

SAVONIA



OPINNÄYTETYÖ - YLEMPI AMMATTIKORKEAKOULUTUTKINTO
TEKNIIKAN ALA

SUUNNITTELUVAATIMUSTEN TAVOITEARVOJEN OPTIMOINTI

Case: Specsoft

TEKIJÄ Ari Tanskanen

Koulutusala Tekniikan ja liikenteen ala	
Tutkinto-ohjelma Teknologiaosaamisen johtamisen tutkinto-ohjelma	
Työn tekijä Ari Tanskanen	
Työn nimi Suunnitteluvaatimusten tavoitearvojen optimointi	
Päiväys	19.04.2025
	41/0
Yhteistyötaho Savonia-ammattikorkeakoulu Oy	
<p>Tämän opinnäytetyön tavoitteena oli kehittää laskennallinen optimointityökalu teollisuuden tuotekehitysprosessin tarpeisiin. Työssä kehitetty optimointityökalu on osa <i>Specsoft teknologiateollisuuden tuotekonseptointisovelluksen kehittäminen</i>-hankkeessa kehitettävää sovellusta. Hankkeessa kehitettävällä konseptointisovelluksella voidaan määritellä tuoteominaisuuksia ja niiden merkitystä asiakkaalle, suunnitteluvaatimuksia ja niiden hierarkiaa, teknisten spesifikaatioiden tärkeyttä eri laatutekijöiden näkökulmista, sekä spesifikaatioiden optimaalisia arvoja eri liiketoimintatavoitteiden näkökulmista. Tämän työn tarkoituksena oli kehittää optimointityökalu suunnitteluvaatimusten tavoitearvojen määrittämiseen.</p> <p>Työssä kehitettiin Python-pohjainen optimointityökalu, joka hyödyntää matemaattista mallinnusta ja kolmea erilaista optimointialgoritmia. Työn teoreettisessa osuudessa esitellään tuotekehitysprosessia, matemaattista mallinnusta ja optimointilaskentaa. Käytännön osiossa kehitetyn optimaattorin lähtökohtana toimi aiemmin kehitetty Excel-pohjainen optimaattori, jonka optimointimenetelmien toimivuus oli jäänyt epäselväksi. Työssä käytettyjä optimointialgoritmeja vertailtiin aiempaan Excel-optimaattoriin suorittamalla testilaskentoja, ja analysoimalla niiden tuloksia tilastollisesti. Työn ensisijainen tutkimusmenetelmä oli siten määrällinen tutkimus.</p> <p>Työssä tehdyn optimaattorin avulla kyettiin vertailemaan eri optimointimenetelmiä <i>Specsoft</i>-sovelluksen tarpeisiin. Työn tuloksena syntynyt optimaattori tukee päätöksentekoa tuotekehitysprosessin alkuvaiheessa tarjoamalla laskennallisesti perustellun lähestymistavan suunnitteluvaatimusten asettamiseen. Tulokset osoittavat, että optimointia voidaan hyödyntää tehokkaasti konseptointivaiheen tukityökaluna. Jatkokehityksenä tässä työssä kehitetty optimaattori sisällytettiin <i>Specsoft teknologiateollisuuden tuotekonseptointisovelluksen kehittäminen</i>-hankkeessa kehitettävän sovelluksen osaksi.</p>	
Avainsanat Tuotekehitys, optimointi	

Field of Study Technology, Communication and Transport	
Degree Programme Master's Degree Programme in Engineering Competence Management	
Author Ari Tanskanen	
Title of Thesis Optimizing the target values of design requirements	
Date 19.04.2025	Pages/Appendices 41/0
Client Organisation /Partners Savonia university of applied sciences	
<p>The aim of this thesis was to develop a computational optimization tool for the needs of the industrial product development process. The optimization tool developed in the work is part of an application being developed in a <i>Specsoft-technology industry product concept application development</i>-project. The application being developed in the project can be used for defining product features and the importance and significance of these features to the customer, as well as for designing requirements and their hierarchy. It can be also utilized for characterizing the importance of technical specifications from the perspective of different quality factors and the optimal values of specifications from the perspective of different business goals. Thus, the purpose of this work was to develop an optimization tool suitable for determining target values of design requirements in the industrial development process.</p> <p>In this thesis a Python-based optimization tool was developed, utilizing mathematical modelling and three different pre-existing optimization algorithms. Literature part of the thesis gives an introduction the product development process, mathematical modelling and optimization calculations. For the practical part, the starting point for optimizer development was a previously designed Excel-based optimizer, which functionality and performance had remained unclear. The optimization algorithms used in this work were compared to the previous Excel optimizer by performing test calculations and analysing their results statistically. Therefore, the primary research method of the work was quantitative research.</p> <p>The optimizer developed in this work enabled a comparison of different optimization methods in the <i>Specsoft</i>-application. The optimizer created supports decision-making in the early stages of the product development process by providing a computationally justified approach to setting design requirements. The results show that optimization can be effectively utilized as a support tool in the conceptualization phase. As a further development, the optimizer developed in this work has been now included as part of the application being developed in the <i>Specsoft-technology industry product conceptualization application development</i>-project.</p>	
Keywords Product development, optimization	

SISÄLTÖ

1	JOHDANTO.....	6
2	TUOTEKEHITYS.....	8
2.1	Tuotekehitysprosessi.....	9
2.2	Konseptointivaihe.....	11
2.3	QFD.....	12
2.4	Kano.....	14
2.5	Analyttinen hierarkia prosessi.....	15
3	MATEMAATTINEN MALLINNUS.....	17
3.1	Mallien rakentaminen.....	17
3.2	Tarkkuus ja luotettavuus.....	19
4	OPTIMOINTILASKENTA.....	21
4.1	Optimoinnin vaiheet.....	21
4.2	Lineaarinen optimointi.....	23
4.3	Epälineaarinen optimointi.....	24
5	TYÖN TARKOITUS JA TAVOITE.....	26
5.1	Tarkoitus.....	26
5.2	Tavoite.....	26
5.3	Tutkimuskysymykset.....	26
6	MENETELMÄLLINEN TOTEUTUS.....	27
6.1	Työn toteutus.....	27
6.2	Tutkimusmenetelmät.....	27
6.3	Työn lähtökohta.....	28
6.4	Aineistonhankinta.....	34
6.4.1	Testilaskennat.....	35
6.5	Aineiston analyysi.....	35
7	TULOKSET.....	37
8	JOHTOPÄÄTÖKSET JA POHDINTA.....	39
	LÄHTEET.....	41

KUVALUETTELO

Kuva 1. Yleinen tuotekehitysprosessi (Ulrich & Eppinger 2016, 14)	9
Kuva 2. Yksityiskohtainen tuotekehitysprosessin kuvaus vuokaaviona (Mital & Desai 2014, 17)	10
Kuva 3. QFD-matriisin perusmuoto (Lager 2019, 12)	12
Kuva 4. Kano-mallin yksinkertainen kuvaus. Y-akseli kuvaa asiakastytyväisyyden määrää ja X -akseli tyytyväisten asiakkaiden määrää. (Lager 2019, 96)	15
Kuva 5. AHP-menetelmän hierarkiarakenteen yleiskuvaus. (Soam, N, Bs, Balasani, S, Marwaha, Kumar & Agrawal 2023, 2)	16
Kuva 6. Parivertailumatriisin esimerkki. (Saaty 2008, 88)	16
Kuva 7. Matemaattisen mallin rakentamisen prosessi. (Humi 2017, 12)	18
Kuva 8. Matemaattisen mallin rakentamisen prosessi ja sitä ohjaavat kysymykset. (Dym 2004, 7)	19
Kuva 9. Optimoinnin yleistetty toimintajärjestys (Nayak 2020, 1).	21
Kuva 10. Optimoinnin vaiheet Nayakin mukaan (2020, 3)	22
Kuva 11. Optimointitekniikoita jaoteltuna perinteisiin ja edistyneisiin menetelmiin. (Nayak 2020, 6)	23
Kuva 12. Esimerkki epälineaarisista funktioista (Baker 2015, 271–272)	24
Kuva 13. Globaalin optimoinnin menetelmiä. (Khan, Thilagavathy, Priyadarsini & Arul 2024, 4)	25
Kuva 14. Yleiskuva alkuperäisestä Excel-optimaattorista	29
Kuva 15. Tuoteominaisuuskohtaiset määrityksen optimaattorissa.	29
Kuva 16. Lisäkustannusten kertymisen vaihtoehdot ja käyrät. (Kirkwood 1997, 65)	30
Kuva 17. Lisämyynnin ja hintajoustokertoimen määrittäminen.....	31
Kuva 18. Optimaattorin tulossarakkeet.....	32
Kuva 19. Excel Solver-lisäosan määritykset.....	32
Kuva 20. Excel Solver -optimointimentelmät.	33
Kuva 21. Esimerkkilaskennan tulos Excel Solver optimoinnilla.	33
Kuva 22. Lokaalin ja globaalin optimin ero GRG Nonlinear menetelmässä. (Engineerexcel 2024)	34
Kuva 23. Testilaskennoissa käytetyt lähtöarvot	35
Kuva 24. Testilaskentojen tulokset neljällä eri menetelmällä.	36
Kuva 25. Eri optimointimenetelmien testilaskentojen spesifikaatioiden ja kateprosentin arvot graafissa. Yksittäiset arvot ovat kuvattu ympyröillä, palkki esittää keskiarvon ja T-viiva keskihajonnan.	37
Kuva 26. Eri menetelmien testilaskentojen spesifikaatioiden arvojen keskiarvot, mediaanit ja moodit.	38

1 JOHDANTO

Tuotekehitysprosessissa asiakastarpeista johdettavat suunnitteluvaatimukset ja tekniset spesifikaatiot, eli tuotetta kuvaavat tekniset määrittelyt, muodostavat konkreettiset määrittelyt millainen tuoteesta tulee, mitä se tekee ja minkälaisia kustannuksia siihen kohdistuu (Ulrich & Eppinger 2016, 16–17). Tässä opinnäytetyössä tutkitaan ja kehitetään menetelmää teknologiateollisuuden tuotteiden suunnitteluvaatimusten tavoitearvojen optimoinnille. Tavoitearvot kuvaavat tasoa, johon yksittäiset suunnitteluvaatimukset pyritään kehittämään.

Työn tilaajana toimii Savonia-ammattikorkeakoulun päätoteuttama *Specsoft-teknologiateollisuuden tuotekonseptointisovelluksen kehittäminen*-hanke. Hankkeen rahoittajana toimii Pohjois-Savon maakuntaliitto ja rahoitusinstrumenttina Euroopan aluekehitysrahasto. Osarahoitajina mukana on neljä pohjoissavolaista teknologiateollisuuden yritystä, joiden kanssa yhteistyössä tuotekonseptointisovellusta kehitetään.

Specsoft-hankkeessa kehitettävä tuotekonseptointisovellus yhdistää neljä aiemmissa hankkeissa kehitettyä työkalua yhdeksi eheäksi kokonaisuudeksi. *CusTech-asiakaslähtöinen tuotteen tekninen määrittely*-hankkeessa on kehitetty erilaisia työkaluja asiakastarpeen määrittelyyn, sekä analyyttisellä hierarkiaprozessilla (AHP) että QFD (Quality function deployment)-menetelmillä (Rakennerahastotietopalvelu 2018). *BisTech asiakas- ja liiketoimintalähtöinen tuotekonsepti*-hankkeessa on kehitetty työkaluja, joilla asiakkaan ääni (voice of the customer, VoC) voidaan muuttaa tuotespesifikaatioiksi, sekä Excel-simulaattori näiden arvojen optimointiin (Rakennerahastotietopalvelu 2020). Vuosina 2023–2025 toteutettavassa *Specsoft*-hankkeessa kehitettävän tuotekonseptointisovelluksen avulla voidaan määrittellä tuoteominaisuuksia, suunnitteluvaatimuksia ja niiden hierarkiaa, ominaisuuksien tärkeyttä ja merkitystä asiakkaan näkökulmasta, sekä teknisten spesifikaatioiden tärkeyttä eri laatutekijöiden näkökulmista ja spesifikaatioiden optimaalisia arvoja eri liiketoimintatavoitteiden näkökulmista (Hanketietopalvelu 2024).

Tämä opinnäytetyö keskittyy *Specsoft*-hankkeessa kehitettävässä uudessa tuotekonseptointisovelluksessa tehtävään teknisten spesifikaatioiden optimaalisten arvojen määrittämiseen. Uuden tuotekonseptointisovelluksen yhdistämät eri menetelmät ja vaiheet voidaan nähdä erillisinä sovellusmoduuleina, jotka on integroitu toisiinsa yhtenäiseksi sovelluskokonaisuudeksi. Sovelluksen käyttäjän näkökulmasta teknisten spesifikaatioiden tavoitearvojen optimointi on viimeinen vaihe sovelluksen käyttöä. Ennen tätä sovelluksessa kuitenkin määritellään muun muassa kehityksessä olevan tuotteen ominaisuudet ja suunnitteluvaatimukset, sekä niiden hierarkia, merkitys ja tärkeys valituista näkökulmista. Näiden eri vaiheiden ja metodien kautta muodostuvat tekniset spesifikaatiot sekä niiden prioriteetit suhteessa toisiinsa. Teknisten spesifikaatioiden optimointi antaa mahdollisuuden arvioida erilaisten liiketoimintatavoitteiden toteutumista asetetuilla spesifikaatioiden arvoilla. Näitä erilaisia liiketoimintatavoitteita voivat olla esimerkiksi liikevaihdon, katteen tai kateprosentin maksimointi.

Tämän työn teoreettinen viitekehys muodostuu tuotekehitysprosessin konseptointivaiheesta ja matemaattisten mallien sekä optimointilaskennan perusteorioista. Nayakin (2020, 1) mukaan matemaattisen optimoinnin käyttö monilla eri aloilla on yleistynyt, ja se on oleellinen osa päätöksentekoon ongelmia esimerkiksi insinööri- ja taloustieteissä. Esimerkiksi lentokoneiden kehityksessä optimoinnin

kohteena tyypillisesti on lentokoneen paino (Nayak 2020, 1). Tässä työssä pysytään heuristisen lähestymisen mukaisesti matemaattisten mallien ja optimointilaskennan teorioissa verrattain yleisellä tasolla siten, että tarkasteluun on otettu vain tutkimuksen kohteen kannalta oleelliset ja tarpeelliset teoriat. Tuotekehitysprosessi sisältää monia laajoja aihealueita ja vaiheita kehitysprojektin suunnittelusta uuden tuotteen tuotannon lanseeraamiseen, joista tässä työssä on keskitytty tarkimmin konseptointivaiheeseen.

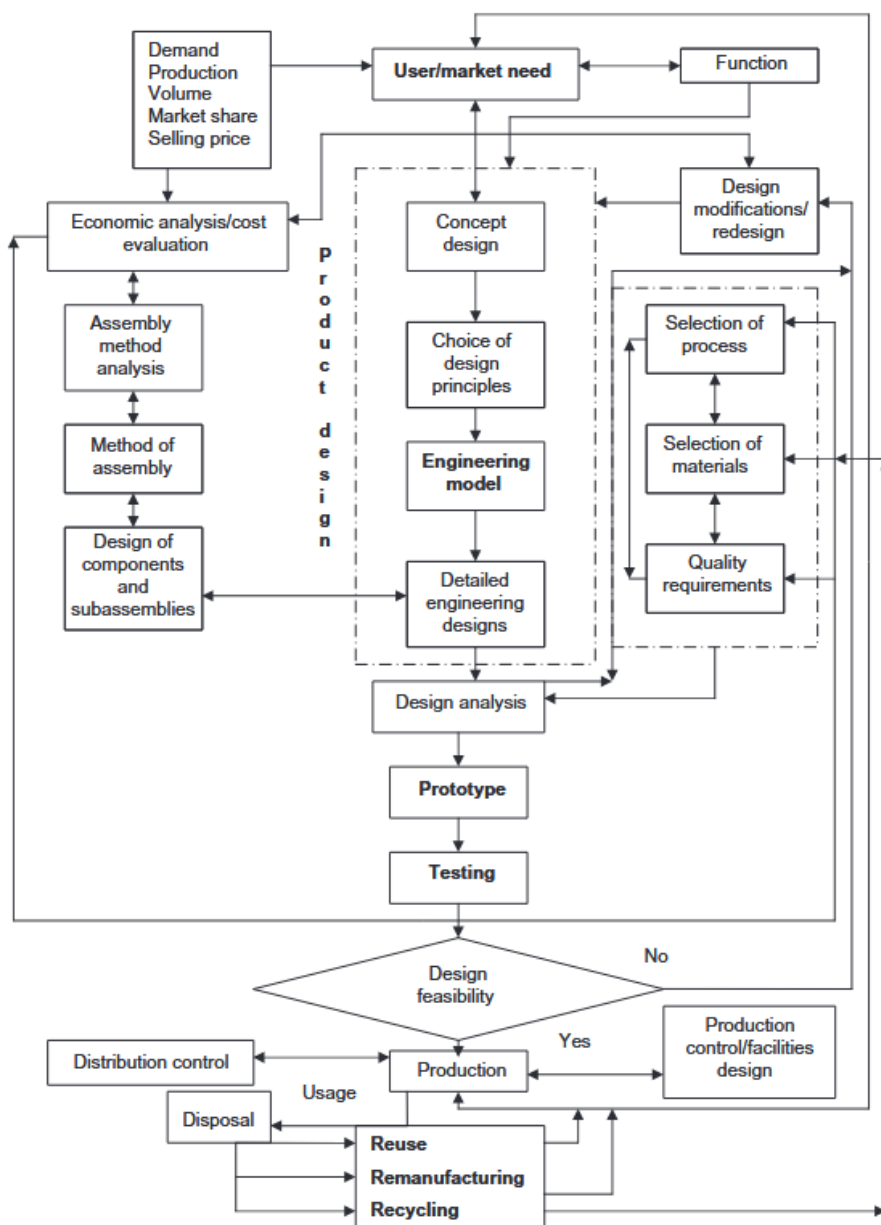
2 TUOTEKEHITYS

Menestyvien tuotteita valmistavien yritysten toiminta on jatkuvaa innovointia, jossa säännöllisesti esitellään uusia tai parannettuja tuotteita, perustuen asiakkaiden tarpeisiin ja toiveisiin. Kokonaisprosessia, joka kattaa näiden tuotteiden suunnittelun, valmistuksen ja myynnin, kutsutaan tuotekehitykseksi. (Mital & Desai 2014, 21). Yrityksien täytyy kehittää riittävän laadukkaita markkinoiden toivomia tuotteita nopeasti, taloudellisesti, helposti ja tehokkaasti. Tämän johdosta onnistuneella tuotekehitysprosessilla on seuraavat ominaisuudet:

1. **Kustannukset:** Tuotteen kustannushinta määrittää tuotteen myyntihintaa, joka vaikuttaa tuotteen kiinnostavuuteen markkinoilla. Tuotteen hinta sisältää sekä valmistuskustannukset että kehityskustannukset konseptista prototyyppiä. Tuote muuttuu kannattavaksi vasta, kun sen kehittämisen aiheuttamat kustannukset on katettu.
2. **Laatu:** Tuotteen laadun on täytettävä asiakkaiden tarpeet, jotta se voi menestyä markkinoilla. Laatu on suurin määräävä tekijä siinä, kuinka paljon asiakkaat ovat siitä valmiita maksamaan, joten se heijastuu myös tuotteen saamaan markkinaosuuteen.
3. **Tuotekehityksen kesto:** Ajallista kestoa tarpeiden arvioimisesta tuotteiden myyntiin voidaan käyttää tuotekehityksen onnistumisen mittarina, keston määrittäessä tuotekehitykseen panostetun sijoituksen kokonaistaloudellista tuottoa.
4. **Kehittämisen tietotaito:** Kyky toistaa onnistunut kehitysprosessi tuotteesta toiseen lyhentää tuotekehitysprosessin kestoa ja vähentää sen kustannuksia, tuotteet saavuttavat merkittävän markkinaosuuden ja tulevat kannattaviksi. (Mital & Desai 2014, 22–23)

Tuotannon ylös-ajo on viimeinen vaihe, jossa tuotteen varsinainen tuotantoprosessi testataan kokonaisuudessaan, siinä missä edellisessä vaiheessa suunniteltuja tuotantotapoja testattiin osissa. Vaiheen tarkoituksena on myös kouluttaa työntekijät tuotteen valmistusprosessiin ja sen yksityiskohtiin. (Ulrich & Eppinger 2016, 15)

Samat tuotekehitysprosessin piirteet ovat löydettävissä myös Mital ja Desain (2014, s. 17) kuvaamana yksityiskohtaisena prosessina (kuva 2.) Heidän mukaansa tuotteiden kilpailukykyinen tuotanto edellyttää hyvää ymmärrystä asiakkaiden tarpeista, markkinoista, käytettävissä olevien materiaalien ja prosessien tehokkaasta ja taloudellisesti hyödyntämisestä, sekä tuoteturvallisuudesta ja huollettavuudesta. Lisäksi, että mietitään miten tuote tulisi valmistaa, pitäisi keskittyä tuotteiden toiminnallisuuteen, jotta tuotteet vastaisivat markkinoiden sekä käyttäjien tarpeita, ja niitä voitaisiin koko elinkaaren ajan myydä taloudellisesti. Tuotekehitysprosessi sisältää siis merkittävän määrän erilaisia huomioitavia elementtejä. (Mital & Desai 2014, 16)



Kuva 2. Yksityiskohtainen tuotekehitysprosessin kuvaus vuokaaviona (Mital & Desai 2014, 17)

Kuvan 2 tuotekehitysprosessikaavio sisältää useita eri vaiheita ja niiden välisiä takaisinkytkentöjä. Vuokaaviossa nuolet osoittavat kuinka prosessi etenee ja miten prosessin vaiheet ovat toisistaan riippuvaisia. Prosessi on siis vahvasti vuorovaikutteinen, ja prosessin vaiheet voidaan jakaa kolmeentoista osa-alueeseen etenemisen mukaan. (Mital & Desai 2014, 17–18)

1. Markkinoiden ja käyttäjien tarpeet sekä toiminnallisuudet
2. Konseptikehitys ja suunnitteluperiaatteiden valinta
3. Materiaalien ja prosessien tunnistaminen
4. Suunnittelun ja prosessien analyysi, suunnitelmien muokkaaminen
5. Laatuvaatimukset
6. Kokoonpantavuuden ja purkamisen analysointi
7. Yksityiskohtainen suunnittelu
8. Taloudellinen analyysi ja tuotteen hinnan arviointi
9. Prototyypin kehitys
10. Testaus ja uudelleen suunnittelu
11. Suunnitelmien toteutettavuus
12. Tuotanto
13. Tuotannon ja jakelun valvonta

Vaikka tuotekehitysprosessi puretaan kuvan 2. mukaisesti tarkkoihin vaiheisiin, voidaan se nähdä yhdenlaisena ongelmanratkaisun muotona, jonka perustoimintoja ovat ongelman, tarpeiden, tavoitteiden ja kriteereiden määrittäminen, ratkaisuvaihtoehtojen konseptointi, tuottaminen sekä analysointi ja parhaan vaihtoehdon valinta ja tarkentaminen. (Mital & Desai 2014, 59–60)

2.2 Konseptointivaihe

Tuotekehitysprosessin alkupäähän sijoittuva konseptointivaihe on tärkeä suunniteltavan tai kehitettävän tuotteen kannalta, koska siinä määritetään, mitä ominaisuuksia tuotteessa tulee olla. Tässä vaiheessa ei vielä oteta kantaa siihen, miten nämä ominaisuudet toteutetaan. Ominaisuudet täyttävät prosessin aiemmassa vaiheessa määritetyt tarpeita tai ovat ratkaisu olemassa oleviin ongelmiin. (Mital & Desai 2014, 62–68)

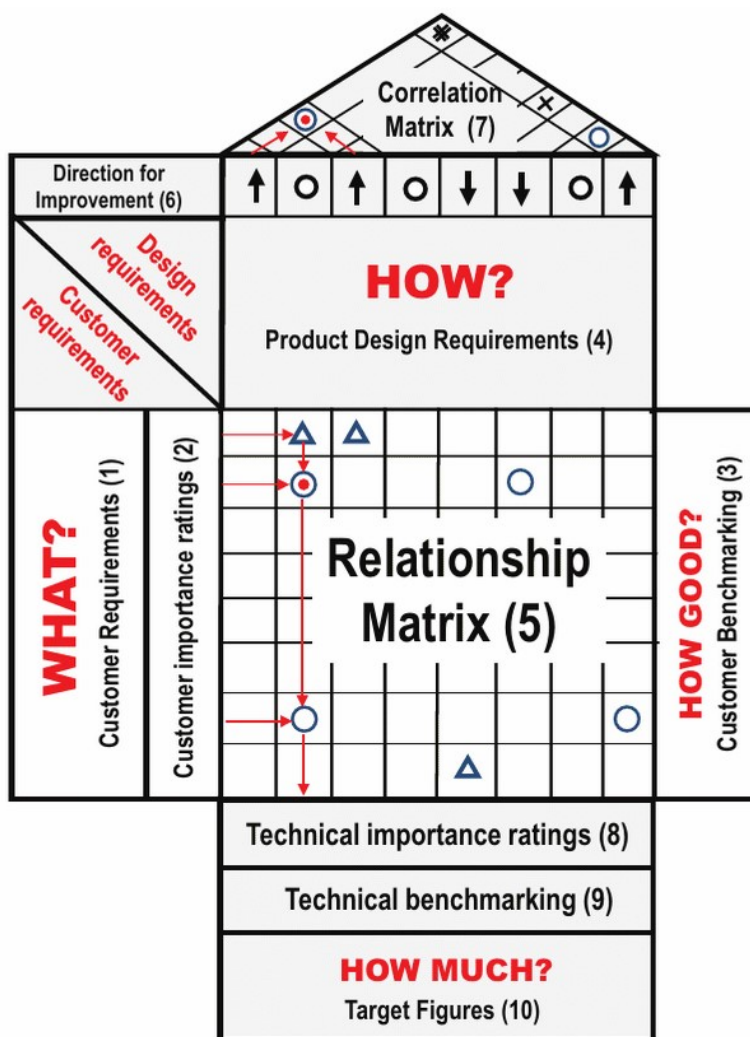
Konseptointivaiheessa tavoitellaan ytimekästä kuvausta siitä, miten tuote täyttää asiakkaan tarpeet. Konseptin laadukkuus määrittää pitkälti, kuinka hyvin lopullinen tuote täyttää asiakkaiden vaatimukset ja kaupallistuu. Tässä vaiheessa määräytyy myös arviolta 85% tuotteen kustannuksista, joten vaiheen tärkeyttä prosessissa ei voi vähätellä. (Rodgers & Milton 2011, 78)

Tuotekehittäjien käytössä on useita menetelmiä, joiden avulla tuotetaan erilaisia tuotekonsepteja vastaamaan asiakkaiden ja markkinoiden tarpeisiin (Rodgers & Milton 2011, 78). Konseptit voidaan nähdä jäsentelemättöminä karkeina suunnitelmina, joista valitaan parhaat vaihtoehdot vertailemalla niiden ominaisuuksia ja kyvykkyyttä täyttää tuotteelle asetetut vaatimukset, toiminnot ja rajapinnat. Konseptit voivat myös sisältää aiemmin kehitettyjä osia, ja konsepti toimii lähtökohtana yksityiskohtaiselle tuotesuunnittelulle. (Weiss 2013, 55–56)

Konseptointivaiheessa ominaisuuksille määritetään myös suorituskykyvaatimukset sekä niiden raja-arvot, eli arvot, jotka määrittävät vaihteluvälin hyväksyttävälle vaihtoehdoille. Suorituskykyvaatimukset eivät saisi olla riippuvaisia mistään tietystä ratkaisusta tai epämääräisiä laadullisia vaatimuksia. Suorituskykyvaatimusten kuului siis olla yleisluontoisia teknisiä ominaisuuksia, kuten esimerkiksi massa, pituus tai tilavuus, jotka saavat jonkin sallitun vaihteluvälin. (Mital & Desai 2014, 69–71)

2.3 QFD

Tuotekehityksen yhtenä päämääränä on ymmärtää paremmin asiakasvaatimuksia, joiden kautta muodostuvien teknisten spesifikaatioiden, laadun ja kustannusten hallintaa varten on kehitetty Japanilainen *Quality Function Deployment* -menetelmä (QFD). Menetelmän tavoitteena on yksinkertaistaa ja selkeyttää tiedonkulkua tuotekehitysprosessissa. Lähtökohtaisesti QFD -menetelmä voidaan kuvata päätöksenteon tukena käytettäväksi työkaluksi, jossa sarjalla erilaisia matriiseja kuvataan kehitettävän tuotteen osa-alueita ja niiden välisiä suhteita. Tarkoituksena on laskea ominaisuuksille numeerisia indikaattoreita, joiden avulla luodaan niin sanottu arvojärjestys, ja esitetään ne graafisesti helposti ymmärrettävässä muodossa. (Maritan 2015, 2–11)



Kuva 3. QFD-matriisin perusmuoto (Lager 2019, 12)

Kuvassa 3 on Thomas Lagerin (2019) esittämä yleiskuvaus QFD-matriisin perusmuodosta, jonka rakenne jakautuu seuraaviin osa-alueisiin.

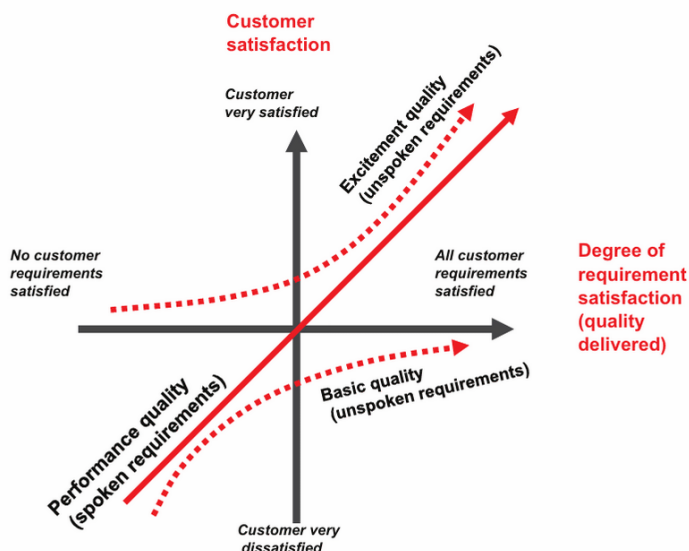
1. *Asiakasvaatimukset* kuvaavat asiakkaan ääntä (englanniksi ”Voice of the Customer”), ja joiden keräämistä varten täytyy määrittää, keitä asiakkaat ovat ja mitä he odottavat tuotteelta. Näiden selvittämiseen käytetään erillisiä menetelmiä, kuten esimerkiksi Kano-mallia, jolla tutkitaan asiakkaiden kertomia laatuvaatimuksia. Asiakkaiden kuvaamia vaatimuksia lajitellaan ja ryhmitellään pyrkien hallittavaan vaatimusten määrään.
2. *Asiakasvaatimusten tärkeysarvot* kuvaavat, kuinka tärkeää asiakkaalle on kunkin vaatimuksen täytyminen. Tämä luokittelu tulisi tehdä asiakkaiden toimesta, etteivät esimerkiksi tuotekehittäjien omat mielipiteet vääristä luokittelua.
3. *Kilpailuanalyysi* on tärkeä osa tuotekehitystä, koska ostopäätöstä tehdessään asiakkaat vertaavat markkinoiden vaihtoehtoja joko järjestelmällisesti tai vaistomaisesti. QFD-matriisin kilpailuanalyysissä asiakasta pyydetään arvioimaan tuotteen suorituskykyä verrattuna kilpailijoihin.
4. *Suunnitteluvaatimuksia* määritettäessä laadulliset asiakasvaatimukset muunnetaan teknisesti mitattaviksi määreiksi. Kuvan 3 mukaisesti asiakasvaatimukset vastaavat kysymykseen siitä, mitä tuotteelta vaaditaan. Vastaavasti suunnitteluvaatimukset vastaavat kysymykseen, miten vaaditut asiat täytetään. Matriisissa jokaista asiakasvaatimusta käsitellään erikseen, ja jokainen yksittäinen asiakasvaatimus voi liittyä useampaan suunnitteluvaatimukseen. Näin muodostuvat asiakas- ja suunnitteluvaatimusten riippuvuussuhteet, joiden tarkasteluun matriisi tarjoaa systemaattisen menetelmän.
5. *Riippuvuussuhdematriisiin* arvioidaan, miten vahva suhde yksittäisellä asiakasvaatimuksella on kuhunkin suunnitteluvaatimukseen. Vahva suhde saa numeerisen arvon 9, keskitasoinen suhde arvon 3, heikko arvon 1 ja mikäli minkäänlaista riippuvuussuhdetta ei ole, tulee arvoksi 0.
6. *Kehittämisen suunta* määritetään erikseen jokaiselle suunnitteluvaatimukselle riippuvuussuhteiden arvioinnin jälkeen, ja se kertoo, täytyykö suunnitteluvaatimusta muuttaa vai pyritäänkö se pitämään ennallaan.
7. *Korrelaatiomatriisi* kuvaa suunnitteluvaatimusten suhdetta toisiinsa, eli mikäli jotakin suunnitteluvaatimusta parannetaan, heikentääkö tai vahvistaako se jotain toista suunnitteluvaatimusta. Konkreettisenä esimerkkinä heikentävästä vaikutuksesta toimii materiaalin kovuuden kasvattaminen, joka usein vähentää materiaalin sitkeyttä. Suhteet kuvataan yleensä neljällä eri vaihtoehdolla, joita ovat vahva positiivinen, heikko positiivinen, vahva negatiivinen sekä heikko negatiivinen. Korrelaatiomatriisin avulla tuotekehityksessä voidaan hakea optimaalista kompromissia suunnitteluvaatimusten välillä.

8. *Tekniset tärkeysarvot* ovat keskeinen osa QFD-matriisia, koska tässä vaiheessa asiakasvaatimusten tärkeysarvot muunnetaan teknisiksi tärkeysarvoiksi. Teknisten tärkeysarvojen laskeminen tapahtuu ns. pystyiveittäin kertomalla jokainen yksittäinen asiakas- ja suunnitteluvaatimukselle määritetty suhteen arvo (0, 1, 3 tai 9) asiakasvaatimuksen tärkeysarvolla, ja laskemalla sitten suunnitteluvaatimuskohtaisesti näiden kertolaskujen summa. Tämä vaihe kuvastaa, kuinka tärkeä kukin suunnitteluvaatimus on asiakkaan arvion mukaan.
9. *Tekninen kilpailuanalyysi* on verrattavissa kohdan 3 asiakkaan suorittamaan kilpailuanalyysiin, mutta tässä vaiheessa arvioidaan sisäisesti tuotteen suunnitteluvaatimusten suorituskykyä verrattuna markkinoilla oleviin tuotteisiin.
10. *Tavoitearvojen asettaminen* tapahtuu viimeisessä osiossa, kun matriisin muihin osioihin on kerätty laajasti faktapohjaista tietoa tuotteesta. Asetettavat tavoiteluvut ohjaavat suunnitteluvaatimusten kehittämistä kehitettävissä tuotekonsepteissa, ja QFD-matriisi kokonaisuudessaan toimii tuotekehityksen tukityökaluna. (Lager 2019, 9–18)

Nämä QFD-matriisiin osioiden kuvaukset ovat koko QFD-menetelmän käytön alkupiste, jossa asiakasvaatimukseen vastataan asettamalla suunnitteluvaatimukset. Tämän jälkeen päästään rakentamaan matriisijärjestelmää, jossa laaditaan useampia samanmuotoisia matriiseja eri näkökulmista. Järjestyksessä seuraavaksi aloitetaan kuvaamaan tuoteominaisuuksien vaatimuksia, jota varten siirretään yläosaan asetetut suunnitteluvaatimukset seuraavan matriisiin asiakasvaatimusten kohtaan. Tämän jälkeen samalla siirtoperiaatteella voidaan laatia matriisit prosessisuunnittelua ja tuotannon suunnittelua varten. Tällä tavalla QFD-menetelmä palvelee siis tuotekehitysprosessia asiakasvaatimuksista tuotantoon asti. (Lager 2019, 18–20)

2.4 Kano

Kano-malli on markkinatutkimusmenetelmä, jolla kuvataan asiakkaiden ilmaisemia kolmenlaisia laatuvaatimuksia. Useimmiten asiakkaat puhuvat haastattelussa ainoastaan suorituskykyyn liittyvistä laatuvaatimuksista (Performance Quality Requirements), jotka eivät kuitenkaan ole ainoita vaatimuksia, jotka tuotteen täytyy täyttää. Niin sanottuja hiljaisia tai ei-puhuttuja vaatimuksia ovat peruslaatuvaatimukset (Basic Quality Requirements) sekä innostavat laatuvaatimukset (Excitement Quality Requirements). Näistä peruslaatuvaatimukset ovat sellaisia vaatimuksia, jotka tuotteen täytyy itseltään selvästi täyttää. Esimerkiksi se, että auton ikkuna ei saa vuotaa vettä, voi olla yksi peruslaatuvaatimus. Innostavat laatuvaatimukset ovat puolestaan sellaisia, jotka ovat yleensä täysin uusia, ja joita asiakas ei osaa odottaa. (Lager 2019, 95–96)



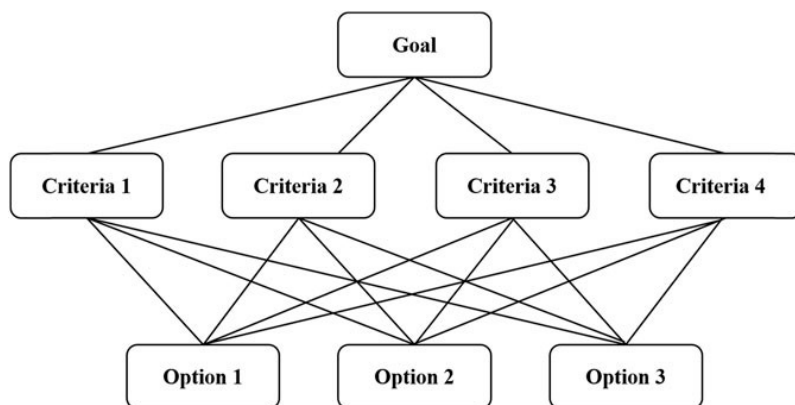
Kuva 4. Kano-mallin yksinkertainen kuvaus. Y-akseli kuvaa asiakastytyvyyden määrää ja X-akseli tyytyväisten asiakkaiden määrää. (Lager 2019, 96)

Peruslaatuvaatimukset ovat kuvan 4 mukaisesti sellaisia, että ne niiden täytyessä suurin osa asiakkaista on tyytyväisiä, mutta tyytyväisyyden taso jää keskinkertaiseksi. Jos nämä taas eivät täyty, ovat asiakkaat erittäin epätyytyväisiä. Suorituskykyyn liittyvät laatuvaatimukset puolestaan kasvattavat asiakastytyvyyden ja tyytyväisten asiakkaiden määrän suhdetta lineaarisesti, ja ne on myös helppo tunnistaa esimerkiksi asiakashaastatteluista. Innostavat laatuvaatimukset nostavat asiakastytyvyyden tasoa merkittävästi, mutta niiden tunnistaminen on vaikeaa, ja vaatii yleensä syvälistä asiakasymmärrystä ja luovaa työtä. (Lager 2019, 95–96) Kano-mallin avulla voidaan siis luokitella asiakasvaatimuksia eri luokkiin, ja tätä luokittelua hyödyntää QFD-matriisin täytössä. Tämä ohjaa vahvasti tuotekehitysprosessia, jotta sillä voidaan saavuttaa mahdollisimman korkea asiakastytyvyys pohjautuen asiakkaiden itse esittämiin vaatimuksiin. (Maritan 2015, 63–64)

2.5 Analyttinen hierarkiaproessi

Erilaiset kognitiiviset harhat vaikuttavat ihmisen päätöksentekoprosessiin. Tutkitusti yleisiä harhoja ovat esimerkiksi kehystysvaikutus sekä ankkurointi. Kehystysvaikutukseksi kutsutaan ilmiötä, jossa yksilön päätöksentekoon voidaan vaikuttaa siten, että esitetäänkö päätettävä asia positiivisessa vai negatiivisessa mielessä. Ihmiset esimerkiksi valitsevat enemmän sijoitusprojektin, jossa luvataan 80% mahdollisuus voittoon, kuin sijoitusprojektin, jossa kerrotaan olevan 20% mahdollisuus häviöön. Ankkurointi puolestaan on ilmiö, jossa ihmiset valitsevat todennäköisemmin heille ensiksi esitetyn vaihtoehdon. (Mu & Pereyra-Rojas 2018, 4)

Analyttinen hierarkiaproessi (AHP) on päätöksentekomenetelmä, joka muiden päätöksentekomenetelmien lailla mahdollistaa vaikeiden ja monimutkaisten ongelmien jäsentämisen. AHP-menetelmässä päätöksenteko jäsenetään hierarkkisesti kuvan 5 mukaisesti, ja siinä näytetään suhde tavoitteiden tai kriteerien sekä vaihtoehtojen välillä. Menetelmän etuna on, että päätöksentekoon saadaan sisällytettyä subjektiiviset kokemukset, mieltymykset ja intuitio loogisella tavalla. (Mu & Pereyra-Rojas 2018, 5)



Kuva 5. AHP-menetelmän hierarkiarakenteen yleiskuvaus. (Soam, N, Bs, Balasani, S, Marwaha, Kumar & Agrawal 2023, 2)

AHP-menetelmän kehittäjä Thomas L. Saaty (2008) jakaa menetelmän käytön seuraaviin neljään osaan:

1. Määrittele ongelma ja millaista tietoa haetaan.
2. Jäsennä päätöksenteon hierarkia siten, että ylimpänä on päätöksen tavoite kuvan 5 mukaisesti. Alemmilla tasoilla ovat kriteerit ja vaihtoehdot, jotka riippuvat toisistaan.
3. Muodosta parivertailumatriiseja, joissa jokaista kriteeriä verrataan jokaiseen vaihtoehtoon kuvan 6 mukaisesti skaalalla 1-9.
4. Muodosta vaihtoehtojen prioriteetit aiemman vaiheen parivertailun perusteella. Parivertailusta jokainen kriteeri saa oman prioriteetin, jolla painotetaan jokaisen vaihtoehdon prioriteettia. Tällä tavalla jokaiselle alimman tason vaihtoehdolle muodostuu globaali prioriteetti.

	<i>Flexibility</i>	<i>Opportunities</i>	<i>Security</i>	<i>Reputation</i>	<i>Salary</i>	<i>Priorities</i>
Flexibility	1	1/4	1/6	1/4	1/8	0.036
Opportunities	4	1	1/3	3	1/7	0.122
Security	6	3	1	4	1/2	0.262
Reputation	4	1/3	1/4	1	1/7	0.075
Salary	8	7	2	7	1	0.506

Kuva 6. Parivertailumatriisin esimerkki. (Saaty 2008, 88)

AHP-menetelmä on laajasti käytetty päätöksentekomenetelmä, joka sisältää prioriteettien määrittämisen lisäksi useita matemaattisia kaavoja ja lausekkeita, kuten ominaisvektori, painoarvot, Lambda-arvo ja konsistenssin (Soam, N, Bs, Balasani, S, Marwaha, Kumar & Agrawal 2023, 2). AHP-menetelmällä onkin paljon käyttöä eri aloilla, esimerkiksi julkishallinnossa, teollisuudessa ja lääketieteessä. Menetelmää on tutkittu laajasti viimeisinä vuosikymmeninä, ja sen avulla on saatu ratkaistua paljon käytännönläheisiä ongelmia. (Srebrenkoska, Apostolova, Dzidrov & Krstev 2023, 138)

3 MATEMAATTINEN MALLINNUS

Bender (1978, 2) määrittelee matemaattisen mallin olevan abstrakti, yksinkertaistettu matemaattinen esitys todellisuudesta, joka on laadittu jotakin tiettyä tarkoitusta varten. Benderin määritelmä on vain yksi monesta, esimerkiksi Humi Mayer (2017, 2) esittää matemaattisen mallin olevan symbolisella kielellä kuvattu todellisen elämän systeemi, jotta sen käyttäytymisestä ja kehittämisestä erilaisissa tilanteissa voidaan tehdä suunnilleen oikeita ennusteita. Dymen (Dym 2004, 4) mukaan mallit taas ovat matemaattisia esityksiä todellisista laitteista ja objekteista. Bakerin (Baker 2015, 2) määritelmässä matemaattiset mallit ovat yksinkertaistettuja esityksiä jostakin tilanteesta tai ongelmasta, jotka rakennetaan ja analysoidaan matematiikan avulla.

Matemaattiset mallit rakennetaan sisältämään ne asiat, jotka ovat relevantteja tutkittavan asian kannalta. Tärkein ja perimmäinen malleihin liittyvä kysymys on, kuinka hyvin ne suoriutuvat siitä ongelmasta, jota varten ne on luotu. Mallin sisältö, josta mallien suoriutuminen muodostuu, voidaan jakaa kolmeen osaan: 1. Asioihin, joita ei huomioida 2. Asioihin, jotka vaikuttavat malliin, mutta eivät ole tutkimuksen kohteita, sekä 3. Asioihin, sisältyvät malliin ja jotka ovat tutkimuksen kohteita. Ensimmäisen osan asiat eivät sisälly malliin, joten tutkittavan ongelman kannalta kriittisten asioiden pois-sulkemisen seurauksena mallilla ei voida tutkia ongelmaa halutulla tavalla. Toisaalta liian monien asioiden mukaan ottaminen tekee mallista liian monimutkaisen. Toisen osan asiat ovat mallin syötteitä (eng. inputs), eli kaavoja, funktioita ja vakioita ynnä muita tutkittavaan asiaan kiinteästi vaikuttavia tekijöitä. Kolmannen osan asiat ovat mallin tuloksia (eng. outputs), eli niitä asioita, joita halutaan tutkia ja selvittää. Kaikki kolme osa-alueita kytkeytyvät toisiinsa, ja niiden määrittely on tehtävä huolella, jotta matemaattinen malli suoriutuisi oikein siitä ongelmasta, jota varten se on luotu. (Bender 1978, 1–3)

3.1 Mallien rakentaminen

Matemaattisten mallien rakentaminen voidaan nähdä luovana toimintana ja prosessina, jossa eri vaiheet seuraavat toisiansa, toteutuvat limittäin tai muodostavat luupin, kunnes hyväksyttävä lopputulos saavutetaan. Mallien rakentamiselle ominaista on, ettei prosessille ole tiukkaan asetettuja sääntöjä. (Bender 1978, 6, Humi 2017, 4)

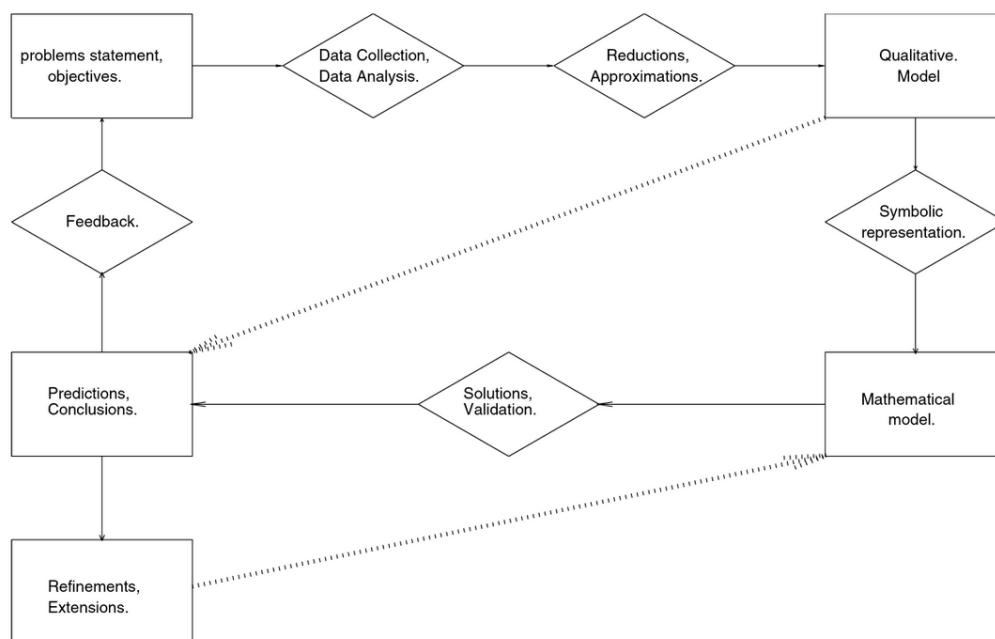
Bender esittää matemaattisen mallin rakentamisen nelivaiheisena prosessina, jossa prosessin vaiheet ovat yleisluontoisia (Bender 1978, 6–7);

1. Ongelman muodostus - Mitä halutaan tietää
2. Mallin hahmottelu – Jaetaan tutkittava asia osiin, joita ovat merkityksettömät asiat, vaikuttavat asiat eli syötteen, ja tulokset.
3. Onko malli käyttökelpoinen – Onko kaikki tarvittava data saatavilla ja voiko sitä käyttää haluttujen ennusteiden saamiseksi. Jos ei, palataan aiempiin vaiheisiin.
4. Mallin testaus – Käytetään mallia ja verrataan saatuja tuloksia olemassa olevaan dataan tai arki-järkeen. Jos tulokset eivät ole hyväksyttäviä, palataan aiempiin vaiheisiin.

Humi (2017, 4–12) puolestaan jakaa prosessin tarkemmin 13 vaiheeseen kuvan 7 mukaisesti;

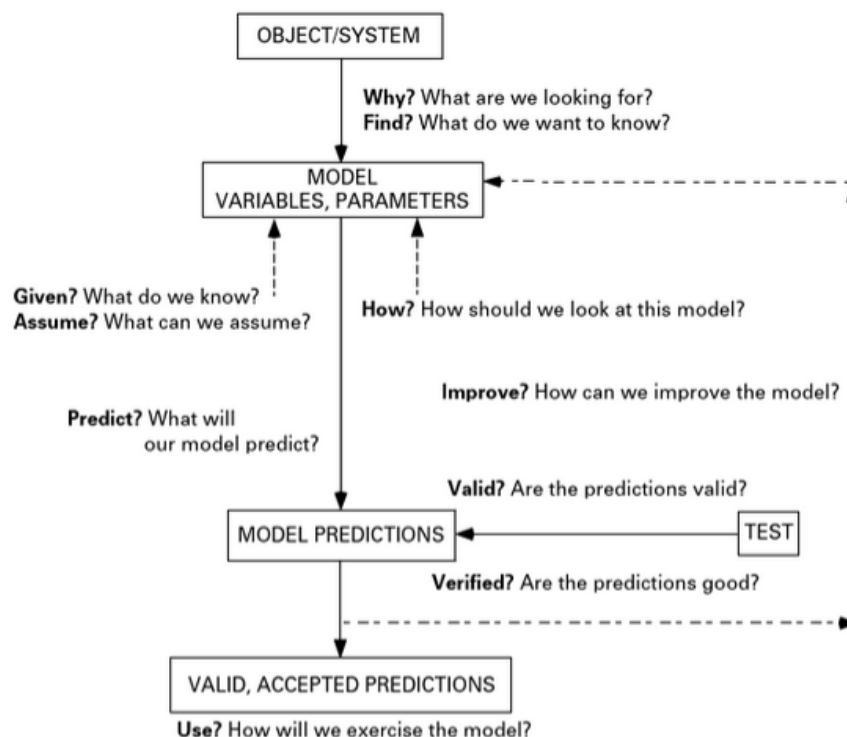
0. Mallin rakentamisen syyn ja päämäärien määrittely niin tarkasti, kuin on mahdollista.
 1. Ongelman tutkiminen todellisessa elämässä.
 2. Datan keräys ja analysointi todellisessa elämässä.

3. Kontrolloidut laboratorio testit tai simulaatiot.
4. Laadullisen ja käsitteellisen mallin rakentaminen tutkittavasta ilmiöstä.
5. Käsitteelliseen malliin perustuvien johtopäätöksien ja ennustusten tekeminen.
6. Abstraktio ja symbolinen esitys.
7. Ilmiötä ohjaavien yhtälöiden johtaminen.
8. Mallin testaus.
9. Mallin rajoitteiden tunnistaminen.
10. Ennusteiden tekeminen ja herkkyysoanalyysi.
11. Mallin jalostaminen.
12. Mallin laajentaminen uusiin käyttötarkoituksiin.



Kuva 7. Matemaattisen mallin rakentamisen prosessi. (Humi 2017, 12)

Toisaalta mallin rakentamisen toimintaperiaate voidaan hahmottaa myös kysymyksillä, kuten Dym (2004, 6–8) esittää kuvassa 8. Ongelman muotoilemisessa alustavat ideat ovat tärkeitä, ja näiden kysymysten voidaan olettaa toistuvan mallin rakentamisen aikana. Selkeä kuva siitä, miksi ja mitä varten matemaattinen malli tehdään, on avaintekijä ongelmanratkaisussa, jonka tueksi kysymykset luovat yleisen ajattelutavan matemaattisten mallien luomiseen. (Dym 2004, 7)



Kuva 8. Matemaattisen mallin rakentamisen prosessi ja sitä ohjaavat kysymykset. (Dym 2004, 7)

3.2 Tarkkuus ja luotettavuus

Matemaattisten mallien pyrkimys on tehdä todellisuutta kuvaavia ennusteita tutkittavasta asiasta tai ilmiöstä, mutta se ei tarkoita, että mallit olisivat absoluuttisen oikeassa. Virhemarginaali kuuluu matemaattisten mallien perusominaisuuksiin, ja usein arvioinnin kohteena on, onko saavutettu virhemarginaali hyväksyttävällä tasolla. Myös samaa asiaa tai ilmiötä jäljittelevät mallit voivat antaa eri tarkkuustason tuloksia. Tämä ei kuitenkaan tarkoita, että mallit olisivat lähtökohtaisesti epäkelvoja, vaan ne on rakennettu olemassa olevien rajoitusten mukaan. (Humi 2017, 2)

Mallien kelpoisuuteen tai oikeellisuuteen liittyen ilmeisin arvioinnin näkökulma on, pystyykö malli ennustamaan tutkittavaa kohdetta. Tähän vaikuttavat suoraan myös mallin rakentamisen alussa tehdyt valinnat liittyen mallin toimintaperiaatteeseen, oletuksiin ja matemaattisiin kaavoihin. Näillä alussa tehdyillä valinnoilla varmistetaan, että mallilla voidaan löytää vastauksia tutkittaviin asioihin. Oikeellisuutta voidaan varmistaa myös vertaamalla mallin tuloksia esimerkiksi kokeellisesti mitattuun dataan, jonka perusteella voidaan tehdä johtopäätös, toimiiko malli oikein vai ei. (Dym 2004, 88–89)

Lähtökohtaisesti matemaattiset mallit sisältävät siis virheitä eli laskennallisten arvojen eroja todellisiin ja täsmällisiin arvoihin. Virheet voidaan jakaa kahteen luokkaan, systemaattisiin ja satunnaisiin. Systemaattisia virheitä ovat toistuvasti ja johdonmukaisesti ilmenevät virheet. Yleisimmin näitä ilmenee esimerkiksi silloin, kun mittalaite ei ole oikein kalibroitu. Satunnaiset virheet ovat nimensä mukaisesti satunnaisia, eli ne ilmenevät odottamattomasti. Näitä tapahtuu yleensä käytännön kokeellisissa testeissä, koska testiasetelmissä on aina jonkin verran vaihtelua. Esimerkiksi laitteiden rikkoutumiset tai vikaantumiset ovat satunnaisia virheitä. (Dym 2004, 93)

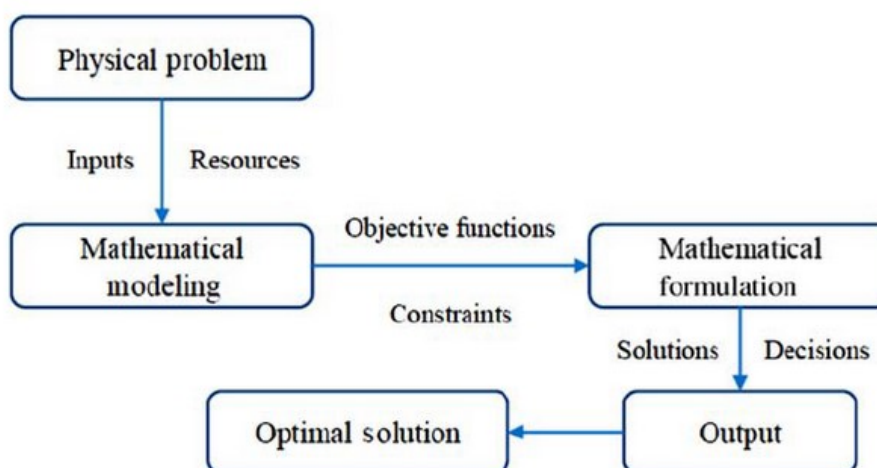
Virheiden vaikutusta voidaan arvioida tarkkuutta tarkastelemalla, joka kuvastaa miten suuria laskennallisten arvojen erot täsmällisiin ovat. Tarkkuutta kuvastetaan yleensä prosentuaalisella erolla, etenkin käytännönläheisissä testeissä. Tärkeä arvioitava ominaisuus on myös toistettavuus, eli kuinka lähellä toisiansa yksittäiset tulokset ovat. (Dym 2004, 94)

4 OPTIMOINTILASKENTA

Optimointi on prosessi, jonka tarkoituksena on löytää parhaat muuttujien arvot tietyillä kriteereillä. Optimointiongelma koostuu joukosta päätöksiä, joiden on täytettävä ongelmalle asetetut kriteerit, ehdot ja rajoitukset. Optimointiongelma voidaan kuvata matemaattisella mallilla, mutta kaikki matemaattiset ongelmat eivät kuitenkaan ole optimointimalleja. Optimointimallien tarkoitus on löytää halutulle muuttujalle paras arvo tai tulos, ja valinnat sen tuottamiseksi. Tällä tavalla optimointi auttaa yrityksiä päätöksenteossa, sekä parantamaan kannattavuutta ja kilpailukykyä (Baker 2015, 2). Optimoinnista onkin tullut suosittu ja olennainen työkalu päätöksenteko-ongelmien ratkaisuun esimerkiksi tekniikan aloilla ja taloustieteissä (Nayak 2020, 1).

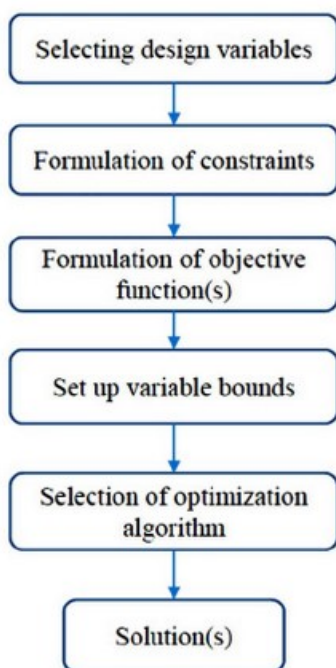
4.1 Optimoinnin vaiheet

Yleistetty toimintajärjestys optimoinnille on esitetty kuvassa 9, jossa optimointiongelma on nimitetty fyysisiksi ongelmaksi. Fyysisestä ongelmasta muodostetaan matemaattinen malli, jolle määritetään funktiot ja raja-arvot. Optimointitekniikoilla saatuja tuloksia tutkitaan "output" vaiheessa, joista muodostuu lopulta optimaalinen ratkaisu ongelmaan. (Nayak 2020, 1)



Kuva 9. Optimoinnin yleistetty toimintajärjestys (Nayak 2020, 1).

Yleistetty prosessi optimointiongelman ratkaisemiseen pilkkoutuu tarkempiin vaiheisiin kuvassa 10. Tämä tarkempi prosessi kuvastaa optimointilaskennan vaiheistusta ja siinä tarvittavia osatekijöitä taustalla olevan tarkoituksen, eli jonkin määritetyn optimointiongelman ratkaisemista varten. (Nayak 2020, 2)



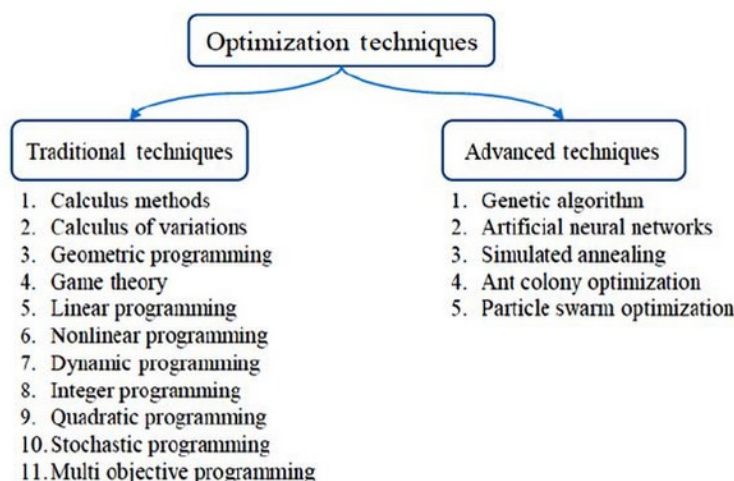
Kuva 10. Optimoinnin vaiheet Nayakin mukaan (2020, 3)

Selecting design variables eli suunnittelumuuttujien valinta on prosessin ensimmäinen vaihe, josta optimointilaskennan muodostaminen alkaa. Suunnittelumuuttujat ovat keskeinen osa optimointilaskentaa, koska ne vaihtelevat laskennan aikana ja ovat erittäin herkkiä sen toiminnalle. Muut laskennassa käytettävät muuttujat ovat joko kiinteitä tai vaihtelevat suunnittelumuuttujien suhteen. Suunnittelumuuttujat riippuvat ratkaistavasta ongelmasta, ja ovat käyttäjän määräämiä, joten niiden muodostamiselle ei ole tarkkoja sääntöjä olemassa. (Nayak 2020, 2–3)

Seuraava vaihe on *formulation of constraints*, joka tarkoittaa optimointiongelman rajoitusten tunnistamista. Tässä vaiheessa määritetään suunnittelumuuttujien ja muiden muuttujien sekä parametrien väliset riippuvuudet. Erilaiset matemaattiset kaavat tai fyysiseen maailmaan liittyvät rajoitteet laaditaan tässä vaiheessa. (Nayak 2020, 4)

Kolmannessa vaiheessa *formulation of objective function(s)* määritetään tavoitefunktio. Kyseessä on se mitä ollaan optimoimassa, ja tyypillisesti kyseessä on jonkin asian minimointi tai maksimointi. Näitä funktioita voi olla myös useita, koska monesti halutaan optimoida useampi asia yhtä aikaa. (Nayak 2020, 4)

Set up variable bounds vaiheessa määritetään suunnittelumuuttujien raja-arvot. Käytännössä kyseessä on suunnittelumuuttujien arvojen minimi ja maksimi väli, jolle optimoitavien arvojen on laskennassa osuttava. (Nayak 2020, 5)



Kuva 11. Optimointitekniikoita jaoteltuna perinteisiin ja edistyneisiin menetelmiin. (Nayak 2020, 6)

Viimeisenä vaiheena on *selection of optimization algorithm*, eli optimointialgoritmin tai tekniikan valinta, jota seuraa vielä *solution(s)* eli ratkaisut. Algoritmin ja tekniikan valinta riippuu täysin optimoitavan ongelmasta. Siihen on erittäin paljon vaihtoehtoja kuvan 11 mukaisesti, kuten muun muassa perinteisinä menetelminä lineaarinen ja epälineaarinen ohjelmointi sekä kehittyneempinä menetelminä geneettiset algoritmit ja neuroverkot. (Nayak 2020, 6)

4.2 Lineaarinen optimointi

Lineaaristen funktioiden avulla voidaan optimoida hyvin laajasti erilaisia tapauksia, ja niitä on käytetty paljon siitä asti, kun tietokoneet alkoivat yleistyä. Tyypillisesti alkuaikojen lineaariset optimointimallit liittyivät toiminnanohjauksen tapauksiin, kuten esimerkiksi kuljetus- ja jakelusuunnitteluun, tuotannon aikataulutukseen ja hankintastrategioihin. Myöhemmin lineaariset mallit yleistyivät muillekin osa-alueille kuten investointeihin, mainoskuluihin ja resursointiin. (Baker 2015, 21)

Lineaarinen optimointi käsittää lineaarisesti käyttäytyvät tavoitefunktiot ja rajoitukset sekä niiden väliset suhteet. Lineaariset menetelmät ovat varsin tehokkaita ratkaista, niiden tulkinta on helppoa ja usein onkin relevanttia olettaa ongelman olevan lineaarinen. Myös silloin, jos ongelma on selvästi epälineaarinen, voi lineaarinen ratkaisu toimia ongelman realistisena ratkaisuna. (Baker 2015, 21–23)

Yksinkertainen lineaarinen tavoitefunktio (Nayak 2020, 10):

$$Z = 30x_1 + 40x_2 \quad (1)$$

Lineaarisen optimointimallin rajoitukset ovat siis myös lineaarisia, joiden suhteet ovat joko pienempi kuin, yhtä suuri tai suurempi kuin. Rajoituksista muodostuu funktion alue, jossa optimaalinen ratkaisu esiintyy. (Baker 2015, 24)

Esimerkki kaavan 1. yksinkertaisen lineaarisen tavoitefunktion rajoitteista (Nayak 2020, 11):

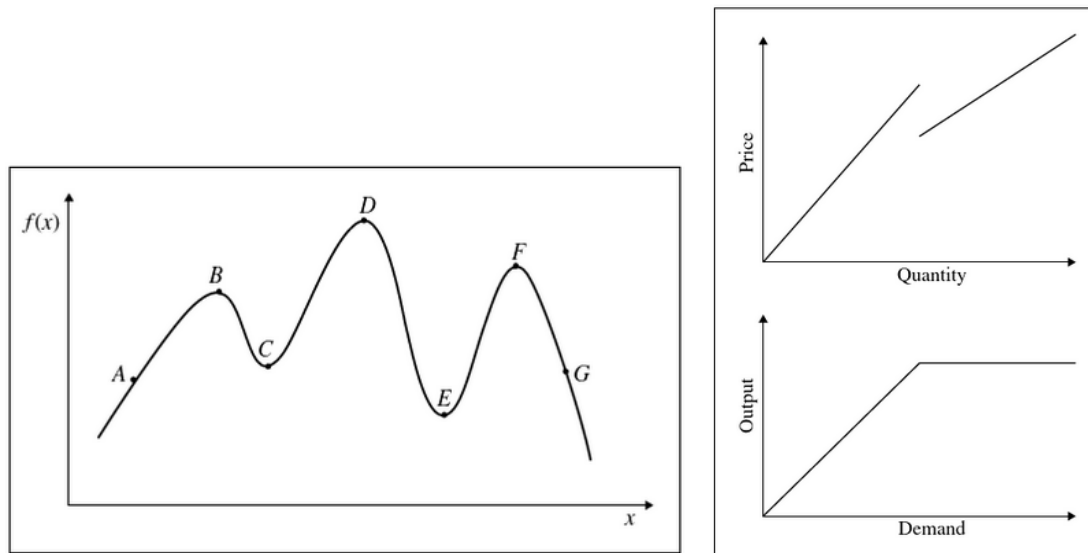
$$4x_1 + 2x_2 \leq 80, \quad (2)$$

$$2x_1 + 5x_2 \leq 180,$$

$$x_1, x_2 \geq 0$$

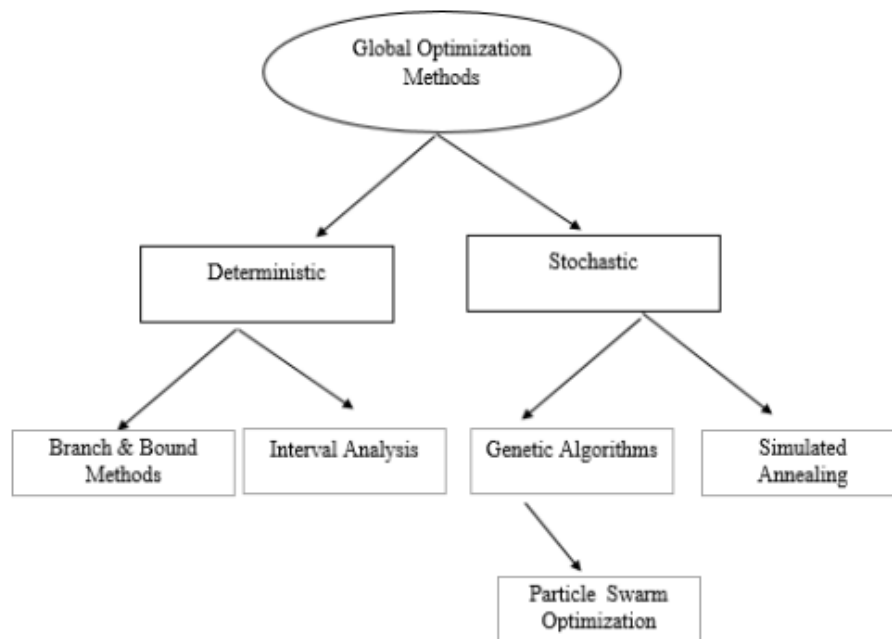
4.3 Epälineaarinen optimointi

Epälineaarinen optimointi tarkoittaa käytännössä kaikkia sellaisia optimointitapauksia, jotka eivät ole lineaarisia. Matemaattisesti lineaariset optimointimallit ovat epälineaarisen mallin osio. Epälineaariset funktiot voivat olla kuvan 12 mukaisesti jouhevia tai epäjatkuvia, jolloin ne sisältävät aukkoja tai teräviä kulmia. (Baker 2015, 270–271)



Kuva 12. Esimerkki epälineaarisista funktioista (Baker 2015, 271–272)

Toisin kuin lineaarisessa optimoinnissa, jossa muuttujien väliset suhteet ovat lineaarisia, epälineaarisissa optimointiongelmassa tavoite- ja rajoitefunktioit ovat epälineaarisia. Epälineaarisuuden vuoksi funktiot ja niiden väliset vuorovaikutukset ovat monimutkaisia, joka voi johtaa useisiin paikallisiin optimiratkaisuihin. Tätä varten on kehitetty globaalien optimoinnin menetelmiä, joiden avulla on mahdollista löytää paras ratkaisu kaikkien paikallisten optimien joukosta. Näitä ongelmia on laajasti eri aloilla, kuten esimerkiksi tekniikassa, taloustieteessä ja biologiassa. (Khan, Thilagavathy, Priyadarsini & Arul 2024, 2)



Kuva 13. Globaalin optimoinnin menetelmiä. (Khan, Thilagavathy, Priyadarsini & Arul 2024, 4)

Globaalin optimoinnin menetelmät voidaan jakaa deterministisiin ja stokastisiin menetelmiin kuvan 13 mukaisesti. Deterministiset menetelmät tunnetaan järjestelmällisestä lähestymistavasta, mikä takaa globaalin optimin löytymisen, mutta ne ovat laskennallisesti yleensä raskaita. Stokastiset menetelmät puolestaan hyödyntävät todennäköisyyden sääntöjä, mikä tekee niistä joustavampia ja tehokkaampia etenkin suuremmissa ongelmakohteissa. Stokastiset menetelmät eivät kuitenkaan takaa globaalin optimin löytymistä. Näitä menetelmiä voidaan myös yhdistää niin kutsutuiksi hybridimenetelmiksi, jotka yhdistävät molempien lähestymistapojen vahvuudet. Hybridimenetelmät tarjoavat ratkaisuja, jotka parantavat optimoinnin luotettavuutta ja tehokkuutta esimerkiksi käyttäen stokastista menetelmää haettavan alueen laajempaan tutkimiseen, ja determinististä menetelmää lupaavien alueiden tarkempaan analysointiin. (Khan, Thilagavathy, Priyadarsini & Arul 2024, 5)

5 TYÖN TARKOITUS JA TAVOITE

5.1 Tarkoitus

Työ on osa Savonia-ammattikorkeakoulu Oy:n *Specsoft*-teknologiateollisuuden tuotekonseptointisovelluksen kehittäminen-hanketta. Työn tarkoituksena on kehittää aiemmassa *Asiakas- ja liiketoimintalähtöinen tuotekonsepti*, *BisTech*-hankkeessa kehitetyn tuoteominaisuuksien spesifikaatioiden määrittämisvälineen optimointimenetelmää siten, että se on siirrettävissä *Specsoft*-hankkeessa tuotettavaan verkkopohjaiseen tuotekehityksen sovellukseen.

5.2 Tavoite

Työn tavoitteena on kehittää aiemman Excel-sovelluksen pohjalta ohjelmoitava optimaattori, jolla voidaan vertailla eri optimointimenetelmiä aiemmissa hankkeissa käytetyllä testiaineistolla.

5.3 Tutkimuskysymykset

Aiemmassa *BisTech*-hankkeessa kehitetyllä Excel-simulaattorilla on pystytty tekemään teknisten spesifikaatioiden optimointi, mutta epäselväksi on jäänyt Excelissä käytettävissä olevien optimointimenetelmien toimivuus ja luotettavuus, koska vertailua ei ole pystytty aiemmin tekemään. Tästä ongelmasta muodostuu tämän työn tutkimuskysymykset - mikä on soveltuva optimointimenetelmä teknisten spesifikaatioiden tavoitearvojen määrittämisessä hankkeessa kehitettävään sovelluksessa ja mitä eroja menetelmien käytössä ilmenee.

6 MENETELMÄLLINEN TOTEUTUS

6.1 Työn toteutus

Suunnitteluvaatimusten tavoitearvojen optimointiin kehitettävä ohjelmoitava optimaattorin tekeminen aloitettiin perehtymällä aiemmissa hankkeissa tuotettuun tuotekehityksen konseptointivaiheen apu-työkaluihin, ja niissä sovellettujen menetelmien periaatteisiin. Näitä menetelmiä olivat pääasiassa Kano-malli, analyttinen hierarkia prosessi sekä QFD-menetelmä. Työn kannalta yksi tärkeimmistä asioista oli aiemmin tuotettuun Excel-optimaattoriin, sen rakenteeseen, laskentaperiaatteisiin ja optimointimenelmiin perehtyminen.

Aiempiin tuotoksiin perehtymisen jälkeen työssä päästiin aloittamaan varsinaisen ohjelmoitavan optimaattorin tekeminen Python ympäristöön. Python-ympäristöön optimaattori koodattiin samoilla laskentaperiaatteilla mitä aiemmissa hankkeissa oli käytetty Excel-optimaattorissa. Ohjelmoitavan optimaattorin koodauksessa käytettiin apuna ChatGPT 3.5 tekoälyä, jonka avulla kartoitettiin ohjelmointimenetelmien vaihtoehtoja sekä tuotettiin koodin osia.

Ohjelmoitavan optimaattorin koodauksen jälkeen seuraavaksi tehtiin testilaskennat. Testilaskentojen aineistona käytettiin aiemmissa hankkeissa käytettyjä tuoteominaisuuksia ja niiden lukuarvoja, jotka liittyivät sähkömoottoripyörään. Testilaskennat tehtiin Python-ympäristöön koodatulla optimaattorilla kolmella eri optimointimenetelmällä ja Excel-optimaattorilla. Lopuksi testilaskennat taulukoitiin ja tuloksia analysoitiin tilastollisilla menetelmillä.

6.2 Tutkimusmenetelmät

Tämän työn ensisijainen tutkimusmenetelmä on määrällinen tutkimus, koska kerättyä aineistoa on käsitelty pääosin tilastollisten analyysien avulla. Aineiston keruu on toteutettu osittain laadullisilla menetelmillä, tukeutuen aiempien hankkeiden dokumentaatioon ja tuloksiin. Lisäksi työn toimeksiantaja on antanut suullisesti omia empiirisiä havaintojaan ja kokemuksiaan työn kohteeseen liittyen. Työ voidaankin luokitella tapaustutkimukseksi, koska se voidaan nähdä tutkimustapana, jossa voidaan käyttää useita eri menetelmiä (Laine, Bamberg & Jokinen 2015, 9). Tapaustutkimuksessa pyritään luomaan laaja kuvaus ainutkertaisesta tapahtumasta, jolloin aineistoa on yleensä useasta eri lähteestä (Tietoarkisto 2024). Tämä teki tapaustutkimuksesta soveltuvan tutkimusmenetelmän tähän työhön, koska toimivan optimointimenetelmän arviointi vaatii eri optimointi algoritmeilla tuotettujen tulosten vertailua.

Määrällisiin menetelmiin liittyy numeeriset laskelmat ja mittaukset, joista muodostuu kahdenlaisia tilastoja, joko kuvailevia tai päätelmällisiä. Kuvailevat tilastot ovat useimmiten tilastollisia tunnuslukuja, jotka kiteyttävät aineiston trendejä. Päätelmälliset tilastot puolestaan kuvaavat aineiston ominaisuuksia, joiden perusteella voi laatia merkittäviä johtopäätöksiä. (Case study research methods, s.2-3). Työn määrällinen aineisto koostui yhtäläisestä testiaineistosta, jota käytettiin erilaisten optimointialgoritmien vertailuun. Määrälliset menetelmät mahdollistivat objektiivisen tavan arvioida eri algoritmien tehokkuutta ja kuvata niiden toimintaa numeroiden kautta.


Tapaustutkimuksesta on hyvä erottaa mikä on tapaus ja mikä tutkimuksen kohde (Laine, Bamberg & Jokinen 2015, 10). Tässä työssä tapauksena on tuotekonseptointisovelluksen suunnitteluvaatimusten tavoitearvojen määrittäminen, ja tutkimuksen kohteena optimaattorimoduulin matemaattinen malli ja optimointimenetelmät. Heuristinen lähestymistapa on erityisen sopiva tutkimuksiin, joissa yritetään muodostaa yleisluontoinen näkemys kohteesta, keskittyen siihen olennaisesti liittyviin seikkoihin (Laine, Bamberg & Jokinen 2015, 179). Tämä lähestymistapa mahdollisti sen, että tutkimuksissa voitiin keskittyä tuotekonseptointisovelluksen optimaattorimoduulin toimintaan olennaisesti vaikuttaviin asioihin. Rinnalla deduktiivinen tutkimusote, eli teorian testaaminen empiirisesti, on hyvä tapa varmistaa matemaattisen mallin ja optimointimenetelmien soveltuvuus käytännön testilaskelmilla (Juhila 2021).

6.3 Työn lähtökohta

Työn lähtökohtana ovat aiemmissa *CusTech*- ja *BisTech*-hankkeissa kehitetyt eri menetelmiin pohjautuvat työkalut, joista tälle työlle erityisen tärkeä on *BisTech*-hankkeessa kehitetty tuoteominaisuuksien spesifikaatioiden määrittäminen työkalu. *Specsoft*-hankkeessa tätä työkalua kehitetään eteenpäin verkkopohjaiseksi sovellukseksi, joka sisältää useita eri tuotekehityksen konseptivaihetta tukevia menetelmiä. Sovelluksessa huomioidaan tuotekehityksen mahdollisuudet, tuoteominaisuudet ja -spesifikaatiot, niiden hierarkia, tärkeys ja merkitys asiakkaalle. Konseptointisovellus tukee yrityksissä tehtävää tuotekehitystä sekä tuotekehityksen opetusta.

Uusien tuotteiden kehittäminen siten, että tuoteominaisuudet vastaavat asiakkaiden tarpeita, on haastavaa ja siihen liittyy korkea epäonnistumisen riski. Epäonnistuminen johtaa myynnin heikkeneemiseen ja asiakaslupauksien pettämiseen. Tuotekehityksen konseptointivaihe sisältää useita menetelmiä, joita ei systemaattisesti hyödynnetä. Tämä puolestaan johtaa tuotekehitystä ohjaavan tiedon vinoumaan, toteutuksen ja asiakastarpeiden epäsuhtaan ja väärinymmärryksiin eri osastojen kesken.

Aiemmin *BisTech*-hankkeessa kehitetty Excel-pohjainen tuoteominaisuuksien spesifikaatioiden tavoitearvojen määrittämis työkalu hyödyntää konseptointivaiheen aiempia vaiheita ja metodeja. Näitä ovat optimointiin tuotavien tuoteominaisuuksien valinta AHP-menetelmällä ja tuoteominaisuuksien prioriteettiarvojen määrittäminen QFD:lla. Nämä muodostavat perustan optimoinnille, määrittäen mitä optimoidaan ja mikä optimoitavien tuoteominaisuuksien keskinäinen tärkeys on.


SPECISOFT OPTIMATOR

Spesifikaatioiden määritykset

Prioriteetti	Suhteellinen prioriteetti	Spesifikaatio	Mittayksikkö	Minimi	Tavoite	Maksimi	Maksimi-kustannus	Roo / kustannusten tyyppi	Lisäkustannus	Lisämyyntikerroin
324	0,329	Latausaika 0 - 80%	min	38	32,0	25	600 €	5	111,7 €	15,2 %
260	0,264	Massa	kg	157	140,0	130	800 €	5	105,1 €	16,6 %
202	0,205	Painopisteen korkeus	cm	44	40,0	40	450 €	5	450,0 €	20,5 %
199	0,202	Kiihtyvyys 0 - 100 km/h	s	6,7	6,0	5	600 €	1	136,0 €	8,3 %
Yht.	985	1					2 450 €		803 €	60,6 %

	Yht.	Tavoite
Lisämyynti, kun spesifikaatioiden arvot maksimissa	150	350
Hintajousto-kerroin	-2,00	
Hinnan ja myyntimäärän suhde (arc elasticity) E	2,45	

	Min - Max		Optimoitu tulos	
	Minimi	Maksimi	Tulos	Muutos
Myyntihinta	12 000,0 €	15 000,0 €	13 959,11	+1959,11
Valmistuskustannus	7 000,0 €	9 450,0 €	7 802,81	+802,81
Myyntihinnan hintajoustovalkutus [kpl]		-80	-56	
Lisämyynti (tuoteparannus) [kpl]		150	91	
Lisämyynti (muut toimenpiteet) [kpl]		0	0	
Myynti [kpl]	200	270	235	+35
Kokonaiskustannukset	1400 000 €	2 551 500 €	1 832 122 €	+ 432 122,11
Liikevaihto	2 400 000 €	4 050 000 €	3 277 646 €	+ 877 646,01
Kate %	41,7 %	37,0 %	44,3 %	3 %
Kate I	1 000 000 €	1 498 500 €	1 445 524 €	+ 445 523,91

Kuva 14. Yleiskuva alkuperäisestä Excel-optimaattorista

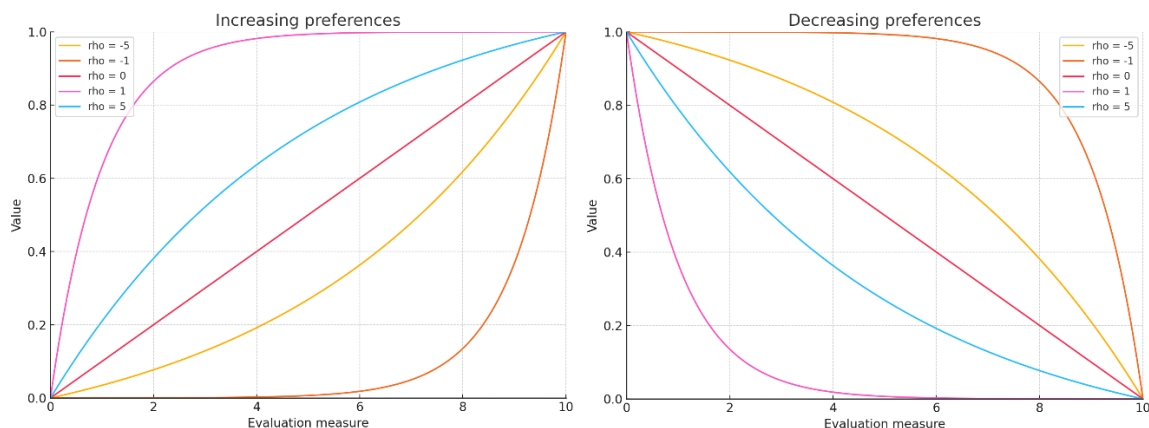
Tuoteominaisuuksien nimet, yksiköt ja prioriteetit tulevat optimaattorin yläosaan, muiden tuoteominaisuuskohtaisten määritysten kanssa. Suhteellinen prioriteetti määräytyy QFD:sta tulleiden prioriteettien kautta, ja se kuvastaa mikä kunkin tuoteominaisuuden prioriteetti on suhteessa muihin optimointiin valittuihin ominaisuuksiin. Kuvassa 15 näkyvät keltaiset sarakkeet ovat muokattavia sarakkeita, joihin käyttäjä asettaa arvot. Vasemmalta lukien ensimmäiset kolme saraketta määrittävät tuoteominaisuuden minimi, maksimi ja tavoitearvot. Näistä minimiarvo kuvastaa mitä kyseinen tuoteominaisuus vähintään on, tai se voi olla myös ominaisuuden nykytila. Maksimi -sarake kuvastaa arvoa, johon kyseinen ominaisuus voidaan enintään kehittää. Tavoitearvo puolestaan on se arvo, johon tuotekehityksellä pyritään.

Spesifikaatioiden määritykset

Prioriteetti	Suhteellinen prioriteetti	Spesifikaatio	Mittayksikkö	Minimi	Tavoite	Maksimi	Maksimi-kustannus	Roo / kustannusten tyyppi	Lisäkustannus	Lisämyyntikerroin
324	0,329	Latausaika 0 - 80%	min	38	32,0	25	600 €	5	111,7 €	15,2 %
260	0,264	Massa	kg	157	140,0	130	800 €	5	105,1 €	16,6 %
202	0,205	Painopisteen korkeus	cm	44	40,0	40	450 €	5	450,0 €	20,5 %
199	0,202	Kiihtyvyys 0 - 100 km/h	s	6,7	6,0	5	600 €	1	136,0 €	8,3 %
Yht.	985	1					2 450 €		803 €	60,6 %

Kuva 15. Tuoteominaisuuskohtaiset määrityksen optimaattorissa.

Maksimikustannus sarake kuvastaa kuinka paljon tuoteominaisuuden kehittämisen maksimiarvoon aiheuttaa tuotekohtaista lisäkustannusta. Viimeinen käyttäjän määrittämän sarake "Roo / kustannusten tyyppi" sitoutuu maksimikustannusten kertymiseen, joka lasketaan soluun "Lisäkustannus". Lisäkustannuksen kertymisen kaava perustuu Kirkwoodin (1997) esittämään kaavaan, jossa kustannukset voivat kertyä tilanteesta riippuen eri tavoilla eri käyrien mukaan kuvan 16. mukaisesti. Optimaattorissa lisäkustannuksen kertyminen valitaan määrittämällä tuoteominaisuuteen sopiva roon arvo, jonka perusteella optimaattori laskee tavoitearvon mukaisen lisäkustannuksen kaavan 3 mukaisesti.



Kuva 16. Lisäkustannusten kertymisen vaihtoehdot ja käyrät. (Kirkwood 1997, 65)

$$v(x) = \frac{1 - \exp\left[-\frac{x - Low}{\rho}\right]}{1 - \exp\left[-\frac{High - Low}{\rho}\right]} \quad (3)$$

$x = Tavoitearvo$

$Low = Minimiarvo$

$High = Maksimiarvo$

$\rho = Roon arvo$

Optimaattorin yläosan viimeisenä sarakkeen on lisämyyntikerroin, joka muodostaa kaavan 4 mukaisesti tuoteominaisuuskohtaisen kertoimen, kuinka paljon kyseisen ominaisuuden kehitys tavoitearvoon aiheuttaa lisämyyntiä. Kerroin muodostuu suhteellisen prioriteetin sekä tuoteominaisuuden minimi, maksimi ja tavoitearvojen kautta. Tuoteominaisuuksien lisämyyntikertoimien summa kuvastaa kuinka paljon optimointiin valittujen tuoteominaisuuksien kehittämisen tavoitearvoihin aiheuttaa lisämyyntiä.

$$\text{Lisämyyntikerroin} = \text{Suht. prioriteetti} \times \frac{(Tavoite - Minimi)}{(Maksimi - Minimi)} \quad (4)$$

Seuraava vaihe optimaattorissa on lisämyynnin määrittäminen kuvan 17 yläosan keltaiseen sarakkeeseen. Tämä on optimaattorin käyttäjän arvio kuinka paljon määrällistä lisämyyntiä voidaan saada, kun optimointiin valitut tuoteominaisuudet kehitetään maksimiarvoihin. Alempana sarakkeeseen "Myynti [kpl]" määritetään nykyinen myyntimäärä, jolloin maksimi myyntimäärä on lisämyyntimäärän ja nykyisen myyntimäärän summa. Myyntimäärä tavoitteessa kertoo myyntimäärän tuoteominaisuuksille asetettujen tavoitearvojen mukaisesti, aiemman lisämyyntikertoimen avulla.

		Max.	Tavoite
Lisämyynti, kun spesifikaatioiden arvot maksimissa	150	350	291
Hintajousto-kerroin	-2,00		
Hinnan ja myyntimäärän suhde (arc elasticity E_p)	2,45		

Min - Max

	Minimi	Maksimi
Myyntihinta	12 000,0 €	15 000,0 €
Valmistuskustannus	7 000,0 €	9 450,0 €
Myyntihinnan hintajoustovaikutus [kpl]		-80
Lisämyynti (tuoteparannus) [kpl]		150
Lisämyynti (muut toimenpiteet) [kpl]		0
Myynti [kpl]	200	270
Kokonaiskustannukset	1 400 000 €	2 551 500 €
Liikevaihto	2 400 000 €	4 050 000 €
Kate %	41,7 %	37,0 %
Kate €	1 000 000 €	1 498 500 €

Kuva 17. Lisämyynnin ja hintajoustokertoimen määrittäminen.

Lisäksi optimaattorissa käyttäjä asettaa myyntihinnan minimiarvoilla, joka voi olla myös nykyinen myyntihinta, sekä maksimiarvoilla, eli kuinka paljon myyntihinta voi enintään olla tuoteominaisuuksien saavuttaessa maksimiarvonsa. Lisäksi minimiarvolle asetetaan myös valmistuskustannus, joka on maksimiarvoilla minimiarvon valmistuskustannus lisättynä tuoteominaisuuksien aiemmin asetetut maksimikustannukset.

Optimoitu tulos kohta esittää tuoteominaisuuksien tavoitearvojen mukaiset laskennalliset tulokset sekä muutoksen minimiarvoihin nähden. Lisäksi kuvassa 18 näkyvä keltainen solu antaa mahdollisuuden lisätä myyntimäärää, mikäli sellaista lisäystä tapahtuu muiden toimenpiteiden myötä.

Optimoitu tulos

Tulos	Muutos
13 959,1 €	+ 1 959,1 €
7 802,8 €	+ 802,8 €
-56	
91	
0	
235	+35
1 832 122 €	+ 432 122,1 €
3 277 646 €	+ 877 646,0 €
44,3 %	3 %
1 445 524 €	+ 445 523,9 €

Kuva 18. Optimaattorin tulossarakkeet.

Varsinainen optimointi tapahtuu Excelin Solver-lisäosalla, joka on ilmaiseksi ladattavissa ja Microsoftin tukema lisäosa. Solver-lisäosan käynnistäminen avaa ponnahdusikkunan, jossa tehdään optimoinnin tarvitsemat määrittelyt. Kuvassa 19 ylhäältä lukien ensimmäinen määrittely on optimoitavan kohteen asettaminen, johon on tässä tapauksessa asetettu solu I38, joka on tässä esimerkissä tuotteen kateprosentti. Seuraavana määritetään, onko kyseessä kohteen arvon maksimointi, minimointi vai saaminen tiettyyn arvoon. Tämän jälkeen valitaan mitä arvoja voidaan muuttaa, tässä tapauksessa muutettavia arvoja ovat tuoteominaisuuksien tavoitearvot. Lisäksi määritetään rajoituksia, joiksi on tässä tapauksessa asetettu tuoteominaisuuksien tavoitearvojen ja minimi- sekä maksimiarvojen suhteen.

Solver Parameters

Set Objective:

To: Max Min Value Of:

By Changing Variable Cells:

Subject to the Constraints:

- \$G\$16 >= \$H\$16
- \$G\$16 <= \$F\$16
- \$G\$15 >= \$H\$15
- \$G\$15 <= \$F\$15
- \$G\$13 >= \$H\$13
- \$G\$13 <= \$F\$13
- \$G\$14 <= \$F\$14
- \$G\$14 >= \$H\$14

Make Unconstrained Variables Non-Negative

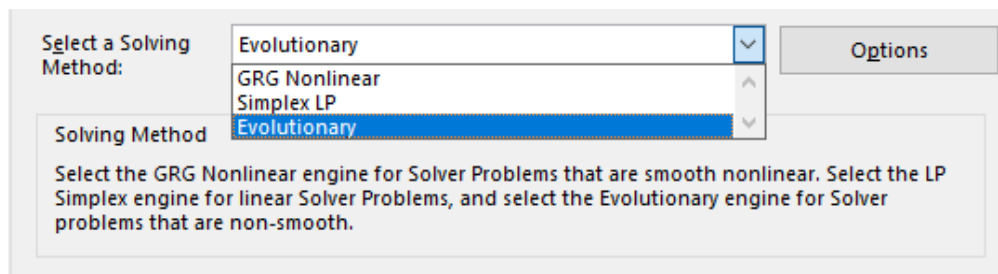
Select a Solving Method:

Solving Method

Select the GRG Nonlinear engine for Solver Problems that are smooth nonlinear. Select the LP Simplex engine for linear Solver Problems, and select the Evolutionary engine for Solver problems that are non-smooth.

Kuva 19. Excel Solver-lisäosan määrittelyt.

Viimeinen vaihe optimoinnin määrittämisessä on valita optimointimenetelmä, jossa vaihtoehtoja on kolme kuvan 20 mukaisesti. Näistä GRG Nonlinear etsii epälineaarisissa tapauksissa lokaalia maksimiarvoa, Simplex LP on tarkoitettu lineaarisiin tapauksiin ja Evolutionary kykenee etsimään epälineaaristen tapauksien globaalia maksimia. Tässä optimointitapauksessa on käytetty Evolutionary menetelmää lisäkustannusten kertymisen epälineaarisuuden takia.



Kuva 20. Excel Solver -optimointimenetelmät.

Näiden määrittämisien jälkeen optimointi voidaan aloittaa, jolloin Solver-lisäosa lähtee etsimään kateprosentin maksimiarvoa muuttamalla tuoteominaisuuksien tavoitearvoja, pysyen asetettujen rajoitusten sisällä. Rajoitusten mukaisesti tavoitearvo ei esimerkiksi voi olla missään tapauksessa numeerisesti pienempi kuin maksimiarvo. Esimerkki optimoinnin lopputuloksesta on esitetty kuvassa 21, josta nähdään, että optimointi asetti tuoteominaisuuksien tavoitearvoille varsin tarkat luvut, joiden kautta kateprosentti on muodostunut.

Prioriteetti	Suhteellinen prioriteetti	Spesifikaatio	Mittayksikkö	Minimi	Tavoite	Maksimi	Maksimikustannus	Roo / kustannusten tyyppi	Lisäkustannus	Lisämyyntikerroin
324	0,329	Latausaika 0 - 80%	min	38	30,6782	25	600 €	5	160,1 €	18,5 %
260	0,264	Massa	kg	157	141,6868	130	800 €	5	74,0 €	15,0 %
202	0,205	Painopisteen korkeus	cm	44	43,2196	40	450 €	5	62,0 €	4,0 %
199	0,202	Kiihtyvyys 0 - 100 km/h	s	6,7	6,3188	5	600 €	1	62,2 €	4,5 %
Yht.	985	1					2 450 €		358 €	42,0 %

		Max.	Tavoite
Lisämyynti, kun spesifikaatioiden arvot maksimissa	150	350	263
Hintajousto-kerroin	-2,00		
Hinnan ja myyntimäärän suhde (arc elasticity E_D)	2,45		

	Min - Max		Optimoitu tulos	
	Minimi	Maksimi	Tulos	Muutos
Myyntihinta	12 000,0 €	15 000,0 €	13 409,4 €	+ 1 409,4 €
Valmistuskustannus	7 000,0 €	9 450,0 €	7 358,3 €	+ 358,3 €
Myyntihinnan hintajoustovaikutus [kpl]		-80	-42	
Lisämyynti (tuoteparannus) [kpl]		150	63	
Lisämyynti (muut toimenpiteet) [kpl]		0	0	
Myynti [kpl]	200	270	221	+21
Kokonaiskustannukset	1 400 000 €	2 551 500 €	1 626 184 €	+ 226 184,0 €
Liikevaihto	2 400 000 €	4 050 000 €	2 963 477 €	+ 563 476,6 €
Kate %	41,7 %	37,0 %	45,4 %	3,7 %
Kate €	1 000 000 €	1 498 500 €	1 337 293 €	+ 337 292,6 €

Kuva 21. Esimerkkilaskennan tulos Excel Solver optimoinnilla.

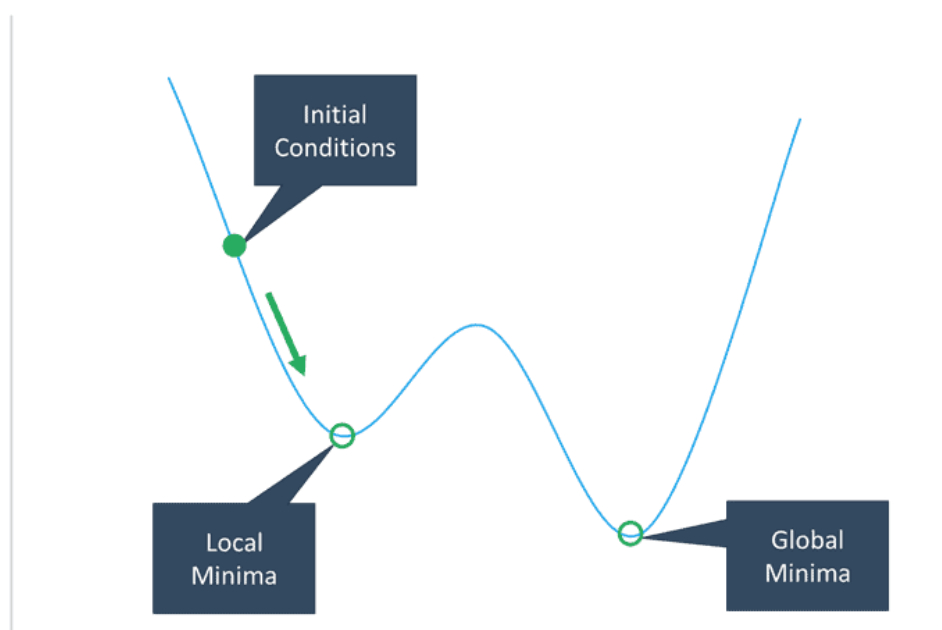
Tuloksista nähdään, että tämän laskennan perusteella kehittämällä esimerkkiin valittuja tuoteominaisuuksia tavoitearvoihin, myyntimäärä paransi 21 kappaleella, liikevaihto kasvasi yli 560 000 euroa, kateprosentti paransi 3,7%-yksikköä ja euromääräinen kate kasvasi yli 300 000 euroa.

6.4 Aineistonhankinta

Optimointimenetelmien vertailu aloitettiin tutustumalla Excel Solver -lisäosan ominaisuuksiin. Kyseisessä lisäosassa on kolme eri menetelmää optimointilaskennan suorittamiselle, Simplex LP, GRG Nonlinear ja Evolutionary.

Simplex LP perustuu matemaatikko George Danzigin toisen maailmansodan aikana kehittämään algoritmiin, jolla kyetään ratkaisemaan lineaarisia optimointiongelmia. Menetelmä on laajasti käytössä nykypäivänä erityisesti erisuuruus ongelmien ratkaisussa. (FrontlineSolvers 2024)

GRG Nonlinear on Leon Lasdonin 1978 esittämä menetelmä epälineaaristen optimointiongelmien ratkaisemiseen. Menetelmä etsii aktiiviset erisuuruus ehdot täyttävää ratkaisua, josta johtuen sen ominaisuutena on löytää lokaali optimi ratkaisu, joka ei välttämättä ole kuitenkaan kyseisen tapauksen globaali optimi. Kuva 17. havainnollistaa tätä eroa. (Engineerexcel 2024, FrontlineSolvers 2024)



Kuva 22. Lokaalin ja globaalin optimin ero GRG Nonlinear menetelmässä. (Engineerexcel 2024)

Evolutionary menetelmä puolestaan perustuu Excel Solver -lisäosassa erilaisiin evolutiivisiin ja geneettisiin algoritmeihin, jotka lähtevät etsimään ratkaisua satunnaisesta aloituskohdasta. Tämän jälkeen algoritmi lähtee muodostamaan niin sanottuja kandidaatti ratkaisuja, joista algoritmi valitsee sopivimman, josta muodostetaan taas uusia kandidaatti ratkaisuja. Menetelmä muodostaa näitä kandidaatti ratkaisuja niin kauan, että asetetut ehdot täyttyvät. (FrontlineSolvers 2024)

Excel Solver -lisäosan tutkimisen lisäksi etsittiin vaihtoehtoja suorittaa optimointi Python ympäristössä. Python on laajasti käytössä datatieteessä ja siihen löytyy useita valmiita kirjastoja, joiden avulla optimointilaskentaa on mahdollista tehdä. Yksi yleisimmistä kirjastoista on SciPy, joka on kehitetty laskennallisen tieteen käyttöön, ja se sisältää useita eri menetelmiä optimointiin sekä lokaalien että globaalien optimien ratkaisemiseen. Scipy kirjastoa kokeilemalla ja dokumentaatiota tutkimalla työssä päädyttiin kokeilemaan seuraavaa kolmea optimointialgoritmia, koska ne vaikuttivat tähän tapaukseen sopivimmilta:

- *Minimize* menetelmä etsii lokaaleja ratkaisuja, ja jota varten täytyy määrittää vähintään optimoitava funktio sekä optimoinnin lähtöpiste. Valittavana on 14 eri algoritmia, joiden perusteella optimointia tehdään.
- *Differential evolution* on puolestaan globaaliin optimointiin tarkoitettu funktio. Menetelmä perustuu Rainer Stornin ja Kenneth Pricen vuonna 1997 tehtyyn julkaisuun.
- *Brute* ratkoo optimointiongelmia käyttämällä ns. brute force menetelmää. Menetelmä laskee funktion alueelta kaikki mahdolliset ratkaisut ja etsii sillä tavalla pienimmän tai suurimman arvon.

6.4.1 Testilaskennat

Eri optimointimenetelmiä vertailtiin tekemällä jokaisella neljällä menetelmällä kymmenen testilaskentaa, joissa jokaisessa oli samat lähtöarvot kuvan 23 mukaisesti. Tärkeimpiä lähtöarvoja näistä ovat tuoteominaisuuksien tavoitearvot sekä kateprosentti. Testilaskentaan optimoinnin kohteeksi valittiin kateprosentin maksimointi, joka muuttuu tuoteominaisuuksien tavoitearvoja muuttamalla. Testilaskennoissa tuoteominaisuuksien tavoitearvot asetettiin puoleen väliin minimi- ja maksimiarvojen eroa.

Lähtöarvot	Minimi	Tavoite	Maksimi	Maksimikustannus	Roo
Spesifikaatio 1	38	31,5	25	600,00 €	5
Spesifikaatio 2	157	143,5	130	800,00 €	5
Spesifikaatio 3	44	42	40	450,00 €	5
Spesifikaatio 4	6,7	5,85	5	600,00 €	1
Lisämyynti, kun spesifikaatioiden arvot maksimissa	150				
Hintajousto-kerroin	-2				
Hinnan ja myyntimäärän suhde (arc elasticity Ep)	2,45				
	Minimi	Maksimi			
Myyntihinta	12 000,00 €	15 000,00 €			
Valmistuskustannus	7 000,00 €	9 450,00 €			
Myyntihinnan hintajoustovaikutus [kpl]		-80			
Lisämyynti (tuoteparannus) [kpl]		150			
Myynti [kpl]	200	270			
Kokonaiskustannukset	1 400 000,00 €	2 551 500,00 €			
Liikevaihto	2 400 000,00 €	4 050 000,00 €			
Kate %	41,67 %	37,00 %			
Kate €	1 000 000,00 €	1 498 500,00 €			

Kuva 23. Testilaskennoissa käytetyt lähtöarvot

Jokaisen testilaskennan tulokset kirjattiin taulukkoon, laskentojen kokonaismäärän ollessa 40, jolloin yksittäisiä tuloksia kertyi 160. Testilaskentojen tuloksista laskettiin spesifikaatiokohtaisesti keskiarvo, keskihajonta, mediaani ja moodi. Näitä havainnollistettiin myös tekemällä tuloksista graafit, joissa saman spesifikaation tulokset esitettiin eri menetelmillä laskettuina.

6.5 Aineiston analyysi

Testilaskentojen yksittäiset tulokset taulukoitiin Excel -taulukkoon, joka on esitetty kuvassa 24. Testilaskennoissa käytettiin yhteensä neljää eri menetelmää. Excel Solver, Evolutionaryn laskennat ovat tehty alkuperäisellä Specsoft optimaattorin Excel Solver -lisäosalla. Tämä laskenta antaa niin sanotun vertailutuloksen muihin laskentoihin nähden, koska tämä menetelmä oli käytössä jo aiemmin.

Kolme muuta testilaskentaa tehtiin Python Scipy kirjaston eri optimointialgoritmeilla, joita olivat Differential Evolution, Minimize sekä Brute Force.

Excel Solver, Evolutionary										
	Testi 1	Testi 2	Testi 3	Testi 4	Testi 5	Testi 6	Testi 7	Testi 8	Testi 9	Testi 10
Latausaika 0 - 80% (min)	30,70	30,36	30,48	30,58	30,81	30,68	30,76	30,72	30,59	30,56
Massa (kg)	141,41	141,22	141,33	141,43	141,53	141,44	141,51	141,55	141,49	141,64
Painopisteen korkeus (cm)	43,27	43,24	43,14	43,33	43,19	43,27	43,23	43,27	43,31	43,13
Kiihtyvyyys 0 - 100 km/h (s)	6,32	6,29	6,30	6,32	6,31	6,32	6,31	6,30	6,31	6,26
Kateprosentti (%)	45,37 %	45,37 %	45,37 %	45,37 %	45,37 %	45,37 %	45,37 %	45,37 %	45,37 %	45,37 %

Python, Differential Evolution										
	Testi 11	Testi 12	Testi 13	Testi 14	Testi 15	Testi 16	Testi 17	Testi 18	Testi 19	Testi 20
Latausaika 0 - 80% (min)	30,58	30,58	30,58	30,58	30,58	30,58	30,58	30,58	30,58	30,58
Massa (kg)	141,41	141,41	141,41	141,41	141,41	141,41	141,41	141,41	141,41	141,41
Painopisteen korkeus (cm)	43,21	43,21	43,21	43,21	43,21	43,21	43,21	43,21	43,21	43,21
Kiihtyvyyys 0 - 100 km/h (s)	6,30	6,30	6,30	6,30	6,30	6,30	6,30	6,30	6,30	6,30
Kateprosentti (%)	45,13	45,13	45,13	45,13	45,13	45,13	45,13	45,13	45,13	45,13

Python, Minimize										
	Testi 21	Testi 22	Testi 23	Testi 24	Testi 25	Testi 26	Testi 27	Testi 28	Testi 29	Testi 30
Latausaika 0 - 80% (min)	30,49	30,49	30,49	30,49	30,49	30,49	30,49	30,49	30,49	30,49
Massa (kg)	141,32	141,32	141,32	141,32	141,32	141,32	141,32	141,32	141,32	141,32
Painopisteen korkeus (cm)	43,12	43,12	43,12	43,12	43,12	43,12	43,12	43,12	43,12	43,12
Kiihtyvyyys 0 - 100 km/h (s)	6,70	6,70	6,70	6,70	6,70	6,70	6,70	6,70	6,70	6,70
Kateprosentti (%)	45,02	45,02	45,02	45,02	45,02	45,02	45,02	45,02	45,02	45,02

Python, Brute										
	Testi 31	Testi 32	Testi 33	Testi 34	Testi 35	Testi 36	Testi 37	Testi 38	Testi 39	Testi 40
Latausaika 0 - 80% (min)	30,47	30,47	30,47	30,47	30,47	30,47	30,47	30,47	30,58	30,58
Massa (kg)	141,37	141,37	141,37	141,37	141,37	141,37	141,37	141,37	141,41	141,41
Painopisteen korkeus (cm)	43,16	43,16	43,16	43,16	43,16	43,16	43,16	43,16	43,21	43,21
Kiihtyvyyys 0 - 100 km/h (s)	6,70	6,70	6,70	6,70	6,70	6,70	6,70	6,70	6,30	6,30
Kateprosentti (%)	45,02	45,02	45,02	45,02	45,02	45,02	45,02	45,02	45,13	45,13

Kuva 24. Testilaskentojen tulokset neljällä eri menetelmällä.

Testilaskentoja tehtiin yhteensä 40 kappaletta, joista jokainen sisältää 5 numeerista arvoa, jolloin vertailua varten yksittäisiä tarkasteltavia arvoja on yhteensä 200 kappaletta. Testilaskentojen optimoinnin kohteena oli kateprosentti, jonka laskentaa varten taulukkoon kirjattiin kaikki neljä tuoteominaisuutta sekä niiden spesifikaatioiden arvot. Nämä arvot ovat optimointilaskennassa muuttuvia arvoja, joita tarkastelemalla voidaan arvioida eri optimointialgoritmien laskemien tuloksien eroavaisuuksia.

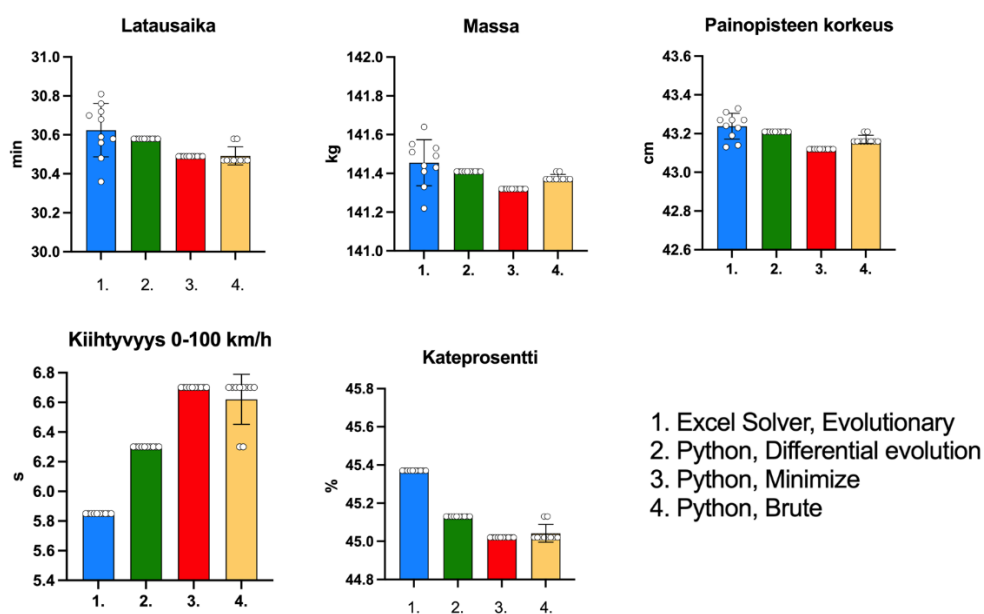
7 TULOKSET

Tarkastelemalla taulukoituja laskentatuloksia voidaan havainnoida optimoinnin käyttäytymistä. Esimerkiksi Excel Solverin Evolutionary optimoinnin tuloksista kaikissa testilaskennoissa kateprosentin optimoitutulos on sama 45,37 %. Tuoteominaisuuksien spesifikaatioiden arvot ovat kuitenkin jokaisessa laskennassa saaneet eri arvot, ja testilaskentojen välillä on erilaisia yhdistelmiä spesifikaatioiden arvoista. Tästä voidaan päätellä, että kateprosentin suhteen samaan lopputulokseen voidaan päästä usealla eri spesifikaatioiden arvojen yhdistelmällä.

Python Differential Evolution optimointi on puolestaan laskenut jokaisella testilaskennalla tulokseksi saman kateprosentin kuin Excel Solver. Kateprosentti on kuitenkin hieman pienempi, 45,13 %, ja tämä menetelmä on myös antanut jokaisella testilaskennalla yksittäisille spesifikaatioille samat arvot. Tämän menetelmän optimointi vaikuttaisi siis pääsevän jokaisella laskentakerralla yhtäläiseen lopputulokseen sekä muutettavien arvojen että optimoitavan tavoitteen suhteen.

Taulukoitujen tuloksien perusteella Python Minimize vaikuttaisi toimivan samoin kuin Python Differential Evolution, eli jokaisella laskennalla tulos on sama, kuten myös yksittäisten spesifikaatioiden arvot. Tulos ja arvot poikkeavat kuitenkin Differential Evolution laskennasta hieman, kateprosentin ollessa hieman pienempi 45,02 %.

Python Brute puolestaan toiminut ensimmäisen kahdeksan laskentaa (testit 31-38) kuten Differential Evolution ja Minimize, jolloin se on saanut tulokseksi kateprosentiksi 45,02 %, kuten Minimize, ja yksittäiset tuoteominaisuuksien arvot ovat pysyneet samoina laskentojen välillä. Viimeiset kaksi laskentaa, testit 39 ja 40, ovat puolestaan saaneet hieman erilaiset arvot aiempiin laskentoihin nähden.



Kuva 25. Eri optimointimenetelmien testilaskentojen spesifikaatioiden ja kateprosentin arvot graafissa. Yksittäiset arvot ovat kuvattu ympyröillä, palkki esittää keskiarvon ja T-viiva keskihajonnan.

Taulukoinnin lisäksi tulosten vertailua varten datasta muodostettiin kaaviot kuvastamaan eri menetelmillä laskettujen tuoteominaisuuksien spesifikaatioiden arvoja sekä kateprosentteja. Kaaviot kerättiin kuvaan 25, josta niitä voi tulkita ja vertailla helposti. Lisäksi tarkempaa numeerista vertailua varten klassiset tilastolliset muuttujat keskiarvo, keskihajonta, mediaani ja moodi laskettiin Excel-taulukoon, joka esitetään kuvassa 26. Numeeristen muuttujien vertailu helpottaa datan tulkintaa ja eroavaisuuksien suuruuden hahmotusta kaavion rinnalla.

Excel Solver, Evolutionary			
<i>Keskiarvo</i>	<i>Keskihajonta</i>	<i>Mediaani</i>	<i>Moodi</i>
30,62	0,14	30,64	-
141,46	0,12	141,47	-
43,24	0,07	43,26	43,27
6,30	0,02	6,31	6,31 & 6,32
45,37 %	0,00 %	45,37 %	45,37 %

Python, Differential Evolution			
<i>Keskiarvo</i>	<i>Keskihajonta</i>	<i>Mediaani</i>	<i>Moodi</i>
30,58	0,00	30,58	30,58
141,41	0,00	141,41	141,41
43,21	0,00	43,21	43,21
6,30	0,00	6,30	6,30
45,13	0,00	45,13	45,13

Python, Minimize			
<i>Keskiarvo</i>	<i>Keskihajonta</i>	<i>Mediaani</i>	<i>Moodi</i>
30,49	0,00	30,49	30,49
141,32	0,00	141,32	141,32
43,12	0,00	43,12	43,12
6,70	0,00	6,70	6,70
45,02	0,00	45,02	45,02

Python, Brute			
<i>Keskiarvo</i>	<i>Keskihajonta</i>	<i>Mediaani</i>	<i>Moodi</i>
30,49	0,05	30,47	30,47
141,38	0,02	141,37	141,37
43,17	0,02	43,16	43,16
6,62	0,17	6,70	6,70
45,04	0,05	45,02	45,02

Kuva 26. Eri menetelmien testilaskentojen spesifikaatioiden arvojen keskiarvot, mediaanit ja moodit.

Kuvan 25 kaavioissa yksittäiset arvot ovat kuvattu ympyröillä, jolloin on helppo tarkastella miten yksittäiset arvot asettuvat eri menetelmien välillä. Pystypalkit esittävät yksittäisten arvojen keskiarvot sekä T-viiva keskihajonnan.

Kaavioita tarkastelemalla nähdään, että yksittäisten arvojen suhteen eniten hajontaa on Excel Solver Evolutionary laskennoissa. Python Differential Evolution ja Minimize laskentojen yksittäisissä arvoissa ei ole ollenkaan hajontaa. Python Brutessa puolestaan on pientä hajontaa yksittäisten arvojen kesken, koska kaksi laskentaa kymmenestä antoi erilaiset tulokset.

Laskentojen välillä on jonkin verran eroavaisuuksia keskiarvojen välillä, mutta kokonaisuudessaan tulokset eri menetelmien välillä ovat samansuuntaisia. Eniten eroa keskiarvoissa on Excel Solver Evolutionary ja Python Minimize menetelmien välillä.

8 JOHTOPÄÄTÖKSET JA POHDINTA

Tässä työssä tarkasteltiin Specsoft -hankkeessa kehitettävän tuotekehityksen konseptointivaiheen tuoteominaisuuksien spesifikaatioiden määrittämisen optimointimenetelmiä. Työn tavoitteena oli saada Exceliin rakennetusta optimaattorista verkkopohjaiseen sovellukseen siirrettävä optimaattori. Aiemmassa Excel optimaattorissa oli epäselvää, kuinka Excelin optimaattori laskentaa tekee, koska vertailupohjaa tai toista vaihtoehtoa optimointilaskennalle ei vielä ollut. Työn tuloksena saatiin Excelin pohjalta rakennettua ohjelmoitava optimaattori, jonka sovelluskehittäjät voivat viedä hankkeessa kehitettävään Specsoft -websovellukseen. Tämä optimaattorimoduulin kehitysvaihe oli tärkeä osa hanketta, ettei sovelluskehittäjien tarvinnut lähteä ratkaisemaan optimaattorin kehittämistä pelkän Excelin pohjalta. Sovelluksessa on iso määrä muita monimutkaisia ja tärkeitä kehityskohteita, joista optimaattorimoduuli on vain yksi osa kokonaisuutta. Työssä Excel optimaattoria tarkasteltiin ja testailtiin kattavasti, sekä sen pohjalta rakennettiin uusi versio Python -ohjelmointiympäristöön. Python ympäristö tarjosi monia algoritmivaihtoehtoja optimointilaskentaan, jonka myötä voitiin tehdä helposti eri optimointimenetelmien vertailua. Pythonin ollessa moderni ja avoin ohjelmointikieli, sen siirtäminen verkkosovellukseen on mahdollista.

Kehitettävä tuote kone- ja laiterakennuksen alalla käsittää yleensä monia satoja tai jopa tuhansia erilaisia suunnitteluvaatimuksia. Jotkut vaatimuksista ovat teknisiä, ja vaikuttavat pääasiassa tuotesuunnitteluun tai valmistukseen, mutta eivät juurikaan näy asiakkaalle. Toiset puolestaan ovat asiakkaille enemmän näkyviä, joista jotkin tietyt ovat yhdelle asiakkaalle tärkeimmät ja toiset vaatimukset toiselle. Erilaiset tuoteominaisuudet ovat enemmän tai vähemmän sidoksissa toisiinsa, ja niiden kehittäminen vaikuttaa tuotteen valmistuskustannuksista myyntimenestykseen asti. Systemaattista ja tehokasta tuotekehitystä varten näistä lukuisista suunnitteluvaatimuksista pitäisi osata valita ne, joita kehittämällä saavutetaan toivottuja tuloksia.

Työssä käsitellyillä tuotekehityksen menetelmillä, QFD:llä, AHP:lla ja Kano-mallilla saadaan kaikkien suunnitteluvaatimusten joukosta rajattua haluttu määrä vaatimuksia tuotekehitystä varten, sekä priorisoitua nämä valitun kehitysnäkökulman mukaisesti. Kehitysnäkökulmia voi olla esimerkiksi asiakas tyytyväisyyden parantaminen, valmistuskustannusten pienentäminen tai myyntimäärän maksimoiminen. Tämän jälkeen päästään määrittelemään kullekin suunnitteluvaatimukselle raja-arvot, eli mitkä kyseisen vaatimuksen minimi- ja maksimi tasot ovat. Tämä tapahtuu tuotteen ja sen valmistus- ja suunnitteluprosessin parhaiten tuntevien henkilöiden toimesta. Minimitasona voidaan yleistetysti pitää nykytilaa, koska tuotekehityksen tavoitteena on useimmiten parantaa tuotteita. Maksimitaso puolestaan on se taso, johon kyseinen vaatimus voidaan parhaillaan kehittää tällä hetkellä. Maksimitasolle kehittäminen voi lisätä tuotteen jatkuvia valmistuskustannuksia, joka puolestaan vaikuttaa esimerkiksi myyntihintaan, myyntimäärään ja tuotteista saatuun katteeseen.

Kun näillä menetelmillä on saatu haarukoitua lukuisten suunnitteluvaatimusten joukosta ne, joita lähdetään kehittämään, jää määriteltäväksi vielä mitä tasoa kunkin ominaisuuden kehittämisessä tavoitellaan. Jos tavoiteltu taso jää liian matalaksi, tuotekehitys ei täytä sille asetettuja vaatimuksia. Toisaalta jos kaikki kehitykseen valitut vaatimukset kehitetään huippuunsa, aiheuttaa se pahimmillaan huomattavia ylimääräisiä kustannuksia kokonaisuudyn jäädessä pieneksi. Tavoiteltu taso on siis jossain määriteltujen minimi- ja maksimitasojen välillä, jonka määrittämiseen ratkaisun tarjoaa optimointilaskenta. Optimointilaskenta puolestaan tarvitsee pohjaksi määritellyn matemaattisen mallin, joka kuvaa käsiteltävän ongelman numeerisesti.

Tuotekehityksen konseptointivaiheeseen *Specsoft*-sovelluksen optimaattorimoduuli tuo selkeästi etuja, koska todellisessa tuotekehitysprojektissa tuoteominaisuuksia voi olla tässä työssä käsiteltyihin neljään nähden moninkertaisesti. Tuoteominaisuuksien suuri määrä tekee spesifikaatioiden arvojen määrittelyn erittäin monimutkaiseksi, koska mahdollisia arvojen yhdistelmiä on erittäin paljon. Siinä kehitetyn kaltainen optimaattori on tarpeellinen aputyökalu, koska sillä saa laskettua tavoitearvoja valitun näkökulman suhteen. Tuotekehitysprojektissa voidaan sitten valita, käytetäänkö suoraan optimaattorin antamia arvoja, vai määritelläänkö niiden pohjalta itse tavoitearvot. Tavoitearvoihin pääsemiseen todellisilla tuotteilla vaikuttaa varsin monet asiat, kuten esimerkiksi tarvittavan suunnittelun määrä, käytettävissä olevat valmistustekniikat, henkilöstön osaaminen jne. Kokonaisuudessaan *Specsoft*-sovellus ja tässä työssä kehitetty optimaattori nivoo yhteen monipuolisesti tuotekehityksen menetelmiä, niiden pohjalta rakennetun matemaattisen mallin sekä optimointilaskennan.

Optimointimenetelmien vertailussa vertailtiin eri laskentojen tuloksia, joissa optimoinnin kohteena oli kateprosentti. Laskennoissa kateprosentti sai hieman vaihtelevia arvoja, mutta kokonaisuudessaan se oli varsin yhdenmukainen. Todellisessa elämässä toteutuvan kateprosentin arviointi kahden desimaalin tarkkuudella ei ole kovinkaan relevanttia, joten siten laskentojen tuloksia voitiin pitää riittävän tarkkoina. Sama pätee myös tuoteominaisuuksien spesifikaatioiden lukuarvoihin – niiden määrittäminen todella tarkkaan on suurimmassa osassa kone- ja laiterakennuksen tuotekehitysprojekteja käytännössä mahdotonta. Laskentojen tuloksena saatavat spesifikaatioiden arvot toimivat kuitenkin hyvinä tavoitteina, joihin tuotekehityksessä voidaan tähdätä. Tuotekehitysprojektin johto voisi myös asettaa vaihteluvälin, mille alueelle lopputuloksen tuoteominaisuuden spesifikaation arvon on saatava suhteessa optimoituihin tavoitearvoihin. Tämä antaisi mielestäni tarkoituksenmukaisen, perustellun ja uskottavan tavoitteen, mihin tuotekehityksessä pyritään teknisten spesifikaatioiden suhteen.

Tutkimuskysymyksenä oli, että mikä on sopiva optimointimenetelmä *Specsoft*-sovellukseen. Työn tuloksia tarkastelemalla mitään yksittäistä optimointimenetelmää ei voi suoraan sulkea pois epäkelpona tai toimimattomana. Näistä menetelmistä kuitenkin Excel Solverin Evolutionary sekä Python Differential Evolution on kehitetty erityisesti epälineaaristen optimointifunktioiden ratkaisemiseen, jollainen Bistech -hankkeessa kehitetty optimaattori on. Vaikka muutkin menetelmät tuloksia tarkastelemalla vaikuttavat toimivan, vastaisi epälineaarisen ratkaisumenetelmän käyttö parhaiten *Specsoft* optimaattorin tarpeita.

Tämä opinnäytetyö syvensi omaa osaamistani optimointimenetelmistä sekä tuotekehityksestä, erityisesti tuotekehityksen konseptointivaiheesta. Optimointilaskenta on erittäin laaja ja haastava aihealue, josta minulla ei ollut aiempaa kokemusta. Myös osaamiseni ohjelmoinnissa kasvoi paljon, koska aiempaa osaamista oli vain vähän perustasolla. Työssä hyödynnettiin ChatGPT 3.5 -tekoälysovellusta. Ohjelmoinnissa tekoälystä oli suuri hyöty, koska siitä sain apua koodin muodostamiseen. ChatGPT 3.5 ei kuitenkaan osannut tehdä suoraan toimivaa ohjelmakokonaisuutta, vaan kykeni antamaan toimivia koodirivejä sekä ohjeistuksia koodin syntaksiin ja toimintaan liittyen. Työ ja sen aihealue oli erittäin haastava, koska minulla oli aiempaa osaamisesta lähinnä tuotekehityksestä. Toisaalta työn haastavuus motivoi työn tekemisessä, ja työtä tehdessä oppi uutta.

LÄHTEET

Työssä on käytetty tekoälyä seuraavasti:

ChatGPT 2023. OpenAI. GPT-3.5. Käytetty Python ohjelmoinnin aputyökaluna, joulukuu 2023.

<https://chat.openai.com>

ChatGPT 2024-2025. GPT-4.o. Käytetty lähdeaineistojen kartoittamisen apuna, kielenkääntäjänä sekä tekstin rakenteen ideointiin. Joulukuu 2024–maaliskuu 2025.

Baker K. R. 2015. *Optimization modeling with spreadsheets*. Hoboken, New Jersey: John Wiley & Sons, Inc.

Bender E. A. 1978. *An introduction to mathematical modeling*. New York: John Wiley & Sons, Inc.

Dym C. L. 2004. *Principles of mathematical modeling*. Amsterdam: Elsevier Academic Press.

Engineerexcel 2024. *Excel Solver: Which Solving Method Should I Choose?*

<https://engineerexcel.com/excel-solver-solving-method-choose/>.

FrontlineSolvers 2024. *Excel Solver - Optimization Methods*. <https://www.solver.com/excel-solver-optimization-methods>.

Hanketietopalvelu E. 2021 2024. Specsoft - EU:n alue- ja rakennepolitiikan hanketietopalvelu ohjelmakaudelle 2021—2027. luettu, <https://eura2021.fi/hanketietopalvelu?koodi=A80089,2024-2-2>.

Humi M. 2017. *Introduction to mathematical modeling*. Boca Raton: CRC press Taylor & Francis.

Juhila K. 2021. Laadullinen tutkimus ja teoria. *Laadullisen tutkimuksen verkkokäsikirja*.

<https://www.fsd.tuni.fi/fi/palvelut/menetelmaopetus/kvali/mita-on-laadullinen-tutkimus/laadullinen-tutkimus-ja-teoria/>.

Khan S. B., Thilagavathy R., Priyadarsini K., Arul L. R. 2024. A Survey on Non – linear Optimization and Global Optimization Methods. *International Journal of Computational Mathematics and Computer Science*.

Kirkwood C. W. 1997. *Strategic decision making: multiobjective decision analysis with spreadsheets*. Belmont: Duxbury Press.

Lager T. 2019. *Contemporary quality function deployment for product and process innovation: towards digital transformation of customer and product information in a new knowledge-based approach*. Singapore: World Scientific.

Laine M., Bamberg J., Jokinen P. 2015. *Tapaustutkimuksen taito*. Helsinki: Gaudeamus Helsinki University Press.

Maritan D. 2015. *Practical Manual of Quality Function Deployment*. Switzerland: Springer International Publishing.

Mital A., Desai A. 2014. *Product development: a structured approach to consumer product development, design, and manufacture*. Amsterdam: Elsevier.

Mu E., Pereyra-Rojas M. 2018. *Practical Decision Making using Super Decisions v3: An Introduction to the Analytic Hierarchy Process*. Cham: Springer.

Nayak S. 2020. *Fundamentals of optimization techniques with algorithms*. London San Diego, CA Cambridge, MA Oxford: Academic Press.

Rakennerahastotietopalvelu 2018. *CusTech - Euroopan aluekehitysrahaston (EAKR) rahoittaman hankkeen kuvaus*. <https://www.eura2014.fi/rrtiepa/projekti.php?projektkoodi=A73574>.

Rakennerahastotietopalvelu 2020. *BisTech - Euroopan aluekehitysrahaston (EAKR) rahoittaman hankkeen kuvaus*. <https://www.eura2014.fi/rrtiepa/projekti.php?projektkoodi=A76050>.

Rodgers P., Milton A. 2011. *Product design*. London: Laurence King Publishing Ltd.

Saaty T. L. 2008. Decision making with the analytic hierarchy process. *International Journal of Services Sciences* 1: 83.

Soam S. K., N S. R., Bs Y., Balasani R., S R., Marwaha S., ym. 2023. AHP Analyser: A decision-making tool for prioritizing climate change mitigation options and forest management. *Frontiers in Environmental Science* 10.

Srebrenkoska S., Apostolova A., Dzidrov M., Krstev D. 2023. Application of Analytical Hierarchy Process (AHP) in the Selection of a Flexible Production System. *Engineering World* 5: 138–143.

Tietoarkisto 2024. *Tutkimusasetelma*.
<https://www.fsd.tuni.fi/fi/palvelut/menetelmaopetus/kvanti/tutkimus/asetelma/>.

Ulrich K. T., Eppinger S. D. 2016. *Product design and development*. New York, NY: McGraw-Hill Education.

Weiss S. I. 2013. *Product and systems development: a value approach*. Hoboken, N.J: John Wiley & Sons.