiden laskennallista vaikeutta ovat käsitelleet tarkemmin esimerkiksi Garey ja Johnson (1979).

lineaarisen tavoitefunktion (1) että neliöllisen tavoitefunktion (3) osalta. Tämän todistuksen perusteella myös tässä työssä ratkaistava ongelma on NP-vaikea.

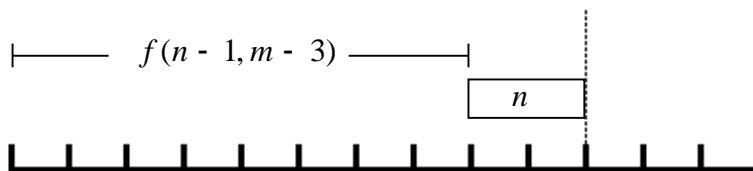
5.4 Esimerkki optimointiongelman koosta

Optimointiongelman kokoa voidaan arvioida esimerkin avulla. Oletetaan kaikkien ajoitettavien tilausten olevan työnvaiheiden lukumäärän suhteen keskenään identtisiä, ja että kukin tilaus koostuu ainoastaan sarjasta peräkkäisiä työnvaiheita. Lisäksi oletetaan, että jokaisella tilauksella on yhtä suuri ajoitusvara, eikä tilausten läpäisyaikaa ole erikseen rajattu. Tällöin optimointiongelman ratkaisuvaihtoehtojen lukumäärään vaikuttavia muuttujia on kolme: tilausten lukumäärä z , yksittäisen tilauksen työnvaiheiden lukumäärä n , sekä tilauskohtainen ajoitusvara m . Ajoitusvaralla m tarkoitetaan tässä yksittäisen työnvaiheen ajoitusvaihtoehtojen lukumäärää. Koska kaikki tilauksen työnvaiheet määriteltiin tehtäväksi peräkkäin, on ajoitusvara m yhtä suuri jokaiselle tilauksen työnvaiheelle.

Yksittäisen tilauksen ajoitusvaihtoehtojen lukumäärä voidaan esittää kahden muuttujan (n, m) funktiona. Ajatellaan aluksi perustilannetta, jossa ajoitettavana on ainoastaan yksi työnvaihe. Tällöin ajoitusvaihtoehtojen lukumäärä määräytyy seuraavasti suoraan ajoitusvaran mukaan

$$f(1, m) = m \quad m^3 - 1. \quad (13)$$

Yleisen tason ongelma voidaan aina pilkkoa pienempiin osaongelmiin, joten yhtälö voidaan ratkaista rekursiivisesti. Tätä on havainnollistettu Kuva 13. Kuvassa tilauksen viimeinen työnvaihe n on ajoitettu päättymään suunnitteluhorisontin kolmanneksi viimeisellä ajanhetkellä. Tällöin ajoitusvaihtoehtojen lukumäärä supistuu riippuvaksi termistä $n - 1$.



Kuva 13. Työnvaiheiden ajoitusvaihtoehtojen ratkaisun supistaminen

Edellisen perusteella ajoitusvaihtoehtojen lukumäärä voidaan ratkaista käymällä kaikki n :nen työnvaiheen ajoitukset läpi, jos ratkaisu tunnetaan eri ajoitusvaroilla työnvaiheiden lukumäärän ollessa $n-1$. Näin ollen yksittäisen tilauksen ajoitusvaihtoehtojen lukumäärä voidaan esittää rekursioyhtälönä, jonka alkuehto tunnetaan

$$\begin{aligned}
 & f(1, m) = m \\
 & f(n, m) = \sum_{i=1}^m f(n-1, i) \quad n > 1 \\
 & m \geq 1.
 \end{aligned} \tag{14}$$

Taulukko 8 on laskettu yksittäisen tilauksen ajoitusvaihtoehtojen lukumääriä eri ajoitusvaran ja työvaiheiden lukumäärän arvoilla.

Taulukko 8. Yksittäisen tilauksen ajoitusvaihtoehtojen lukumäärä ajoitusvaran ja työvaiheiden lukumäärän funktiona.

		Ajoitusvara									
		1	2	3	4	5	6	7	8	9	10
Työvaiheita	1	1	2	3	4	5	6	7	8	9	10
	2	1	3	6	10	15	21	28	36	45	55
	3	1	4	10	20	35	56	84	120	165	220
	4	1	5	15	35	70	126	210	330	495	715
	5	1	6	21	56	126	252	462	792	1287	2002
	6	1	7	28	84	210	462	924	1716	3003	5005
	7	1	8	36	120	330	792	1716	3432	6435	11440

Esimerkin yksinkertaistamiseksi ajoitettavien tilausten oletettiin olevan työvaiheiden lukumäärän ja ajoitusvaran osalta identtisiä. Eri tilauksiin kuuluvien työvaiheiden ajoitukset eivät ole toisistaan riippuvia, joten ajoitusvaihtoehtojen lukumäärä kasvaa tilausten lukumäärän suhteen eksponentiaalisesti.

$$\begin{aligned}
 & f(1, m, z) = m^z \\
 & f(n, m, z) = \sum_{i=1}^m f(n-1, i) z^i \quad n > 1 \\
 & m \geq 1, z \geq 1.
 \end{aligned} \tag{15}$$

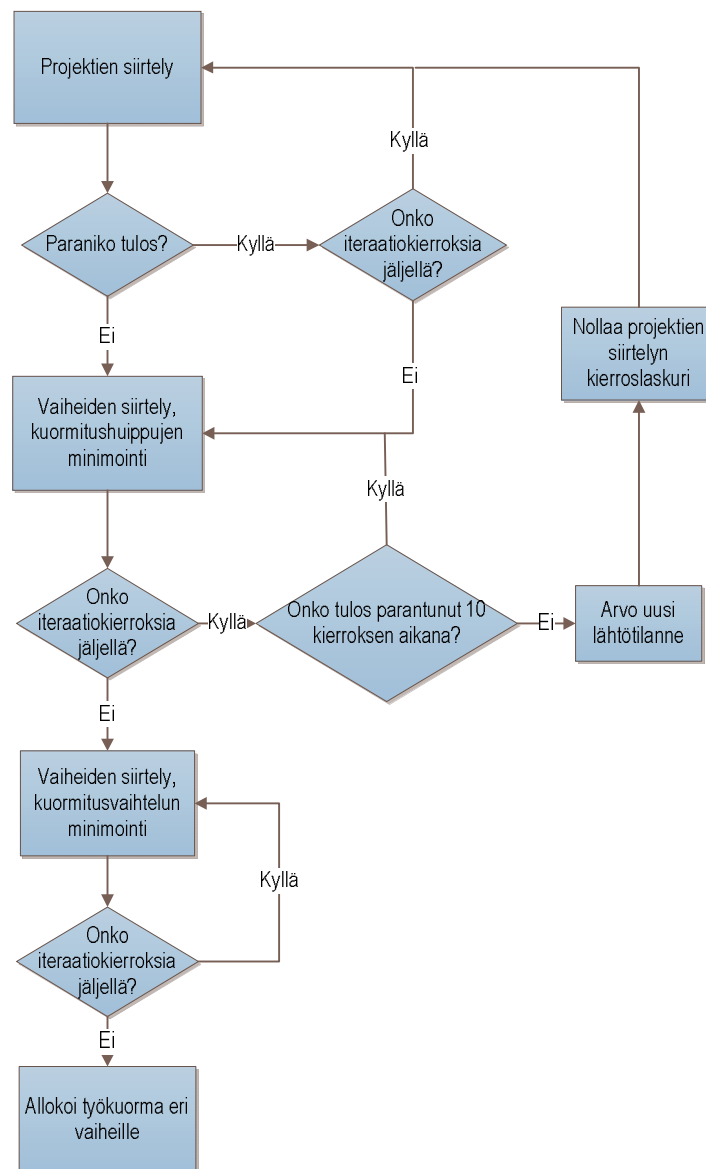
Realististen optimointiongelmiä monimutkaisuutta voidaan havainnollistaa seuraavalla esimerkillä. Ajoitettavien tilausten lukumäärä on kymmenen ja jokainen niistä sisältää viisi työvaihetta. Oletetaan jokaisen tilauksen suunnitteluhorisontin olevan kaksi viikkoa, jolloin ajoitusvara on kymmenen työpäivää. Tällöin eri ajoitusvaihtoehtojen lukumäärä on yhtälön (14) ja Taulukko 8 perusteella noin $2 \cdot 10^{13}$. Vaihtoehtojen lukumäärän suhteen polynomisesti ja tilausten lukumäärän suhteen eksponentiaalisesti kasvavan ongelman koon vuoksi on varsin ilmeistä, että systemaattinen ajoitusvaihtoehtojen läpikäynti (engl. *brute force* -metodi) ei ole tehokas tapa kyseisen ajoitusongelman ratkaisemiseksi.

Kaikki tässä luvussa esitetyt yhtälöt ja laskut ovat esimerkkejä, eikä niitä voida suoraan yleistää koskemaan tilanteita, joissa luvun alussa esitetyt oletukset eivät toteudu. Esimerkit kuvaavat kuitenkin varsin realistisesti eri tekijöiden vaikutusta optimointiongelman kokoon.

6. OHJELMAN KUVAUS

6.1 Hierarkkinen ratkaisualgoritmi

Heuristiikan perusajatuksena on hierarkkinen rakenne, jossa optimointiongelma muuttuu monimutkaisemmaksi aina alemmalle tasolle siirryttäessä. Tällä pyritään nopeuttamaan algoritmin suoritusta. Ajoitusalgoritmi perustuu yksinkertaiseen työnvaiheiden siirtelymetodiin, jossa eri ajoitusten välistä paremmuutta verrataan kohdassa 5.3. esitellyillä tavoitefunktioilla (1) ja (4). Optimointialgoritmin kolmessa ensimmäisessä vaiheessa työnvaiheille pyritään etsimään mahdollisimman tasaisen resurssien kuormituksen mahdollistava ajoitus. Optimoinnin viimeisessä vaiheessa kunkin työnvaiheen työkuorma allokoidaan eri ajanhetkille vaiheen ajoituksen puitteissa. Kuva 14 on esitetty algoritmin pääohjelman logiikka virtauskaaviona.



Kuva 14. Ajoitusalgoritmin toiminta

Projektien siirtely

Algoritmin ensimmäisessä vaiheessa projekteja liikutellaan kokonaisina, eli projektien valmistusajankohtia muutetaan, mutta vaiheiden suhteellinen ajoitus säilytetään ennallaan. Kunkin projektin osalta käydään läpi sen kaikki mahdolliset ajoitukset. Projekteja siirrellään yksi kerrallaan niin, että muiden projektien ajoitusta ei muuteta yhtä siirrettäessä. Jokaisen siirron jälkeen lasketaan eri resurssien kuormitushuippujen summa ja paras siirto tallennetaan. Siirtelyn jälkeen projekti palautetaan sen lähtöajoitukseen ja siirrytään käsittelemään seuraavaa projektia. Kun kaikki projektit on käsitelty, toteutetaan paras tallennettu siirto ja siirtely aloitetaan uudelleen ensimmäisestä projektista. Tätä jatketaan niin kauan kunnes käyttäjän määrittelemä siirtelykierrosten maksimilukumäärä saavutetaan, tai kustannusfunktion arvoa ei saada enää parannettua.

Kokonaisten projektien siirtelyn ajatuksena on tuottaa nopeasti kohtuullinen suunnitelma, vaikka lähtötilanne olisi huono. Tällä menetelmällä projektit saadaan hajautettua melko tasaisesti koko suunnitteluhorisontin ajalle.

Vaiheiden siirtely, kuormitushuippujen minimointi

Optimoinnin toisessa vaiheessa siirrellään yksittäisiä työvaiheita. Siirtelyprosessi vastaa pitkälti projektien siirtelyä. Tämä vaihe on kuitenkin huomattavasti edellistä monimutkaisempi, koska työvaiheita käsitellään yksi kerrallaan. Algoritmin nopeuttamiseksi kuormitushuippujen pienenemisen edellytykset tarkastetaan ennen kunkin työvaiheen siirtelyn aloittamista. Jotta työvaiheen siirrolla voidaan pienentää jonkin resurssin kuormitushuippua, tulee seuraavien ehtojen toteutua:

1. Työvaiheen täytyy olla ajoitettu sellaisen resurssin kuormitushuipun alueelle, jota se kuormittaa.
2. Työvaiheella täytyy olla niin paljon ajoitusvaraa, että se voidaan ajoittaa ainakin osittain tämän alueen ulkopuolelle.

Tässä vaiheessa työvaihetta siirrellään siis ainoastaan, mikäli molemmat ehdot toteutuvat. Kaikki kuormitushuippuja parantavat siirrot tallennetaan siirtelykierroksen aikana. Kun kaikki työvaiheet on prosessoitu, suoritetaan kierroksen aikana tallennetut siirrot. Tässä vaiheessa tarkastetaan jokaisen siirron kohdalla, että se yhä parantaa kustannusfunktion arvoa ja on ajoitusreunaehtojes puitteissa mahdollinen.

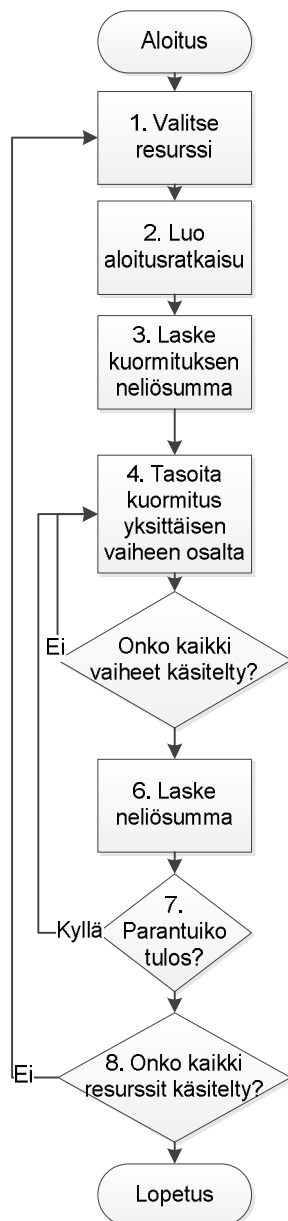
Vaiheiden siirtely, vaihtelun minimointi

Työnvaiheiden ajoituksen tuottamisen viimeisessä vaiheessa resurssien kuormitusvaihtelua pyritään minimoimaan myös kuormitushuippujen ulkopuolelta. Tämä tekee tilanteesta edelleen laskennallisesti monimutkaisemman. Tässä vaiheessa ensisijaisena tavoitteena on edelleen kuormitushuippujen summan minimointi. Tämän lisäksi pyritään minimoimaan kullekin ajanhetkelle sijoitettavissa olevan kuorman neliösummaa. Sijoitettavissa olevan kuorman neliösumman minimointi vähentää aikataulun kuormitusvaihtelua. Jokaisen suoritettun siirron jälkeen joudutaan myös laskemaan molempien tavoitefunktioiden arvo, sillä kuormitushuippujen summan kasvamista ei sallita, vaikka toisen funktion arvo paranisi. Myös tässä vaiheessa kaikki tulosta parantavat siirrot toteutetaan jokaisen siirtelykierroksen lopussa.

Algoritmin kolme ensimmäistä vaihetta tuottavat tuloksena tuotannon karkeasuunnitelman, jossa jokaisen tilauksen ja työnvaiheen aloitushetket on määritetty suunnittelussa käytetyn aikaresoluution tarkkuudella. Viimeistä vaihetta voidaan nimittää hienosuunnitteluksi, sillä siinä ratkaistaan töiden ajoitus karkeasuunnittelun suunnitteluresoluutiota suuremmalla tarkkuudella.

Työn allokointi (hienokuormitus)

Optimoinnin viimeisessä vaiheessa ratkaistaan täysin erilainen ongelma (Kuva 15). Tässä työnvaiheiden ajoitukset ovat reunaehtoja ja tavoitteena on sijoittaa kullekin vaiheelle kuuluva työkuorma niin, että kunkin resurssin kokonaiskuorman vaihtelu (neliösumma) minimoituu. Työn allokoinnin on toteuttanut tutkija Kimmo Ikkala ja sen toteutus on kuvattu artikkeleissa (Toivonen et al., 2006a) ja (Toivonen et al., 2006b). Vaiheissa 2 ja 4 työnvaiheet käsitellään myöhäisimmän valmistumisajan ja lyhyimmän keston mukaan järjestettyinä.



Algoritmin kuvaus:

1. Valitaan käsiteltävä resurssi.
2. Mikäli aikaisempaa kuormitusprofiilia ei ole laskettu, asetetaan profiilin arvot äärettömäksi. Asetetaan kunkin työnvaiheen kokonaiskuorma sen ensimmäiselle ajanhetkelle. Mikäli työkuorma ylittää kuormitushuipun minimiarvon, tai aikaisemman kuormitusprofiilin arvon kyseisellä ajanhetkellä, siirretään ylittävän osan työkuorma myöhemmälle ajanhetkelle.
3. Lasketaan kuormitusprofiilin neliösumma.
4. Valitun työnvaiheen kuorma allokoidaan niin, että kuormitusprofiilista tulee mahdollisimman tasainen koko työnvaiheen keston ajalle. Tämä tehdään siirtämällä työnvaiheen kuormaa suuremman kokonaiskuormituksen ajanhetkiltä matalamman kuormituksen alueelle.
5. Palataan vaiheeseen 4, kunnes kaikki työnvaiheet on käsitelty.
6. Lasketaan kuormitusprofiilin neliösumma.
7. Mikäli neliösumma on parantunut enemmän kuin etukäteen määritetty minimi, palataan vaiheeseen 4.
8. Palataan vaiheeseen 1, kunnes kaikki resurssit on käsitelty.

Kuva 15. Hienokuormitusalgoritmi (muokattu lähteestä Toivonen et al., 2006a)

6.2 Alihankinnan käsittely optimointisovelluksessa

Alihankinnan mallintamiseen voidaan käyttää kahta erilaista lähestymistapaa. Alihankintana tehtävä työvaihe ei varsinaisesti aiheuta resurssikuormitusta tuotantoyksikköön, joten se voidaan mallintaa ilman työkuormaa, jolloin se toimii ainoastaan rajoittavana reunaehtona. Toisaalta voidaan ajatella, että myös alihankinnan kuormitusvaikutuksien tasaaminen hyödyttää koko arvoketjua. Tämän vaikutus näkökulman merkitys korostuu, jos yrityksellä on suhteellisesti suuri vaikutus alihankkijan kuormitusasteeseen. Tällöin alihankkija ei välttämättä pysty tasaamaan tilauskantaansa muilla toimituksilla ja toimitusketjun kokonaistehokkuus heikkenee.

Mikäli alihankintana toimitettavat työvaiheet mallinnetaan ilman resurssikuormaa, jäävät ne rajoittaviksi ajoitusreunaehdoiksi optimointiproseduuriin. Algoritmia on pyritty muuttamaan tehokkaammaksi siten, että työvaiheiden ajoitusta ei muuteta, jollei muutoksella ole mahdollisuutta parantaa kustannusfunktion arvoa. Tämä tarkoittaa sitä, että vaiheen ajoitusta ei muuteta, ellei vaihe aiheuta resurssikuormaa sellaisella aikavälillä, jossa resurssin kuormitus on huipussaan. Tämä puolestaan johtaa siihen, että ilman resurssikuormaa mallinnettuja vaihteita ei siirretä missään vaiheessa, vaikka ne voivat estää paremman optimointituloksen löytymisen.

Tämä ongelma on ratkaistu algoritmista poikkeuskäsittelyllä. Ennen optimoinnin alkua luodaan taulukko, jossa on alihankintana suoritettavat työvaiheet, sekä niitä seuraavat työvaiheet. Vaiheiden siirtelyn aikana siirretään alihankintana suoritettava vaihe aikaisimpaan mahdolliseen aloitushetkeen. Kun seuraavaa vaihetta on siirretty, siirretään alihankintana suoritettava vaihe päättymään välittömästi ennen seuraavan vaiheen alkua. Tämä maksimoi sekä alihankintaa edeltävien että sitä seuraavan vaiheen liikkumavaran optimointiproseduurissa, ja varmistaa algoritmin toiminnan suunnitellusti.

Tutkimuksessa käytetyssä testidatassa käsiteltiin ainoastaan tehtaan sisäisiä toimintoja, eikä alihankinnan mallinnuksella ollut käytännön merkitystä. Karkean tason kapasiteettisuunnittelussa voi alihankinnan mallintaminen kuitenkin olla tarpeellista. Tilanne on sama myös, jos työkalua käytetään projektisuunnitteluun pitkäkestoisissa hankkeissa.

6.3 Työnvaiheiden dynaaminen lyhentäminen

Alihankinnan lisäksi myös muut työnvaiheet voidaan joskus suorittaa nopeammin kuin ajoituksen perusmallissa on ennakoitu. Tällöin aikatauluun jää työnvaihteita, joiden seuraavaa vaihetta ei voida aloittaa, vaikka edeltävä vaihe on jo valmistunut. Optimointialgoritmiin on lisätty optio, joka tarkastaa vaihtelun minimoointikierrosten välissä todellisen resurssiallokaation ja lyhentää työnvaiheiden kestoja, jos työkuorma

niiden alussa tai lopussa on nolla. Tällä muutoksella saadaan lisää ajoituspelivaraa muille vaiheille, jolloin aikataulua voidaan mahdollisesti parantaa. Työnvaiheiden lyhentäminen vaatii myös hienokuormituksen laskemisen, minkä vuoksi sitä käytetään ainoastaan algoritmin lopussa, kun tarkastetaan voiko jo löydettyä hyvää ratkaisua entisestään parantaa.

6.4 Ohjelmakoodi

Optimointisovellus sisältää kaikkineen noin 1800 riviä ohjelmakoodia, minkä lisäksi käyttöliittymä, tietokannanhallinta ja tulosten muotoilu muodostavat yhteensä suunnilleen samankokoisen koodilohkon. Ohjelma on kirjoitettu Visual Basic for Applications (VBA) makrokielellä. Kuva 16 on esimerkki aliohjelmasta, joka toteuttaa algoritmin kokonaisten tilausten siirtelyvaiheen. Aliohjelmassa yksittäisen tilauksen jokaista vaihetta siirretään lopusta alkuun, ja etsitään tilaukselle eniten tuotanto-ohjelmaa parantava aloitushetki. Tämä toistetaan jokaiselle tilaukselle, mutta ainoastaan eniten suunnitelmaa parantanut siirto toteutetaan. Esimerkistä näkyy ahneen algoritmin luonne.

Kuvan selitteen viittaus lukittuihin tilauksiin tarkoittaa sovelluksen ominaisuutta, jolla käyttäjä voi kiinnittää yksittäisen tuotantotilauksen ajoituksen aikataulussa. Ominaisuus on tärkeä käytännön kannalta, sillä tuotannosuunnittelu on jatkuvaa toimintaa. Tuotantojärjestelmässä on yleensä sekä aloitettuja että aikataulutettuja tilauksia, jotka asettavat suunnittelulle rajoitteita. Uudelleensuunnittelu voi myös aiheuttaa häiriöitä ja merkittäviä kustannuksia tuotannossa (Simpson, 1999). Suunnittelua juoksevassa aikahorisontissa pidetään joskus myös omana suunnittelumenetelmänä, jolla ratkaistavaa ongelmaa saadaan yksinkertaistettua (Li & Ierapetritou, 2010).

VBA toimii MS Excelin sisällä tulkittavana makrokielenä, mutta sovellus voidaan myös kääntää konekieliseksi. Tulkittavalla kielellä tehtyä ohjelmaa ajettaessa tulkki kääntää komennot konekielisiksi käskyiksi prosessorille lennossa ohjelman suorituksen aikana. Projektin aikana tehdyssä kokeilussa koodin kääntäminen toi yllättävän pienen nopeushyödyn, joten siitä luovuttiin. Huomattavasti merkittävämpi nopeusetu saavutettiin asettamalla Excelin ruudunpäivitys pois päältä suorituksen ajaksi. Optimointialgoritmin voi kääntää myös itsenäiseksi ratkaisijaksi, jolle annetaan parametreina tilauskanta ja reunaehdot. Tällöin sovelluksen käyttöliittymä ja tietokanta voidaan toteuttaa halutussa ympäristössä esimerkiksi laajemman tietojärjestelmän sisällä.

```

' --- Kokonaisten projektien siirtely alkaa tästä
AllTimeBest = INFINITE           'Kustannusfunktion kaikkien aikojen parhaan arvon alustus
BestSchedule = Finish           'Parhaan löydetyn ajoituksen alustus
For i = 1 To Projectrepetitions  'Projektiensiirtelykierrosten laskuri
    ProjektiKierrosAika = Timer

    Besttotal = INFINITE         'Alustetaan kustannusfunktion paras arvo - väliaikainen apumuuttuja
    For tilaus = 1 To Tilauksia  'Tilausten laskuri
        If Lukittu(tilauksenRiviNro(tilauk), 1) Then 'Tilauk on lukittu
            tilauksenAloitust = INFINITE
            tilauksenLopetus = 0
            For j = 1 To vaiheita(tilauk) 'Etsitään ensimmäisen vaiheen aloitus ja viimeisen vaihe
                rivi = tilauksenRiviNro(tilauk) + j - 1
                If Finish(rivi) - Makespan(rivi) + 1 < tilauksenAloitust Then tilauksenAloitust = Finish(rivi) - M
                If Finish(rivi) > tilauksenLopetus Then tilauksenLopetus = Finish(rivi)
            Next j
            For j = 1 To vaiheita(tilauk)
                rivi = tilauksenRiviNro(tilauk) + j - 1
                EarliestStartingTime(rivi) = tilauksenAloitust 'Poistetaan ajoitusvara tilauksen päistä
                ProjectDueDate(rivi) = tilauksenLopetus
            Next j
        Else 'Tilaukusta ei ole lukittu
            TempSchedule = Finish 'Talletetaan nykyinen ajoitus apumuuttujaan
            TempPeakMatrix = PeakMatrix
            TempPeakValue = PeakValue
            TempWCMatrix = WCMatrix

            For Slack = 0 To (InitialProjectStartSlack(tilauk) + InitialProjectEndSlack(tilauk)) 'Laskuri: käyt
            For vaihe = 1 To vaiheita(tilauk) 'Käydään läpi kaikki projektin vaiheet - laskuri
                rivi = tilauksenRiviNro(tilauk) + vaihe - 1 'Tilauksesta ja vaiheesta rivinro
                sarake = InitialSchedule(rivi) + InitialProjectEndSlack(tilauk) - Slack 'Lähdetään liikkee
                vanhaloppu = Finish(rivi)
                Finish(rivi) = sarake 'Muutetaan rivillä olevan vaiheen ajoitusta
                Call paivitaWCMatrix(rivi, vanhaloppu)
            Next vaihe 'seuraava vaihe
            Total = TotalPeakValue() '!!! Lasketaan kustannusfunktion arvo ja asetetaan apumuuttujaan T

            If Total < Besttotal And Not tabu(Finish) Then 'Jos arvo on pienempi kuin paras tähänastinen
                Besttotal = Total 'Otetaan uusi paras arvo käyttöön
                BestSchedule = Finish 'ja samoin vastaava ajoitus
                BestWCMatrix = WCMatrix
                BestPeakMatrix = PeakMatrix
            End If
        Next Slack 'seuraava ajoitus pelivaran puitteissa

        Finish = TempSchedule 'Palautetaan projekti siihen missä se oli ennen edellä tehtyä siirtely
        PeakMatrix = TempPeakMatrix
        PeakValue = TempPeakValue
        WCMatrix = TempWCMatrix
    End If
Next tilaus 'Siirrytään käsittelemään seuraavaa tilausta

Finish = BestSchedule 'Kaikki tilaukset on käyty läpi ja otetaan paras löydetty ajoitus käyttöön
PeakMatrix = BestPeakMatrix
WCMatrix = BestWCMatrix
Call UpdateTabuList(Finish, (TabuPaivitystenLkm Mod TabuListLength) + 1) 'Päivitetään tabulista nyt 1
TabuPaivitystenLkm = TabuPaivitystenLkm + 1

If Besttotal < AllTimeBest Then 'Jos paras ajoitus on parempi kuin kaikkien aikojen paras niin...
    AllTimeBest = Besttotal 'päivitetään kaikkien aikojen paras
    AllTimeBestSchedule = BestSchedule 'ja päivitetään vastaava ajoitus
Else
    ProjektiKierrosAika = Timer - ProjektiKierrosAika
    ProjektiKierrosAika = ProjektiKierrosAika * (Projectrepetitions - i)
Exit For
End If
If Besttotal < AllTimePeakValue Then AllTimePeakValue = Besttotal
ProjektiKierrosAika = Timer - ProjektiKierrosAika
ProjektiKierrosAika = ProjektiKierrosAika * (Projectrepetitions - i)
kayttoliittyma.Label38.Caption = "Siirrellään projekteja... " & i + 1 & "/" & Projectrepetitions
KirjoitaLokirivi 'Kirjoitetaan tämän kokeilukierroksen tulokset Excel-taulukkoon
Next i 'Seuraava projektiensiirtelykierr

```

Selite

Muuttujien alustus

Etsitään tilauksen aikaisimman vaiheen aloitus ja myöhäisimmän lopetus

Poistetaan ajoitusvara lukituista tilauksista

Siirretään tilauksen jokaista vaihetta lopusta alkuun

Päivitetään siirron vaikutus resurssien työkuormaan

Lasketaan kustannusfunktion uusi arvo Verrataan tulosta parhaaseen löydettyyn, tallennetaan jos arvo parani

Palataan lähtötilanteeseen ennen seuraavan tilauksen siirtoa

Toteutetaan paras löytynyt siirto

Tallennetaan paras löydetty ajoitus

Jos tulos ei parantunut, siirrytään yksittäisten vaiheiden siirtelyyn

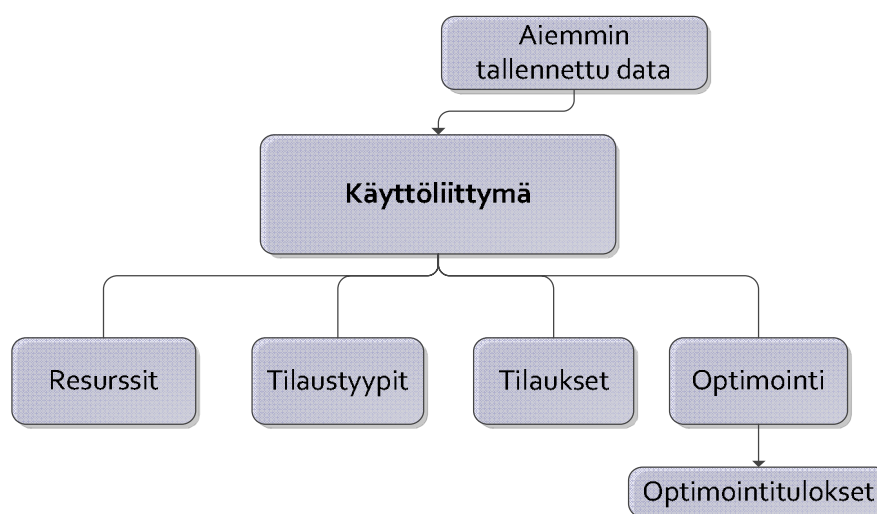
Tilatietojen päivitys

Seuraava kierros

Kuva 16. Esimerkki ohjelmakoodista Microsoft Excelissä

6.5 Käyttöliittymän yleiskuvaus

Optimointisovellus on toteutettu Microsoft Excelin VBA -makrokielellä. Microsoft Exceliä käytetään tietojen tallentamiseen sekä optimointitulosten esittämiseen. Rakenteeltaan käyttöliittymä on jaettu neljään osaan alla olevan Kuva 17 mukaisesti. Resurssit-sivulla määritellään käytettävissä olevat resurssit sekä niiden suhteelliset painokertoimet ja mahdolliset kombinaatiot. Tilaustyyppejä voi käyttää uuden tilauksen pohjana, jolloin tilausten tietojen syöttäminen helpottuu ja nopeutuu. Tilaukset ovat optimointiproceduurin syötetietoja, ja Optimointi-sivulla määritetään optimointiproceduuriin liittyvät parametrit.



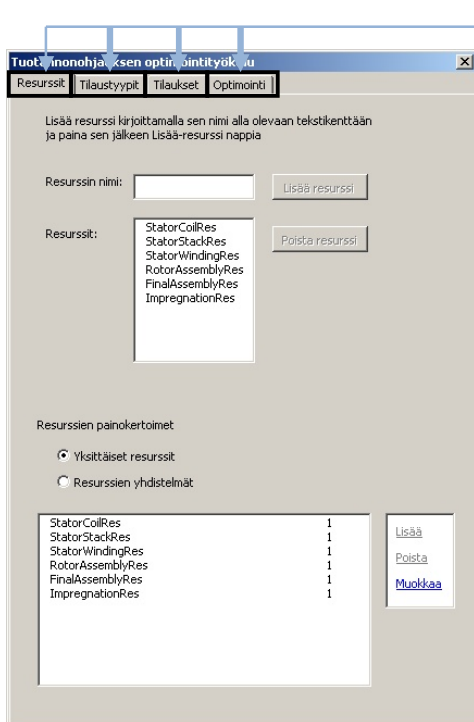
Kuva 17. Käyttöliittymän rakenne

Käyttöliittymän tarkoituksena on helpottaa tietojen syöttämistä ja havainnollistaa eri syötetietojen merkitystä. Käyttöliittymää käytettäessä tarkistetaan myös syötetietojen oikeellisuus niiden antohetkellä. Tilaustiedot voi syöttää myös suoraan Microsoft Excelin laskentataulukkoon, jolloin syötteiden arvoja ei tarkisteta. Optimointiajon suorittamisen alussa sovellus tarkistaa vielä syötetiedot tyypillisimpien virheiden osalta, mutta kaikkia virheitä ei tässä välttämättä havaita. Käytännössä ongelmallisimpia ovat loogisuusvirheet, jotka eivät riko optimoinnin reunaehtoja.

Resurssit

Optimointiongelma ei ole kapasiteettirajoitteinen, joten resursseille ei aseteta muita parametreja kuin niiden nimet. Samalla sivulla voidaan kuitenkin asettaa painokertoimet eri resursseille. Painokertoimilla voidaan muuttaa resurssien suhteellista vaikutusta tavoitefunktioiden (kts. Luku 5.3., yhtälöt (1) ja (4)) arvoon. Painokerroin voidaan asettaa myös kahden resurssin yhdistelmälle, jolloin optimoinnissa pyritään tasaamaan

myös näiden yhteenlaskettua kuormaa. Kuva 18 on esitetty suunnittelutyökalun pääkäyttöliittymän resurssi-välilehti, sekä tilauksen määrittely -ikkuna.



Lisää resurssi kirjoittamalla sen nimi alla olevaan tekstikenttään ja paina sen jälkeen Lisää-resurssi nappia

Resurssin nimi: Lisää resurssi

Resurssit:

- StatorCoilRes
- StatorStackRes
- StatorWindingRes
- RotorAssemblyRes
- FinalAssemblyRes
- ImpregnationRes

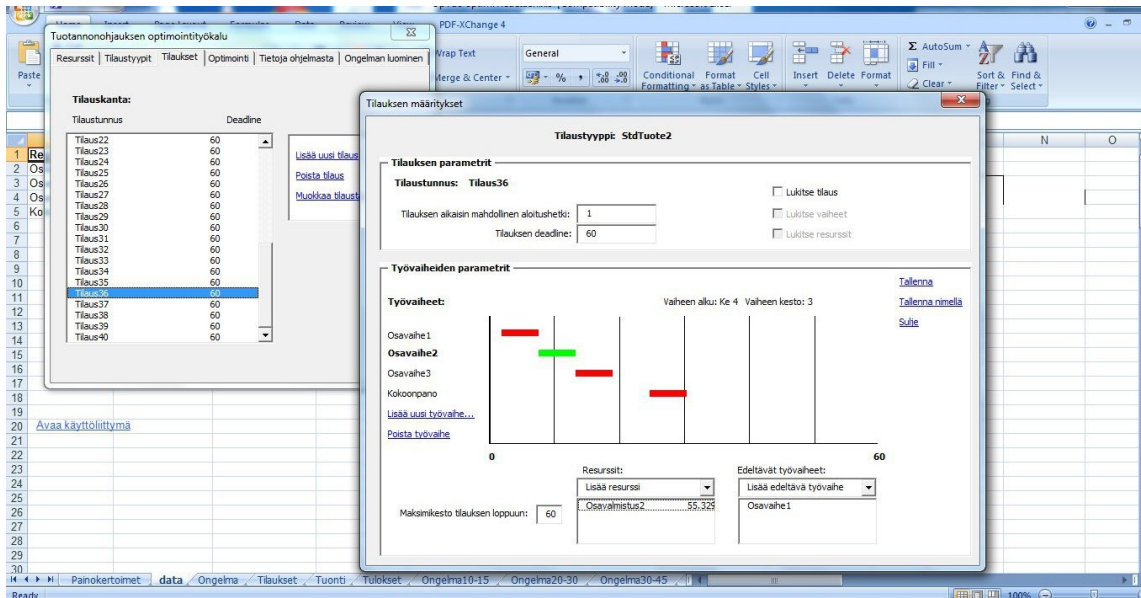
 Poista resurssi

Resurssien painokertoimet

Yksittäiset resurssit
 Resurssien yhdistelmät

StatorCoilRes	1	Lisää
StatorStackRes	1	Poista
StatorWindingRes	1	Muokkaa
RotorAssemblyRes	1	
FinalAssemblyRes	1	
ImpregnationRes	1	

- Resurssien määrittely
- Kuormitusmallien määrittely
- Ajoitettavien tilausten määrittely
- Optimointiparametrit ja sovelluksen käynnistys



Tilaukset

Tilaustunnus	Deadline
Tilaus22	60
Tilaus23	60
Tilaus24	60
Tilaus25	60
Tilaus26	60
Tilaus27	60
Tilaus28	60
Tilaus29	60
Tilaus30	60
Tilaus31	60
Tilaus32	60
Tilaus33	60
Tilaus34	60
Tilaus35	60
Tilaus36	60
Tilaus37	60
Tilaus38	60
Tilaus39	60
Tilaus40	60

Tilauksen määrittely

Tilaustyyppi: StdTuote2

Tilauksen parametrin: Tilaustunnus: Tilaus36

Tilauksen aikaisin mahdollinen aloitushetki: 1
Tilauksen deadline: 60

Työvaiheiden parametrin: Työvaiheet: Osavaihe1, Osavaihe2, Osavaihe3, Kokoonpano

Resurssit: Osavalmistus2 (55.325)

Kuva 18. Suunnittelutyökalun MS Excel -käyttöliittymä

Tilaustyypit

Uusien tilauksien luominen ja vanhojen poistaminen ovat useimmin toistuvia tehtäviä sovelluksen normaalikäytössä. Tämän vuoksi varsinkin uusien tilausten luomisen tulisi olla mahdollisimman yksinkertaista ja nopeaa. Käyttöliittymässä tämä on otettu huomioon antamalla käyttäjälle mahdollisuus luoda tilaustyyppejä. Tilaustyypille määritetään samat parametrit kuin tilauksillekin, mutta työvaiheiden ajoitus määritellään ainoastaan suhteessa tilauksen alkamishetkeen. Tilaukset, jotka poikkeavat vain hiukan toisistaan, kannattaa ryhmitellä yhdeksi tilaustyypiksi.

Tilaukset

Tilaukset-välilehdellä määritellään tilauskanta, joka toimii optimointisovelluksen lähtötietoina. Tässä määritellään kaikki ajoitettavat tilaukset, niiden lähtöajoitus sekä ajoitusreunaehdot.

6.6 Suunnittelutyökalun parametrit

Seuraavassa on esitetty sekä tilaustyypeille että tilauksille määritettävät parametrit sekä niiden kuvaukset. Kunkin parametrin jälkeen on esitetty suluissa sitä vastaava muuttuja Luvun 5 muotoilussa.

Työvaiheiden parametrit

Työkuorma (W_{ik})

Vaiheen työkuorma voidaan määrittää kullekin resurssille erikseen sen mukaan, kuinka paljon vaihe kuormittaa eri resursseja. Työkuorma on positiivinen reaaliluku, jonka yksikköä ei ole rajattu. Yleensä se annetaan tunneissa.

Vaiheen kesto (p_i)

Määrittää kuinka monelle ajanhetkelle vaiheen työkuorma voidaan ajoittaa. Käyttöliittymässä työvaiheen kesto määritetään graafisena Ganttin kaaviona.

Edeltävät työvaiheet ($P(i)$)

Määrittelee työvaiheet, joiden tulee olla valmiita ennen kuin vaiheen suorittaminen voidaan aloittaa.

Vaiheen aloitushetki

Vaiheen aloitushetki määrittää vaiheen ajoituksen lähtötilanteessa.

Maksimikesto tilauksen valmistumiseen (b_i)

Parametri rajoittaa työvaiheen aikaisinta mahdollista ajoitushetkeä suhteessa tilauksen viimeisen työvaiheen valmistumishetkeen. Tämä parametri mahdollistaa tilauksen tai sen tietyn haaran läpäisyajan rajoittamisen.

Tilauskohtaiset parametrit*Aikaisin mahdollinen aloitushetki (a_i)*

Määrittää tilauksen ensimmäisen vaiheen aikaisimman mahdollisen aloitushetken.

Tilauksen deadline (d_i)

Määrittää tilauksen viimeisen vaiheen viimeisen mahdollisen valmistumishetken.

Optimointiparametrit

Seuraavassa on esitetty optimointiajoihin liittyvät parametrit. Parametrien merkitys on kuvattu tarkemmin kohdassa 6.1., jossa esitetään optimointialgoritmin toiminta.

Tilauksen siirtelykierrosten lukumäärä

Optimoinnin ensimmäisessä vaiheessa siirrellään ainoastaan kokonaisia tilauksia näiden ajoitusreunaehtojen puitteissa. Tässä vaiheessa tulokseen saadaan yleensä nopeasti kohtalainen parannus, josta kannattaa jatkaa yksityiskohtaisemmalla optimoinnilla.

Vaiheiden siirtelykierrosten lukumäärä

Toisessa vaiheessa siirrellään kutakin vaihetta yksitellen ja pyritään minimoimaan resurssien kuormitushuippujen summaa.

Jälkikäsitteilykierrosten lukumäärä

Määrittää optimoinnin viimeisen vaiheen pituuden. Jälkikäsitteilykierröksillä vaiheita siirrellään yksi kerrallaan, mutta tulokseen käytetään toista kustannusfunktiota, joka pyrkii minimoimaan resurssikuormituksen vaihtelua. Ensimmäisen kustannusfunktion arvon heikkenemistä ei kuitenkaan sallita.

6.7 Optimointitulokset

Kuva 19 on esitetty optimointiajon jälkeinen tulostusnäkyminen Excelissä. Käyttöliittymään ei ole rakennettu erillistä tulostusnäkymistä, koska tulosten käsittely on helpompaa suoraan taulukkolaskennassa.

Tilaustunnus	Tilauksen aloitus	Tilauksen valmistuminen	Vaihe	Vaiheen aloitus	Vaiheen valmistuminen	Resurssi	Alkuperäinen huippukuorma	Optimoitu huippukuorma
Tilaus1	11	29	Osavaihe1	11	12	Osavalmistus1	78	60
			Osavaihe2	13	15	Osavalmistus2	64,28571429	45,38461538
			Osavaihe3	16	20	Osavalmistus3	77,89473684	60
			Kokoonpano	25	29	Kokoonpano	71,05263158	55,38461538
Tilaus2	12	23	Osavaihe1	12	15			
			Osavaihe2	17	19	Yhteensä	291,2330827	220,7692308
			Osavaihe3	12	15			
			Kokoonpano	20	23	2. kust.funktio	52124500	23311800
Tilaus3	17	36	Osavaihe1	23	26	Neliösumma	2110667,555	1790311,147
			Osavaihe2	31	32			
			Osavaihe3	17	20			
			Kokoonpano	33	36			

Kuva 19. Tulostusnäkyminen MS Excelissä

Optimoinnin tuloksena esitetään tilauskohtaisista tiedoista tilaustunnus, ensimmäisen vaiheen aloitushetki ja tilauksen valmistumishetki. Lisäksi esitetään yksittäisten vaiheiden ajoitukset, joissa alkutilanteesta aikaistetut vaiheet on korostettu vihreällä ja myöhemmäksi siirretyt punaisella. Tuloksista näkyy jokaisen resurssin kuormitushuippu, niiden summa, yhtälön (4) mukaisen optimoinnissa käytetyn kustannusfunktion arvo, sekä yhtälön (3) mukainen kuormitusvaihtelun neliösumma.

Kuva 20 näkyy hienokuormituksen jälkeinen tilanne, jossa vihreä vaakapalkki esittää työnvaiheen kestoa ja yksittäisessä solussa oleva luku kuvaa yksittäiseen aikaikkunaan kuormitettua työmäärää. Hienokuormituksen algoritmin ja tulostusnäkyminen on toteuttanut Kimmo Ikkala.

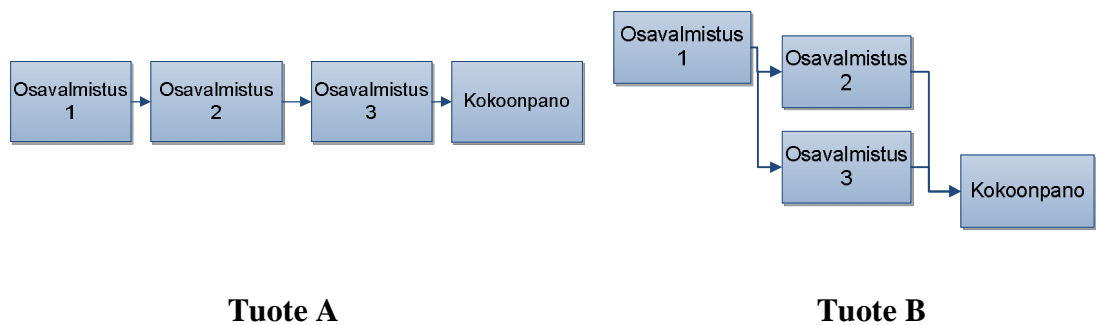
iköitä resurssilla	Tulosta kuormituskuvaaja CTRL+t ja alkuperäinen kuormituskuvaaja CTRL+SHIFT+t	541637,5	499330,3	498695,8	498383,1	498177,4	498058,8	498016,4	497985,2	497959,9	
Order Code		11	12	13	14	15	16	17	18	19	20
Tilaus1	Osavaihe1	13,33333	13,33333	13,33333							
Tilaus2	Osavaihe1								38,4375	0,520833	
Tilaus3	Osavaihe1										
Tilaus4	Osavaihe1				40						
Tilaus5	Osavaihe1					40					
Tilaus6	Osavaihe1										
Tilaus7	Vaihe1					33,33333	33,33333	33,33333			
Tilaus8	Osavaihe1										
Tilaus9	Vaihe1									37,91667	
Tilaus10	Osavaihe1										

Kuva 20. Esimerkki hienokuormituksesta

7. TULOKSET

7.1 Koejärjestely

Ajoitusproseduurin tehokkuuden arviointia varten luotiin joukko testiongelmia, joiden optimiratkaisut tunnettiin. Optimiratkaisuissa jokaiselle resurssille muodostui täysin tasainen kuormitusprofiili maksimaaliselle ajoitushorisontille, jolloin sekä kuormitushuiput että kuormituksen neliösummat minimoituivat. Testiongelmassa käytettiin neljästä työvaiheesta koostuvia tilauksia, joissa jokainen työvaihe kuormitti yhtä resurssia. Tuotteiden valmistusprosessit on esitetty Kuvassa 21. Kumpaakin tuotetyyppiä esiintyi jokaisessa ongelmassa yhtä paljon.



Kuva 21. Esimerkkituotteiden A ja B valmistusprosessit

Ensimmäisessä tuotteessa työvaiheet seurasivat toisiaan sarjassa, kun taas toisessa sallittiin rinnakkaisuus kahden keskimmäisen työvaiheen välillä. Tilausten lukumäärä vaihteli 20:stä 80:een, joten ajoitettavia työvaiheita oli vastaavasti 80 – 360. Ongelmien ajoitushorisontti pidettiin vakiona suhteessa ajoitettavien tilausten määrään. Lyhyin ajoitushorisontti oli 30 aikayksikköä, ja pisin oli 120 aikayksikköä. Ongelmat jaettiin kahteen luokkaan, joista ensimmäisessä jokaisen työvaiheen kesto oli tasan kolme aikayksikköä, ja toisessa neljä tai viisi aikayksikköä (keskimäärin 4,5). Kunkin vaiheen työkuormat arvottiin satunnaisesti niin, että ongelman optimiratkaisuksi muodostui täysin tasainen kuormitusprofiili kaikille resursseille.

Optimointiongelmiin keskeisimmät parametrit on esitetty Taulukko 9. Tilausten lukumäärän ja ajoitushorisontin pituuden lisäksi taulukossa on ilmaistu työvaiheiden kestosta ja ajoitushorisontin pituudesta riippuva tunnusluku, jossa on laskettu yhtä resurssia yhdellä ajanhetkellä kuormittavien työvaiheiden keskimääräinen lukumäärä. Edellytykset kuormituksen tasoittamiseen työvaiheiden sisäisiä kuormitusintensiteettejä vaihtelemalla paranevat tämän tunnusluvun kasvaessa. Parametrien ohella taulukossa on ilmaistu ongelman ratkaisujoukon koko, mikä on suuntaa antava indikaattori ongelman suhteellisesta vaikeudesta verrattuna toisiin saman luokan ongelmiin.

Taulukko 9. Koesarjan ongelmamäärittelyt

Ongelma #	Tilausten lkm	Ajoitushorisontti	Käynnissä olevia töitä / resurssi / aikayksikkö	Ajoitusvaihtoehtoja
1	20	30	2,42	$>10^{80}$
2	30	45	2,26	$>10^{147}$
3	40	60	2,19	$>10^{220}$
4	80	120	2,09	$>10^{546}$
5	20	30	3,91	$>10^{72}$
6	30	45	3,55	$>10^{140}$
7	40	60	3,40	$>10^{213}$
8	80	120	3,19	$>10^{540}$

7.2 Optimointitulokset

Optimointikokeiden tulokset on esitetty Taulukko 10. Tulokset on esitetty sekä kuormitushuippujen summan että neliöllisen tavoitefunktion osalta suhteellisenä erona optimiratkaisuun. Tulokset ja suoritusajat ovat kymmenen toiston keskiarvoja, mutta kuormitushuippujen summan osalta on esitetty myös paras ja huonoin yksittäisen kokeen tulos. Yksittäisen kokeen optimointiparametrit määritettiin niin, että projektien ja vaiheiden siirtelyä suoritettiin niin kauan kuin tulos parani ja jälkikäsitteilykierroksia suoritettiin kaksikymmentä. Tämä toistettiin viisi kertaa, minkä jälkeen paras saavutettu tulos tallennettiin. Työnvaiheiden kesto pidettiin ajoitusproseduurissa vakiona, jotta optimiratkaisut tunnettaisiin.

Taulukko 10. Koesarjan tulokset, vakiopituiset työnvaiheet

Ongelma #	Kuormitushuippujen summa	Minimitulos	Maksimitulos	Kuormituksen neliösumma	Optimointiaika (s)
1	7,11 %	3,40 %	10,85 %	3,45 %	10,5
2	10,12 %	8,00 %	13,12 %	4,36 %	38,9
3	10,59 %	7,07 %	13,26 %	4,12 %	133,5
4	15,71 %	11,73 %	19,65 %	7,32 %	1570,3
5	3,34 %	0,00 %	9,37 %	2,08 %	8,0
6	3,72 %	0,63 %	7,12 %	1,83 %	44,2
7	5,88 %	3,92 %	9,37 %	3,04 %	160,2
8	4,52 %	2,30 %	6,40 %	2,07 %	2301,6

Ensimmäisen ongelmaluokan (neljä ensimmäistä ongelmaa) tulokset olivat kuormitushuippujen summan osalta noin seitsemästä kuuteentoista prosenttia optimiratkaisua huonompia. Kuormituksen neliösumma oli puolestaan 3,45 - 7,32 prosenttia optimiratkaisua suurempi. Toisessa ongelmaluokassa työnvaiheiden

rinnakkaisuutta esiintyi enemmän. Kuormitushuippujen summa oli keskimäärin 3,34 – 5,88 prosenttia optimia suurempi. Helpoimmalle ajoitusongelmalle löydetty optimiratkaisu on merkitty tuloksissa lihavoituna. Kuormituksen neliösummat olivat noin kahdesta kolmeen prosenttia optimiarvoja suurempia.

Taulukko 11 on esitetty vastaavien kokeiden tulokset, kun työnvaiheiden lyhentäminen sallittiin. Kokeiden tulokset on jälleen suhteutettu optimointiongelmien alkuperäisiin optimiratkaisuihin. Koejärjestelyssä työnvaiheiden lyhentäminen voi kuitenkin pidentää yksittäisen resurssin ajoitushorisonttia, joten todellisia optimiratkaisuja ei tiedetty. Tämän vuoksi tuloksissa esiintyy myös negatiivisia arvoja. Lisäksi tuloksissa on esitetty työnvaiheiden lyhentämisellä saatujen tulosten erotus edellisen koesarjan tuloksiin.

Taulukko 11. Koesarjan tulokset, muuttuvan pituiset työnvaiheet

Ongelma #	Kuormitus-huippujen summa	Minimi	Maksimi	Erotus lyhentämätömään (%-yks.)	Kuormituksen neliösumma	Erotus lyhentämätömään (%-yks.)
1	6,49 %	4,11 %	9,29 %	-0,62 %	1,08 %	-2,37 %
2	8,74 %	5,19 %	11,95 %	-1,38 %	3,24 %	-1,12 %
3	8,54 %	6,19 %	10,82 %	-2,05 %	2,62 %	-1,50 %
4	14,04 %	11,93 %	18,06 %	-1,67 %	4,06 %	-3,26 %
5	-2,88 %	-8,28 %	0,66 %	-6,22 %	-5,09 %	-7,17 %
6	0,18 %	-1,58 %	3,69 %	-3,54 %	-1,64 %	-3,47 %
7	1,37 %	-0,92 %	3,16 %	-4,51 %	-0,79 %	-3,83 %
8	3,14 %	1,36 %	5,43 %	-1,38 %	0,51 %	-1,56 %

Työnvaiheiden lyhentämisen sallimisella saavutettiin parhaimmillaan keskimäärin 6,22 prosenttiyksikön parannus kuormitushuippujen summaan. Neliösumman osalta ero oli suurimmillaan 7,17 prosenttiyksikköä. Voidaan todeta, että joissain tilanteissa työnvaiheiden lyhentämisellä saavutettu etu ajoituksen reunaehdoissa parantaa lopputulosta selvästi.

Tuloksia analysoitaessa kuormitushuippujen summa on yksinkertainen ja helposti tulkittava mittari. Optimiratkaisuun verrattuna siitä voidaan tulkita kuinka paljon yksittäisen resurssin suurin kuormitus keskimäärin poikkeaa optimiratkaisusta. Kuormitushuippujen summan absoluuttisella arvolla ei ole käytännön merkitystä, sillä eri resurssien kuormitushuiput voivat ajoittua eri ajanhetkille. Jos resursseilla esimerkiksi kuvataan henkilöstön työtunteja, tämä mittari ei kerro suurinta työntekijätarvetta. Tuotannosuunnittelun kannalta on oleellista tarkastella kuormitushuippujen summan lisäksi myös yksittäisten resurssien kuormitushuippuja, sillä näiden arvot voivat poiketa toisistaan merkittävästi. Käyttöliittymässä esitetään myös resurssikohtaiset tulokset.

Kuormituksen neliösumma on huomattavasti vaikeammin tulkittava mittari, koska sillä pyritään kuvaamaan kuormituksen käyttäytymistä koko ajoitushorisontissa. Optimiratkaisun ollessa täysin tasainen kuormitusprofiili, poikkeaa kuormituksen neliösumma optimiratkaisusta suhteellisesti korkeintaan yhtä paljon kuin kuormitushuippu.

Erot tuloksissa kahden eri ongelmasarjan (suunnitteluongelmat 1-4 ja 5-8) välillä olivat huomattavia. Tämä oli odotettavaa, sillä työnvaiheiden rinnakkaisuuden lisääminen parantaa edellytyksiä kuormituksen tasaamiseen työnvaiheiden sisäisiä kuormitusintensiteettejä vaihtelemalla. Tulos kuitenkin osoittaa, että optimointiongelman parametreilla on suuri vaikutus algoritmin tehokkuuteen.

Yksittäisten optimointiajojen tuloksissa oli varsin suuria poikkeamia. Lokaaliin optimiratkaisuun juuttuminen on ahneille algoritmeille tyypillinen ongelma, mitä voidaan kuitenkin lieventää suorittamalla useita optimointiajoja eri lähtötilanteista.

Vertailu kaupalliseen optimointimoottoriin

Vertailu kaupalliseen optimointiohjelmaan osoittaa työssä esitetyn algoritmin tehokkuuden (Taulukko 12). Tarkempi koejärjestely on kuvattu artikkelissa (Toivonen et al, 2006). Heuristisella algoritmilla saavutettiin noin kuudessatoista minuutissa kummankin tarkasteltavan kriteerin osalta paremmat tulokset, kuin kaupallisella ohjelmistolla seitsemässä tunnissa. Heuristisella menetelmällä tulokset ovat kummankin kriteerin osalta hyviä, kun taas yksittäistä kriteeriä optimoitaessa tulokset toisen kriteerin osalta jäivät heikoiksi. Tämä vahvistaa osaltaan olettamusta kummankin tavoitefunktion merkityksellisyydestä lopputuloksen hyvyyden kannalta.

Taulukko 12. Vertailu kaupalliseen optimointityökaluun

	Lähtötilanne	Heuristinen proseduri			CPLEX	
		Vakio vaiheiden kesto	Vaihteleva vaiheiden kesto	Paloittain optimointi	Kuormitus-huippujen minimointi	Neliö-summien minimointi
Kuormitus-huippujen summa	193.3	113.7	110.9	133.1	118.5 / 115.3	144.1 / 122.2
Neliösumma	308912	227099	222917	240151	239264 / 235184	244228 / 224026
Optimointiaika		15 min 50 s	16 min 14 s	1 min 29 s	n. 20 min / 7 h	n. 20 min / 7 h

Vertailuun otettu CPLEX on yleisesti käytetty kaupallinen optimointityökalu tämääntyyppisten matemaattisten ongelmien ratkaisuun. Nopeusvertailussa heuristinen

proseduuri hyöttyy siitä, että se on alusta asti kehitetty ratkaisemaan vain tämän tyyppinen ongelma.

Paloittain optimoinnissa on ajoitettu pieni osa tilauskanta kerralla, minkä jälkeen tilausten ajoitus on jäädytetty ja uusi erä on valittu ajoitettavaksi. Tilanne jäljittelee todellista toimintatapaa teollisuudessa, sillä ajoitettujen töiden järjestykseen ei yleensä haluta puuttua ilman pakottavaa tarvetta. Tuotantojärjestyksen muutokset lisäävät virhemahdollisuuksia ja epäjärjestyä tuotannossa. Menetelmä on todella nopea, koska ratkaistavat osaongelmat ovat helppoja. Lopullinen tulos on selvästi huonompi kuin kokonaisen tilauskannan kerralla ajoittaminen, koska optimoinnin vapausasteet ovat olleet pienemmät. Tulosta voi tulkita niin, että lyhyellä suunnittelujänteellä tehtaassa joudutaan ylläpitämään lähes 20 prosenttia ($133 / 111 - 1 = 19,8 \%$) korkeampaa kapasiteettia, jotta tilaukset eivät koskaan myöhästyisi.

8. PÄÄTELMÄT

8.1 Suunnittelumenetelmä

Tuotannosuunnitteluongelma voidaan jakaa määrittely- ja ratkaisuvaiheeseen. Tärkein kysymys suunnittelumenetelmän suhteen onkin ”*ratkaistaanko tehtaan käytännön kannalta merkityksellistä suunnitteluongelmaa?*”. Suunnitteluongelman muotoilussa pyritään kuvaamaan tuotannon tavoitteet ja reunaehdot realistisesti. Tuotantostrategian tulisi ohjata tuotannon tavoitteiden määrittelyä, jotka puolestaan vaikuttavat suunnittelumetodiin ja järjestelmän ohjaustapaan. Suunnitteluongelman reunaehto- ja tavoitteiden asettaminen vaatii syvällistä ymmärrystä tuotantoprosessista, sekä niistä vapausasteista, joita tuotannolla on käytettävissä. Tutkimuksessa suunnitteluongelma on kuvattu kapasiteettirajoittamattomana resurssikuormituksen -ongelmana. Käytännössä tämä tarkoittaa, että aikataulujen pitämisestä ei haluta tinkiä missään tilanteessa.

Työssä valitussa lähestymistavassa on pyritty löytämään tasapaino menetelmän käytännön sovellettavuuden ja matemaattisen ongelmanratkaisukyvyyn välillä. Menetelmän käyttökelpoisuutta lisää ratkaisun nopeus, johon vaikuttaa etenkin heuristinen ratkaisualgoritmi ja kapasiteettirajoituksen puuttuminen. Näistä jälkimmäinen yksinkertaistaa tuotannosuunnitteluongelmaa siten, että vähintäänkin huono ratkaisu on aina löydettävissä ja valittavissa suunnittelun lähtökohdaksi. Kapasiteettirajoitetuissa malleissa tavoitteena voikin olla löytää ensimmäinen kapasiteetti- ja ajoitusreunaehdot toteuttava ratkaisu. Työssä ei ole vertailtu vaihtoehtoisia ratkaisualgoritmeja, sillä tämä vaatisi suuren lisätyömäärän ja vertailu itsessään olisi erillinen tutkimus.

Suunnittelumenetelmää voidaan pitää teollisuuden kannalta relevanttina etenkin tilausohjatussa tuotantoympäristössä. Asetettu tavoite on asiakaslähtöinen, helpottaa kapasiteettisuunnittelua ja mahdollistaa ohjattavuuden parantamisen. Ongelman muotoilussa käytettävät lähtötiedot ovat tyypillisesti tuotannosta jo kerättyjä tai helposti saatavilla olevia. Suurin epävarmuus liittyy töiden vaiheaikoihin. On olemassa myös lähestymistapoja, jotka pyrkivät huomioimaan tämän epävarmuuden. Stokastisten menetelmien käyttö on kuitenkin vaikeaa suunnittelumenetelmän käyttökohteeksi ajattelussa kohtalaisen pienillä volyyymeilla toimivassa teollisuudessa.

Heuristiseen ratkaisumenetelmään on kohtalaisen helppoa lisätä erityisiä yksittäiseen prosessiin liittyviä tuotannon reunaehtoja. Esimerkiksi fyysisesti suurikokoisten töiden ajoittumisesta samaan aikaan voidaan rankaista erikseen, tai lämpökäsittelyn kustannusfunktiota voidaan muuttaa tiettyyn raja-arvoon hakeutuvaksi. Myös resurssien ja resurssiyhdistelmien painokertoimilla voidaan vaikuttaa ratkaisuihin, joita algoritmi suosii tai kiertää.

8.2 Saavutetut tulokset

Tuotannosuunnittelun vapausasteita ovat suunnitteluhorisontin pituus, eli kerralla suunniteltavien töiden lukumäärä, sekä aiemmin suunniteltujen mutta aloittamattomien töiden uudelleensuunnittelu. Tulosten perusteella pidempi suunnitteluhorisontti, jossa ajoitettavia tilauksia on enemmän, mahdollistaa tasaisempien kuormitusprofiilien tuottamisen. Toisaalta suunnittelu pidemmälle tulevaisuuteen kasvattaa suunnitelmaan liittyvää epävarmuutta. Tässä työssä ei ole yritetty vastata kysymykseen ”*kuinka tuotannosuunnittelua tulisi tehdä?*”, mutta on hyvä huomata suunnitteluprosessin vaikuttavan lopputulokseen yhtä merkittävästi kuin suunnittelutyökalunkin. Nopeutensa ansiosta algoritmi soveltuu hyvin myös tutkimustyökaluksi ja sitä on käytetty suunnitteluprosessin analysointiin eri tuotantoympäristöissä (Toivonen & Niemi, 2007).

Työn aikana teollisella ja satunnaisesti tuotetulla datalla tehtyjen kokeiden perusteella suunnittelumenetelmä toimii hyvin tyypillisessä konepajaympäristössä. Menetelmän soveltaminen myös muihin ympäristöihin voi olla mahdollista, mutta vaatii tapauskohtaisen arvioinnin. Tehdyissä kokeissa huonoimmat tulokset poikkesivat varsinkin kuormitushuippujen summan osalta merkittävästi optimituloksesta, mutta algoritmin nopeus mahdollistaa tulosten parantamisen useamman toiston avulla. Kuormitusvaihtelun suhteen tulokset olivat kauttaaltaan melko hyviä. Työkalu itsessään voisi olla pienen yrityksen hyödynnettävissä sellaisenaan. Sovelluksen käytettävyyttä on parannettu lähtötietojen virheentarkastuksella, resurssiryhmien painoarvokertoimilla, graafisella käyttöliittymällä ja suunnittelumallien käytöllä. Työkalun käytettävyyden haasteina ovat lähinnä tiedon ylläpitäminen ja integraatio yrityksen tilauskantaan sekä työjonoihin. Näitä ei voi hallita tehokkaasti irrallisella Excel-työkalulla kovinkaan monimutkaisessa ympäristössä. Laajempi sovellettavuus tutkimuskäytön ulkopuolella vaatisi, että algoritmi otetaan käyttöön kaupallisessa suunnittelutyökalussa.

Tutkimuksen perusteella ei voida päätellä, ettei jokin toinen ratkaisumenetelmä tuottaisi parempia tuotantosuunnitelmia tai olisi vielä nopeampi. Vertailu yleiseen kaupalliseen työkaluun kuitenkin osoittaa, että sovellettu menetelmä ei ole huono. Kun huomioidaan tuotannosuunnittelun pragmaattisuus ja käytettävään informaatioon liittyvät epävarmuudet, voidaan todeta tutkimuksessa kehitetyn konstruktion tuottavan riittävän hyviä tuloksia ja soveltuvan sille tarkoitettuun ongelmakenttään.

8.3 Kehitysmahdollisuudet

Metaheuristinen ratkaisualgoritmi etsii parhaan mahdollisen ratkaisun annettujen ajoitusreunaehtojen puitteissa, mutta ei takaa että saatavilla olisi riittävä kapasiteetti suunnitelman toteuttamiseen. Jos riittävää kapasiteettia ei ole saatavilla, tuotannosuunnittelija joutuu löysentämään yksittäisten tilausten ajoitusreunaehtoja toteutettavissa olevan suunnitelman aikaansaamiseksi. Suunnittelumenetelmä hyötyisi

merkittävästi analytiikasta, joka kertoisi minkä tilauksen ajoitusreunaehdoja muuttamalla saavutetaan suurin kokonaisvaikutus.

Muutoksenhallinta on suunnitteluprosesseissa vaikeasti hallittava tekijä, johon tässä työssä ei ole paneuduttu. Tuotanto on luonteeltaan dynaaminen ympäristö, josta tiedetään että kaikki ei aina etene suunnitelman mukaan. Toimitukset eivät saavu ajallaan, osissa on virheitä ja työntekijät sairastuvat. Tuotannosuunnittelun näkökulmasta kiinnostavia tutkimuskohteita ovat esimerkiksi varakapasiteetin mitoitus ja uudelleensuunnitteluprosessi. Uudelleensuunnitteluun liittyvät tavoitteet muutosten minimointiin ja myöhentämiseen, joilla pyritään häiritsemään operatiivista toimintaa mahdollisimman vähän.

Yleisemmällä tasolla ymmärryksen lisääminen koko tuotannosuunnitteluprosessista on mielenkiintoinen tutkimusalue, joka elää tuotannonohjauksessa tapahtuvien muutosten mukana. Nykyinen nopeasti etenevä digitalisaatiotrendi mahdollistaa aiempaa laajempien kokonaisuuksien reaaliaikaisen hallinnan, mikä todennäköisesti vaikuttaa myös siihen, miten suunnitteluprosessi tulisi järjestää.

LÄHTEET

- Amaro, G., Hendry, L., Kingsman, B. (1999). Competitive advantage, customisation and a new taxonomy for non make-to-stock companies. *International Journal of Operations & Production Management*, Vol. 19 (4), ss. 349-371.
- ANSI/ISA-95.00.03-2005, Enterprise-Control System Integration, Part 3: Models of Manufacturing Operations Management.
- Ballard, H. G. (2000). The last planner system of production control (Doctoral dissertation, The University of Birmingham).
- Baykasoglu, A. (2001). MOAPPS 1.0: Aggregate production planning using the multiple-objective tabu search. *International Journal of Production Research*, Vol. 39 (16), ss. 3685-3702.
- Benton, W. C., & Shin, H. (1998). Manufacturing planning and control: The evolution of MRP and JIT integration. *European Journal of Operational Research*, 110(3), ss. 411-440.
- Brucker, P., Drexl, A., Möhring, R., Neumann, K. Pesch, E. (1999). Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research* Vol. 112, ss. 3 – 41.
- Buxey, G. (2003). Strategy not tactics drives aggregate planning. *International Journal of Production Economics*, 85(3), ss. 331-346.
- Chen, Y.-K. & Liao, H.-C. (2003). An investigation on selection of simplified aggregate production planning strategies using MADM approaches. *International Journal of Production Research*. Vol 41(14), ss. 3359-3374.
- De Reyck, B., Herroelen, W. (1998). A branch-and-bound procedure for the resource-constrained scheduling problem with generalized precedence relations. *European Journal of Operational Research*, Vol 111(1), ss. 152 – 174.
- Demeter, K. (2003). Manufacturing strategy and competitiveness. *International Journal of Production Economics*, 81, ss. 205-213.
- Garey, M., Johnson, D. (1979). *Computers And Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, W.H. Freeman and Company. 338 s.
- Ganesh, K., Punniyamoorthy, M. (2005). Optimization of continuous-time production planning using hybrid genetic algorithms-simulated annealing. *The International Journal of Advanced Manufacturing Technology*, Vol 26 (1-2), ss. 148-154.

- Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: harmony search. *Simulation*, 76(2), 60-68.
- Geem, Z. W. (2007). Harmony search algorithm for solving sudoku. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems* (pp. 371-378). Springer Berlin Heidelberg.
- Glover, F. (1990). Tabu Search: A Tutorial. *Interfaces*, Vol. 20 (4), ss. 74-94.
- Goldratt, E. M., & Cox, J. (1984). *The goal: Excellence in manufacturing*. North River Press.
- Goldratt, E. M. (1988). Computerized shop floor scheduling. *The International Journal of Production Research*, 26(3), ss. 443-455.
- Harjunkoski, I., Maravelias, C. T., Bongers, P., Castro, P. M., Engell, S., Grossmann, I. E., Hooker, J., Méndez, C., Sand, G., Wassick, J. (2014). Scope for industrial applications of production scheduling models and solution methods. *Computers & Chemical Engineering*, 62, 161-193.
- Holt, C., Modigliani, F., Muth, J., Simon, H. (1960). *Planning Production, Inventories, and Work Force*. Englewood Cliffs, N. J. Prentice-Hall, Inc. 419 s.
- Hopp, W. J., & Spearman, M. L. (2008). *Factory physics*. McGraw-Hill Irwin Irwin.
- Kallrath, J. (2002). Planning and scheduling in the process industry. *OR Spectrum*, Vol 24 (3), ss. 219-250.
- Kirkpatrick, S., Gelatt, C., Vecchi, M. (1983). Optimization by simulated annealing. *Science, New Series*, Vol. 220, No. 4598, ss. 671-680.
- Kis, T. (2005). A branch-and-cut algorithm for scheduling of projects with variable-intensity activities. *Mathematical Programming*, Vol. 103(3), ss. 515 – 539.
- Li, Z., & Ierapetritou, M. G. (2010). Rolling horizon based planning and scheduling integration with production capacity consideration. *Chemical Engineering Science*, 65(22), ss. 5887-5900.
- Little J. D. C. (1961). A proof for the queuing formula: $L = \lambda W$. *Operations Research*, Vol. 9 (3), ss. 383–387
- Man, K., Tang, K., Kwong, S. (1996). Genetic Algorithms: Concepts and Applications. *IEEE Transactions on Industrial Electronics*, Vol 43 (5), ss. 519-534.

- Mula, J., Poler, R., Garcia-Sabater, J. P., & Lario, F. C. (2006). Models for production planning under uncertainty: A review. *International journal of production economics*, 103(1), ss. 271-285.
- Nelder, J., Mead, R. (1965). A simplex method for function minimization. *The Computer Journal*, Vol. 7 (4), ss. 308-313.
- Neumann, K., Zimmermann, J. (1999). Resource levelling for projects with schedule-dependent time windows. *European Journal of Operational Research*, Vol. 117 (3), ss. 591-605.
- Niemi, E. (2004). Resource levelling in make-to-order mechanical engineering industry, Eighth International Conference on Manufacturing and Management: Proceedings, Gold Coast, Australia, Vol. 1, ss. 163-170.
- Ohno, T. (1988). *Toyota production system: beyond large-scale production*. crc Press.
- Olkkonen, T. (1993). *Johdatus teollisuustalouden tutkimustyöhön*. Teknillinen korkeakoulu.
- Parejo, J. A., Ruiz-Cortés, A., Lozano, S., & Fernandez, P. (2012). Metaheuristic optimization frameworks: a survey and benchmarking. *Soft Computing*, 16(3), 527-561.
- Paraskevopoulos, D. C., Tarantilis, C. D., & Ioannou, G. (2016). An adaptive memory programming framework for the resource-constrained project scheduling problem. *International Journal of Production Research*, 1-19.
- Piper, C. J., Vachon, S. (2001). Accounting for productivity losses in aggregate planning. *International Journal of Production Research*, Vol. 39 (17), ss. 4001 – 4012.
- Plenert, G. (1999). Focusing material requirements planning (MRP) towards performance. *European Journal of Operational Research*, 119(1), ss. 91-99.
- Roth, A. V., & Miller, J. G. (1992). Success factors in manufacturing. *Business Horizons*, 35(4), ss. 73-81.
- Simpson, N. C. (1999). Multiple level production planning in rolling horizon assembly environments. *European Journal of Operational Research*, 114(1), ss. 15-28.
- Spearman, M. L., & Zazanis, M. A. (1992). Push and pull production systems: issues and comparisons. *Operations Research*, 40(3), ss. 521-532.

- Stevenson, M., Hendry, L. C., Kingsman B. G. (2005). A review of production planning and control: the applicability of key concepts to the make-to-order industry. *International Journal of Production Research*, Vol. 43 (5), ss. 869-898.
- Toivonen, V., Niemi, E., Ikkala, K. (2006a). Project Scheduling System for Planning Make-To-Order Production. *International Journal of Agile Manufacturing*, Vol. 9 (1), ss. 87-98.
- Toivonen, V., Ikkala, K., Niemi, E. (2006b). Planning and scheduling system for make-to-order production. *CIRP Annals-Manufacturing Technology*, Vol. 55(1), ss. 493-496.
- Toivonen, V., & Niemi, E. (2007). Production planning issues in a multi-product make-to-order environment. In *Proceedings of the 17th International Conference on Flexible Automation and Intelligent Manufacturing*, June 18-20, 2007, Philadelphia, USA.
- Wang, R.-C., Fang, H. (2001). Aggregate production planning with multiple objectives in a fuzzy environment. *European Journal of Operational Research*, Vol. 133 (3), ss. 521-536.
- Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *Evolutionary Computation*, *IEEE Transactions on*, 1(1), ss. 67-82.
- Yang, X. S. (2013). Metaheuristic optimization: Nature-inspired algorithms and applications. In *Artificial Intelligence, Evolutionary Computing and Metaheuristics* (pp. 405-420). Springer Berlin Heidelberg.
- Zhou, H., & Benton, W. C. (2007). Supply chain practice and information sharing. *Journal of Operations management*, 25(6), ss. 1348-1365.