

Laura Rantonen

**Kombinatorinen optimointi
vuorovaikutussuunnittelussa**

Tietotekniikan kandidaatintutkielma

30. huhtikuuta 2018

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Laura Rantonen

Yhteystiedot: `laura.h.rantonen@gmail.com`

Työn nimi: Kombinatorinen optimointi vuorovaikutussuunnittelussa

Title in English: Combinatorial Optimization in Interaction Design

Työ: Kandidaatintutkielma

Sivumäärä: 24+0

Tiivistelmä: Käyttöliittymien sekä ihmisen ja tietokoneen välisen vuorovaikutuksen suunnittelu on muuttunut järjestelmien monimutkaistuessa yhä hankalammaksi tehtäväksi. Ongelmaan on ehdotettu ratkaisuksi kombinatoristen optimointikeinojen hyödyntämistä käyttöliittymien suunnittelun tukena. Tämän tutkielman tarkoituksena on esitellä tunnetuimpia hyvän käyttöliittymäsuunnittelun periaatteita, sekä käydä läpi kombinatorisen optimoinnin keinoja, joita tähän asti on käyttöliittymäoptimoinnin saralla tutkittu. Tavoitteena on selvittää, millaisia hyötyjä optimointiavusteisella suunnittelulla on mahdollista saavuttaa, sekä pohtia minkälaisia seikkoja täytyy ottaa huomioon kun ihmiskeskeiseen suunnitteluun otetaan mukaan algoritmisia optimointimenetelmiä.

Avainsanat: vuorovaikutussuunnittelu, optimointi, laskennalliset menetelmät, käytettävyys

Abstract: Designing user interfaces has become a challenging process, as systems grow to be more complicated over time. Combinatorial optimization has been discussed as a solution to support user interface designers in their work. The purpose of this thesis is to introduce HCI guidelines for better design, and to introduce research related to combinatorial optimization on user interfaces. The goal is to examine the benefits of optimization assisted design methods, and to consider what subjects need to be considered when bringing algorithmic applications into a traditional design process.

Keywords: interaction, optimization, computational methods, usability

Kuviot

Kuvio 1. Käytettävyyden hierarkia.	4
---	---

Sisältö

1	JOHDANTO	1
2	VUOROVAIKUTUSSUUNNITTELU JA KÄYTETTÄVYYS	3
	2.1 Käytettävyyden ominaisuudet	3
	2.2 Predikttiiviset mallit	5
	2.2.1 GOMS-malli	6
	2.2.2 KLM-malli	7
	2.2.3 Fittsin laki	8
	2.2.4 Hick-Hymanin laki	9
3	KOMBINATORINEN OPTIMOINTI.....	10
	3.1 Geneettiset algoritmit	10
	3.1.1 Interaktiiviset geneettiset algoritmit	12
4	KÄYTTÖLIITTYMIEN VUOROVAIKUTUKSEN OPTIMOINTI	13
	4.1 Suunnitteluongelman määrittely	13
	4.2 Tyypillisimmät suunnitteluongelmat	14
	4.3 Vuorovaikutuksen optimoinnin hyödyt	15
5	YHTEENVETO	18
	KIRJALLISUUTTA	20

1 Johdanto

Ihmisen ja tietokoneen välisen vuorovaikutuksen suunnittelussa esiintyy nykypäivänä monia eri haasteita. Perinteisten käyttöliittymien suunnittelu on muuttunut muun muassa esineiden internetin sekä sulautettujen järjestelmien ratkaisujen myötä aiempaa monimutkaisemmaksi. Esimerkiksi erilaisten kodinkoneiden ja viihdeelektroniikan käyttöliittymien suunnittelussa on tavallisten suunnitteluongelmien lisäksi otettava huomioon myös muun muassa laitteiden erilaiset ruutukoot sekä kehitettävän järjestelmän laitteiston rajallinen suorituskyky. Hyvän vuorovaikutteisuuden takaavia järjestelmiä suunnitellessa on otettava teknologian lisäksi huomioon myös järjestelmän käyttäjä, eli ihminen, joka luo omat haasteensa käyttöliittymän suunnitteluprosessiin. Ihmiskaspektin huomioimista tutkiva monitieteinen tieteenala, ihmisen ja tietokoneen välinen vuorovaikutus (engl. *Human–Computer Interaction*) pyrkii vastaamaan näihin haasteisiin (Dix, Finlay, Abowd & Beale 2009).

Käyttöliittymäsuunnittelu on perinteisesti ollut täysin ihmisen tekemien suunnittelupäätösten ja -metodien sekä intuition varassa. Mahdollisena vaihtoehtona perinteisille suunnittelumetodeille onkin tutkittu laskennallisten menetelmien sekä tietokonepohjaisten algoritmien hyödyntämistä suunnittelun tukena. Oulasvirta ym. (Oulasvirta, Kristensson, Bi & Howes 2018) ovat teoksessaan käsitelleet laskennallisista menetelmistä erityisesti kombinatorista optimointia, jonka avulla on pystytty saavuttamaan käytettävyydsarvoiltaan optimaalisia käyttöliittymiä. Tietokonepohjaisten algoritmien hyödyntämisen johdosta onkin noussut esille myös kysymys siitä, miten hyvin laskennallisten menetelmien avulla pystyttäisiin mallintamaan ihmisen ja teknologian välisen vuorovaikutuksen käsitteiden, kuten esimerkiksi ihmisen käyttäytymisen piirteiden monimuotoisuutta.

Tämä tutkimus keskittyy käsittelemään kombinatorisen optimoinnin menetelmien sekä mallien käyttöä digitaalisten järjestelmien käyttöliittymien suunnittelussa. Aihetta tarkastellaan erityisesti ihmisen ja tietokoneen välisen vuorovaikutuksen sekä käyttöliittymäsuunnittelijan näkökulmasta. Tutkimusmetodina toimii systemaattinen kirjallisuuskatsaus, jonka tarkoituksena on avata aihealueen käsitteitä sekä tar-

kastella lähivuosien aikana julkaistuja tutkimuksia ja saavutuksia käyttöliittymäoptimoinnin saralla.

Tutkimuksen toisessa luvussa avataan ihmisen ja tietokoneen välisen vuorovaikutuksen sekä vuorovaikutussuunnittelun käsitteitä, ja tuodaan esille käyttöliittymien käytettävyyden eri ominaisuuksia, joiden avulla hyvä käyttöliittymäsuunnittelu on mahdollista saavuttaa. Tässä kohtaa tutkielmaa tutustutaan myös ihmisen ja tietokoneen välistä vuorovaikutusta kuvaaviin prediktiivisiin malleihin, joita voidaan hyödyntää käyttäjän toiminnan simulointiin ja ennustamiseen.

Kolmannessa luvussa esitellään kombinatorisia optimoinnin käsitettä, sekä esitellään tunnetuimpia kombinatorisen optimoinnin ongelmia. Tutkielman painopisteenä ovat kombinatorisiin optimointimenetelmiin kuuluvat geneettiset algoritmit, ja luvussa avataan lyhyesti geneettisten algoritmien toimintaa sekä sovellutuksia käyttöliittymäoptimoinnin saralla.

Neljäs luku yhdistää edellä käsitellyt teemat, avaamalla käyttöliittymäoptimoinnin käsitettä sekä optimointitehtävien suunnitteluongelmien määrittelyä. Luvussa tutustutaan myös tyypillisimpiin suunnitteluongelmiin, joita käyttöliittymäoptimoinnin avulla voidaan ratkaista.

Tutkielman lopussa sijaitseva johdanto sisältää pohdintaa sekä päätelmiä optimoinnin hyödyllisyydestä, vuorovaikutussuunnittelun näkökulmasta. Tutkimuksen tavoitteena on myös vertailla perinteisiä suunnittelumetodeja optimointiavusteisiin suunnittelijametodeihin, sekä pohtia miten suunnittelijan työskentely muuttuu kun optimointi liitetään mukaan suunnitteluprosessiin.

2 Vuorovaikutussuunnittelu ja käytettävyys

Ihmisen ja tietokoneen välinen vuorovaikutus (HCI, engl. *Human–Computer Interaction*) on tieteenala, jonka tarkoituksena on tutkia kuinka teknologia vaikuttaa ihmisen ajatteluun ja toimintaan, minkälaisia ominaisuuksia hyvillä interaktiivisilla järjestelmillä on, ja kuinka suunnittelua voitaisiin parantaa. HCI on monitieteinen tutkimusalue, jossa ihmisen ja teknologian välistä vuorovaikutusta voidaan tarkastella esimerkiksi psykologian, kognitiotieteiden, ergonomian tai käyttäjäkokemuksen näkökulmista (Dix ym. 2009).

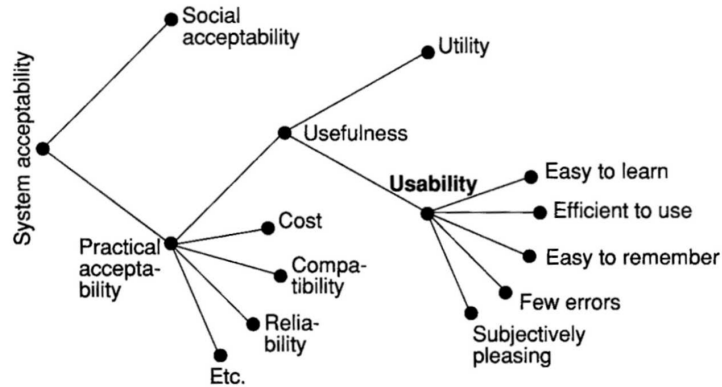
Vuorovaikutussuunnittelu on tärkeä osa HCI:n tieteenalaa sekä digitaalisten järjestelmien käyttöliittymien suunnittelua. Suunnittelua voidaan yleisesti kuvata prosessina, jonka tarkoituksena on saavuttaa tietty tavoite määrättyjen rajoitteiden eli ulkoisten tekijöiden perusteella. Tavoitteisiin pääseminen edellyttää käyttäjäkohde-ryhmän, heidän tarpeidensa sekä suunniteltavan objektin tarkoituksen ymmärtämistä.

Ihmisen ja tietokoneen välisen vuorovaikutuksen suunnittelussa, eli lyhyemmin vuorovaikutussuunnittelussa yhdistyvät sekä ihmisen eli käyttäjän, ja teknologian kuten esimerkiksi laitteiston tuomat rajoitteet. Vuorovaikutussuunnittelun perimmäisenä tavoitteena on siis rakentaa vuorovaikutusjärjestelmiä, jotka ottavat ihmisten ominaisuuksia monipuolisemmin huomioon ja ovat käytettävyydeltään mahdollisimman helppokäyttöisiä (Dix ym. 2009).

2.1 Käytettävyyden ominaisuudet

Tässä alaluvussa käydään läpi käytettävyyden eri ominaisuuksia, jotka pohjautuvat käytettävyydsiantuntija Jakob Nielsenin (Nielsen 1994) ajatuksiin käytettävyydestä. Käytettävyys (engl. *Usability*) on vuorovaikutussuunnittelun metodologia, jolla voidaan mitata kuinka hyvin järjestelmä ottaa käyttäjän rajoitteet sekä tarpeet huomioon. Nielsenin (Nielsen, J. 1994) mukaan käytettävyyttä voidaan arvioida vastaa sitten, kun järjestelmä on ensin saavuttanut hyväksyttävyyden (engl. *Acceptability*)

sekä hyödyllisyyden (engl. *Usefulness*). Kuvio 1 voidaan hahmottaa käytettävyyden eri ominaisuudet, sekä hyväksyttävyyden ja hyödyllisyyden sijainnit käytettävyyden hierarkiassa. Nielsenin (Nielsen, J. 1994) mukaan käytettävyyden keskeisiä ominaisuuksia ovat järjestelmän opittavuus, tehokkuus, muistettavuus, virheiden hallinta sekä käyttäjän subjektiivinen tyytyväisyys järjestelmään.



Kuvio 1. Käytettävyyden hierarkia.

Järjestelmän tulisi olla helposti opittava, jotta käyttäjä pääsisi mahdollisimman nopeasti suorittamaan vaadittavaa tehtävää. Järjestelmän tulisi olla joko tarpeeksi intuitiivinen tai vaihtoehtoisesti opastaa käyttäjää mahdollisissa ongelmatilanteissa.

Tehokkuus on toinen osa käytettävyyttä. Järjestelmän tulisi olla tarpeeksi tehokas käytettävyydeltään, jotta käyttäjä pystyisi saavuttamaan tavoitteensa mahdollisimman nopeasti ja tuottavasti. Tehokkuuden mittana voidaan käyttää esimerkiksi aikaa, joka käyttäjältä kuluu tehtävän suorittamiseen.

Järjestelmän tulisi myös vähentää käyttäjän muistin kuormittamista. Muistettavuudella tarkoitetaan sitä, että järjestelmää on kohtuullisen helppo käyttää myös uudestaan edellisen käyttökerran jälkeen, ilman että järjestelmän käyttämisestä tarvitsisi opetella uudestaan. Järjestelmän muistettavuus auttaa myös käyttäjää navigoimaan järjestelmän ympäristössä, mikäli käyttäjän ei tarvitse muistaa kaikkia ympäristönsään tekemiä toimintoja ulkoa — järjestelmän tulisi kertoa tilasta jossa käyttäjä on nyt, sekä mahdollisesti myös askeleet joiden kautta tilaan on päästy.

Käytettävyyden osa-alueisiin kuuluu myös järjestelmän virheiden hallinta. Järjestelmän virhealttius tulisi olla matala, jotta käyttäjät tekisivät mahdollisimman vähän virheitä järjestelmän käytön aikana. Jos virheitä kuitenkin tapahtuu, virheistä toimimisen täytyisi olla myös vaivatonta ja nopeaa, jotta käyttäjä pääsisi jatkamaan suorittamansa tehtävän parissa mahdollisimman nopeasti.

Viimeisenä ominaisuutena Nielsen (Nielsen, J. 1994) mainitsee käyttäjän subjektiivisen tyytyväisyyden. Järjestelmän tulisi olla miellyttävä, jotta käyttäjät pitäisivät järjestelmän käyttämisestä. Eli toisin sanoen järjestelmän käyttökokemuksen on tärkeää olla käyttäjälle miellyttävä, sekä käytettävyyden että esteettisyyden kannalta. Esteettisessä miellyttävyydessä tärkeitä osatekijöitä ovat esimerkiksi järjestelmän käyttöliittymässä käytettyjen värien välinen harmonia, kontrasti, sekä valkoisen tilan oikeaoppinen hyödyntäminen.

Näiden viiden ominaisuuden tueksi on myös määritetty 10 heuristiikkaa, joiden pohjalta käytettävyys tulisi saavuttaa käyttöliittymiä suunniteltaessa (Nielsen, J. 1994).

2.2 Prediktiiviset mallit

Prediktiiviset mallit (engl. *Predictive models, engineering models*) ovat laajalti ihmisen ja tietokoneen välisessä vuorovaikutuksessa käytettyjä sekä kehitettyjä monitieteisiä malleja. Näiden mallien tarkoituksena on ennustaa sekä mallintaa jonkun tietyn prosessin — ihmisen ja tietokoneen välisen vuorovaikutuksen tapauksessa erityisesti käyttöliittymän käyttäjän eli ihmisen toimintaa. Prediktiivisiä malleja on voitu luoda esimerkiksi ihmisen motorisen järjestelmän, kognitiivisen tai visuaalisen havaitsemisen toiminnoista. Prediktiivisiä malleja on HCI:ssä käytetty erilaisten mahdollisten suunnitteluskenaarioiden hypoteettiseen kokeilemiseen sekä tutkimiseen (Carroll 2003). Tässä kontekstissa käydään läpi vain tutkielman tavoitteen kannalta olennaisimmat, käyttöliittymäsuunnittelun käytännön sovellutuksien kannalta merkittävimmät prediktiiviset mallit. Näiden mallien tarkoituksena on kuvata muun muassa ihmisen motoristen kykyjen sekä kognitiivisten toimintojen, kuten päätöksen tekemiseen kuluva reaktioaika.

2.2.1 GOMS-malli

Tunnetuin HCI:ssä käytetty teoreettinen konsepti, GOMS (*Goals, Operators, Methods & Selection*), pyrkii mallintamaan ihmisen suorituskäytännön jakamalla sen osa-alueita eri muuttujiin (Bonnie, J. & Kieras, D. 1996). GOMS-mallin kehitti HCI:n pioneiritutkija Stuart Card kollegoineen jo 80-luvun alussa, ja sen tavoitteena oli korvata kalliita ja aikaavieviä oikeilla ihmisillä tehtäviä käyttäjätutkimuksia (Oulasvirta ym. 2018). Ihmisen ja teknologian välisen vuorovaikutuksen tutkimuksen edetessä on kehittynyt useita erilaisia GOMS -teoriaa hyödyntäviä malleja sekä muunnelmia eri malleista, joista tässä tutkielmassa keskitytään tarkastelemaan niistä kaikkein yksinkertaisinta. Tärkeää GOMS -malleja käytettäessä on kuitenkin ensin ymmärtää sen eri komponenttien merkitykset.

Tavoitteet (engl. *Goals*) kuvaavat sitä, mitä käyttäjä pyrkii toiminnallaan saavuttamaan. Monimutkaisemmissa tehtävissä tavoitteita voidaan yleensä jakaa myös osatavoitteisiin, jolloin tavoitteiden kuvaaminen on helpompaa. Esimerkiksi voidaan sanoa, että verkkopankissa asioidessa käyttäjän tavoitteena on ”maksaa lasku”, mutta tavoitetta voidaan jakaa vielä pidemmälle osatavoitteisiin, kuten ”Kirjaudu sisään” — tai mahdollisesti vielä pienempiin osatekijöihin, kuten ”Kirjoita salasana” tai ”Paina painiketta”.

Operaattorit (engl. *Operators*) selittävät käyttäjän toimintaa, jonka avulla käyttäjä toimii tavoitteiden saavuttamiseksi. Operaattorit voivat olla käyttäjän motorisia, kognitiivisia tai havaitsemiseen liittyviä toimintoja — tai näiden eri yhdistelmiä (Bonnie ym. 1996). Käyttäjän toimintaa kuvaavia tekijöitä ovat esimerkiksi hiiren liikuttaminen haluttuun suuntaan, tai näppäimistön painikkeen painaminen. Operaattoreita voidaan usein mitata suoritusajan avulla.

Metodeilla (engl. *Methods*) tarkoitetaan eri tekijöiden kokonaisuuksia, joilla käyttäjä pääsee tavoitteisiinsa. Menetelmät ovat siis käyttäjän toimintaa selittävien tekijöiden ketjutuksia, joilla voidaan suorittaa tietty tehtävä. Esimerkiksi tekstin kopioiminen tekstikentästä toiseen voisi olla käyttäjän tarvitsema menetelmä — käyttäjä joutuu ensin maalaamaan tekstin klikkaamalla hiiren vasenta painiketta, liikuttamaan hiir-

tä, klikkaamaan hiiren oikeaa painiketta sekä viimeiseksi valitsemaan ilmestyvästä alavetovalikosta oikean toiminnon hiiren vasenta painiketta painamalla.

Valintasäännöillä (engl. *Selection rules*) tarkoitetaan käyttäjän itsensä tekemiä, sisäisiä valintoja jotka vaikuttavat siihen minkälaisia metodeja hän käyttää tavoitteidensa saavuttamiseksi (Bonnie ym. 1996). Käyttäjä esimerkiksi voi edellä mainitun metodin sijaan käyttää metodia, jossa hän käyttää tekstin kopioimiseen hiiren sijaan pelkästään näppäimistön näppäinyhdistelmiä. Käyttäjän valintasääntöihin ja valintoihin vaikuttavat käyttäjän omat mieltymykset sekä tottumukset joihin hän on mukautunut käyttäessään erilaisia käyttöliittymiä.

2.2.2 KLM-malli

Ensimmäinen ja tunnetuin GOMS -teoriaa soveltava malli on käyttöliittymässä suoritettavan tehtävän kokonaissuoritusaikaa ennustava malli KLM (*Keystroke-Level-Model*). KLM on myös kaikista malleista yksinkertaisin ja helppoiten ymmärrettävä esimerkkitapaus.

KLM koostuu kuudesta eri operaattorista — neljästä motorisen hallinnan operaattorista, yhdestä mentaalista operaattorista sekä yhdestä järjestelmän vasteaikaa kuvaavasta operaattorista. Motorisia operaattoreita ovat näppäimen painallus (K, engl. *key stroking*), kursorin osoittaminen (P, engl. *pointing*), ohjautuminen (H, engl. *homing*) sekä viivojen piirtäminen hiirellä (D, engl. *drawing*) (Carroll, J. 2003). Lisäksi mentaalilla valmistautumisella joka on operaattoreista vaikeaselitteisin (Dix ym. 2009) tarkoitetaan käyttäjän psyykkistä tekemäänsä toimintoon valmistautumista (M), ja järjestelmän vasteajalla (R) tarkoitetaan parametreja jotka kuvaavat aikaa, joka käyttäjällä mahdollisesti kuluu järjestelmän vastaamisen odottamiseen.

$$T_{execute} = T_K + T_P + T_H + T_D + T_M + T_R. \quad (2.1)$$

KLM:ssä tavoitteena on laskea tarkasteltavan tehtävän kokonaissuoritus aika, joka saadaan yksinkertaisesti laskemalla yhteen kaikkien sen yksittäisten operaattorien

viemät suoritusajat. Kaavassa (2.1) muuttujien summa $T_{execute}$ kuvaa tehtävän kokonaissuoritusaikaa. Sitä sekä GOMS:in muita malleja on yksinkertaisuudestaan huolimatta käytetty käyttöliittymien suunnittelussa vielä nykyaikanakin, mutta ne soveltuvat parhaiten triviaalimpien järjestelmien käyttöliittymien suunnitteluun ja arviointiin. KLM:ää on käytetty käyttäjän suoriutumisen ennustamiseen muun muassa hierarkisten valikkorakenteiden suunnittelussa sekä pikanäppäinyhdistelmien kehittämisen apuna (Carroll, J. 2003). Ainoastaan suoritusaikaa mittaavissa malleissa on kuitenkin muistettava ettei suoritusajaltaan tehokkain käyttöliittymä tai käyttöliittymäkomponentti ole kaikissa tilanteissa paras vaihtoehto, vaan käyttäjän käyttökokemukseen liittyy useita muitakin tekijöitä.

2.2.3 Fittsin laki

Psykologi Paul Fitts kehitti 50-luvun alussa prediktiivisen mallin, jolla voidaan arvioida nopeasti tähdätyn liikkeen, kuten esimerkiksi hiiren kursorin liikituksen suoritusaikaa (MacKenzie 1992). Fittsin mukaan mitä suurempi ja mitä lähempänä kohde on, sitä helpommin käyttäjä tulee siihen osumaan. Käyttäjän suoriutumisista tehtävistä voitaisiin siis Fittsin lain mukaan helpottaa lisäämällä tähdättävien objektien, kuten erilaisten käyttöliittymän painikkeiden kokoa tai tuomalla niitä lähemmäs käyttäjän liikerataa. MacKenzien mukaan Fittsin laki on osittain johdettu tunnetusta Claude Shannonin informaatioteoreemasta, jonka mukaan käyttäjän keskittymiskykyä voidaan parantaa kasvattamalla tiedon osuutta ja vähentämällä käyttöliittymän hälyn osuutta.

$$MT = a + b \log_2\left(\frac{2A}{W} + K\right). \quad (2.2)$$

Kaavassa (2.2) esitetty MT (engl. *Movement time*) kuvaa tarkasteltavan liikkeen suorittamiseen kuluvaa aikaa, A on etäisyys ja W kohteen leveys. Muuttujat a ja b ovat tilanteesta, kuten ohjaimen ominaisuuksista, tai käyttäjän fyysisistä ominaisuuksista riippuvia tekijöitä jotka mahdollisesti vaikuttavat tulokseen. Fittsin laki on todistettu toimivaksi erilaisissa käyttötilanteissa (MacKenzie, S. 1992), ja siitä on myös

myöhemmin kehitelty tarkempia versioita.

2.2.4 Hick-Hymanin laki

Hick-Hymanin laki on psykologien William Edmund Hick sekä Ray Hymanin tutkimuksista kehitetty prediktiivinen malli, joka kuvaa aikaa joka henkilöllä kuluu päätöksen tekemiseen, kun valinta on tehtävä rajallisesta joukosta eri vaihtoehtoja. Mitä enemmän käyttäjälle annetaan informaatiota eli eri vaihtoehtoja, sitä kauemmin käyttäjällä kuluu päätöksen tekemiseen — malli siis ennustaa lineaarista suhdetta reaktioajan sekä käyttäjälle välitetyn tiedon välillä (Seow 2005).

$$RT = a + bH_T. \quad (2.3)$$

Hick-Hymanin lain määritelmässä RT on käyttäjän reaktioaika, ja muuttujat a ja b ovat empiirisesti päätettyjä muuttujia. H_T on käyttäjälle välitetty tieto, jota voidaan kuvata tarkemmin Shannon-Weinerin kehittämällä informaation mittakaavalla (Seow, S. 2005). Hick-Hymanin laki ei ole ollut yhtä keskusteltu ihmisen ja tietokoneen välisen vuorovaikutuksen tutkimuksen prediktiivisenä mallina kuin aiemmin mainittu Fittsin laki, mutta se on silti hyödyllinen malli ihmisen käyttäytymisen perustoimintoja ennustettaessa.

3 Kombinatorinen optimointi

Kombinatorisella optimoinnilla tarkoitetaan optimointitehtäviä, jotka ovat usein vaikeita ratkaistavia. Tällaisissa optimointitehtävissä tarkoituksena on löytää jokin optimaalinen ratkaisu erilaisten ratkaisujen äärellisestä joukosta. Jos n on muuttujien lukumäärää kuvaava kokonaisluku, kombinatorisissa optimointitehtävissä on yleensä $n!$ tai 2^n erilaista ratkaisumahdollisuutta (Haataja 2004). Kombinatorisen optimoinnin keinoin voidaan muun muassa laskea optimaalisin reitti julkisen liikenteen kulkuneuvolle, tai löytää ratkaisu erilaisiin aikataulutusesongelmiin.

Kenties tunnetuin kombinatorisilla optimointimenetelmillä ratkaistava optimointiongelmaksi on kauppatkustajan ongelma (engl. *Traveling salesman problem*), joka on NP-täydellinen ongelma. Kauppatkustajan ongelmassa tavoitteena on löytää optimaalinen reitti kaupunkien rajatun joukon välillä kaupunkien väliset etäisyydet huomioiden, niin että kaupunkien välillä kulkeva "kauppatkustaja" kulkee jokaisen kaupungin kautta vähintään kerran.

Tässä tutkielmassa keskitytään tarkastelemaan käyttöliittymien sekä ihmisen ja tietokoneen välisen vuorovaikutuksen suunnittelua kombinatorisen vaikeana optimointitehtävänä. Vaikean tehtävästä tekee se, että käyttöliittymän on toteutettava useampia vaatimuksia sekä tavoitteita samaan aikaan, yhdessä yhtenäisesti toimivassa kokonaisuudessa. Tällaisia vaatimuksia ovat järjestelmäkohtaiset vaatimukset, sekä aiemmissa luvuissa esiteltyt, käyttöliittymäsuunnittelussa noudatettavat hyvän käytettävyyden periaatteet.

3.1 Geneettiset algoritmit

Vaikeiden kombinatoristen optimointitehtävien ratkaisemiseen soveltuu useita eri ratkaisumenetelmiä, mutta tässä alaluvussa keskitytään tutkielman aiheen kannalta olennaisimpiin, geneettisiin algoritmeihin. Geneettiset algoritmit (GA, engl. *Genetic Algorithms*) vaativat tehtäväkohtaista määrittelyä ja virittelyä, jotta optimoinnissa voitaisiin päästä luotettavaan lopputulokseen (Haataja, J. 2004). Geneettiset algorit-

mit kuuluvat laajempaan evoluutiopohjaisten algoritmien käsitteeseen, ja ne ovat nimensä mukaisesti luonnossa tapahtuvaa evoluutiota matkivia algoritmeja, jotka mukautuvat ja muuttuvat optimointitehtävän aikana.

Geneettisten algoritmien toiminta perustuu luonnonvalintaan, jossa perinnölliset hyödylliset ominaisuudet säilyvät ja vähemmän hyödylliset ominaisuudet harvinaistuvat lajin populaatiossa sukupolvien välillä. Näitä algoritmeja on yleisesti käytetty optimointitehtävissä, joissa mahdollisten ratkaisujen joukko on erittäin suuri (Haataja, J. 2004). Geneettisten algoritmien tavoitteena on löytää mahdollisten ratkaisujen joukosta kohtuullisen hyviä ratkaisuja, eikä välttämättä vain yhtä kaikkein optimaalisinta ratkaisua. Muun muassa aiemmin mainittu NP-täydellinen kauppa-matkustajan ongelma olisi mahdollista ratkaista geneettisellä algoritmilla.

Keskeistä geneettisten algoritmien toiminnassa on, että ratkaisuavaruuden vektorit, eli ratkaisujoukon mahdolliset ratkaisut kuvataan sopivalla tavalla "geneettiseksi koodiksi" eli "yksilöksi", jota voidaan kuvata muuttujalla x .

Alla on kuvattu Haatajan (Haataja, J. 2004) teoksessa esitetty yksinkertainen pseudokoodinen esimerkki geneettisestä algoritmista, jonka tarkoituksena on avata geneettisten algoritmien toiminnan perusidea:

```
luo alkupopulaatio  $P_1 = x^i, i = 1, \dots, l$ ;  
for  $k = 1, 2, \dots, m$  do  
    risteytä populaation  $P_k$  yksilöitä keskenään;  
    generoi mutaatioita populaatiossa  $P_k$  ;  
    if löytyy tarpeeksi hyvä ratkaisu  $X \in P_k$  then  
        aseta  $x^* = x$  ja lopeta iterointi;  
    else  
        valitse populaatiosta  $P_k$  parhaat yksilöt populaatioksi  $P_k + 1$  ;  
    end  
end
```

Aluksi valitaan alkupopulaatio P_1 , joka edustaa yksilöiden vektorijoukkoa $x^i, i = 1, \dots, l$. Alkupopulaation valinnan jälkeen kaksi ennalta valitulla mekanismilla valit-

tua yksilöä risteytetään keskenään. Tämän jälkeen risteytyksestä saatua "jälkeläistä" muutetaan satunnaisen mutaation avulla. Alkioita x voi siis verrata kromosomeihin, ja kukin arvo x^j , $i = 1, \dots, n$ kuvastaa yhtä geeniä. Hyvien ominaisuuksien vertaamiseksi tarvitaan hyvyysfunktio $f(x)$, jolle pyritään saamaan mahdollisimman korkea arvo. Geneettisten algoritmien toiminta perustuu siis luonnonvalintaan sekä risteytysmekanismien hyödyntämiseen (Haataja, J. 2004).

Geneettisiä algoritmeja on hyödynnetty käyttöliittymäoptimointiin muun muassa Paulo Salemin tutkimuksessa (Salem 2017), jossa geneettisiä algoritmeja käytettiin onnistuneesti verkkosivuston aloitussivun suunnittelun optimointiin. Salemin tutkimuksessa todistettiin, että geneettisten algoritmien avulla on mahdollista koostaa toimiva käyttöliittymä pienemmistä, erikseen optimoiduista komponenteista. Valikkohierarkian optimointiin on myös pystytty todistetusti hyödyntämään geneettisten algoritmien menetelmiä (Matsui, S. & Yamada, S. 2008).

Geneettisiä algoritmeja eli geneettistä ohjelmointia on hyödynnetty myös muilla suunnittelun osa-alueilla, kuten 3d-mallinnuksessa. Suunnittelun tukena hyödynnettävistä, evoluutiopohjaisista algoritmeista käytävistä tekniikoista käytetäänkin yleisesti nimitystä "Evolving design", kehittyvä suunnittelu.

3.1.1 Interaktiiviset geneettiset algoritmit

Käyttöliittymien optimointiin on onnistuttu hyödyntämään myös interaktiivisia geneettisiä algoritmeja (IGA, engl. *Interactive Genetic Algorithms*) (Quiroz 1999). Interaktiiviset geneettiset algoritmit ovat geneettisten algoritmien alalaji, jossa käyttäjän antamaa informaatiota käyttämällä voidaan vuorovaikuttaa käyttöliittymän vaikeasti määriteltävien optimointitavoitteisiin — esimerkiksi verkkosivustolla sijaitsevien kuvien tai värien estetiikan muokkaamiseen käyttäjän preferenssien mukaisesti. Käyttäjän osoittaessa algoritmille sekä hyviä että huonoja yksilöitä mahdollisten ratkaisujen populaatiosta, hyvien yksilöiden ominaisuuksia siirretään sukupolville eteenpäin, ja huonojen yksilöiden ominaisuuksia jätetään pois.

4 Käyttöliittymien vuorovaikutuksen optimointi

Käyttöliittymäoptimoinnin tutkimus on edennyt kohtalaisen hitaasti. Vielä 2000-luvun alussa näppäimistöjen näppäinten sijoitteluongelma on ollut ainoa tutkimuskohde käyttöliittymäoptimoinnin saralla (Oulasvirta 2017). Käyttöliittymiä suunniteltaessa sekä optimoidessa tarvitaan tietoa käyttäjästä — tähän voidaan käyttää tutkielmassa aiemmin mainittuja HCI:n alalla käytettyjä prediktiivisiä malleja sekä vuorovaikutussuunnittelun yleisiä ohjenuoria ja heuristiikoita. Prediktiivisiä malleja voidaan käyttää käyttöliittymän ja käyttäjän välisen vuorovaikutuksen ennakoimiseen sekä simuloimiseen, ja täten vaikuttaa optimoinnin tuloksiin. HCI:n hyvää suunnittelua ja käytettävyyttä tukevia heuristiikkoja puolestaan voidaan hyödyntää optimoinnissa tarvittavien rajoitteiden sekä tavoitteiden tarkempaan määrittelyyn.

4.1 Suunnitteluongelman määrittely

Jotta käyttöliittymien optimointi olisi mahdollista, on optimointialgoritmillemme ensin määriteltävä suunnitteluongelma. Suunnitteluongelman määrittely koostuu kolmesta osasta — päätösmuuttujista, rajoitteista sekä tavoitefunktioista (Oulasvirta, A. 2017).

Päätösmuuttujilla (engl. *Decision variable*) tarkoitetaan muuttujia, joille optimointialgoritmi hakee optimaalista arvoa optimointitehtävän aikana. Siis toisin sanoen, optimointialgoritmi manipuloi päätösmuuttujia niin, että ne tuottaisivat optimaalisimman arvon tavoitefunktioille. Käyttöliittymäoptimoinnin tapauksessa päätösmuuttujat voivat olla esimerkiksi jonkin käyttöliittymään sisältyvän elementin, kuten esimerkiksi tietyn valikkorakenteen sijaintia tai kokoa kuvaavia muuttujia. Päätösmuuttujat ovat siis arvoja, joiden yhdistelmästä optimaalinen tulos muodostuu.

Rajoitteet (engl. *Constraints*) ovat optimointitehtävälle asetettuja rajoitteita (Oulasvirta ym. 2018). Esimerkiksi Quasi-Qwerty-näppäimistön kirjoitusnopeuden optimoinnissa (Bi, Smith & Zhai 2010) rajoitteisiin kuului muun muassa sääntö, jonka mukaan näppäinten sijainteja ei saanut muuttaa alkuperäisen Qwerty-näppäimistön

asettelusta kuin yhden näppäimen verran. Rajoitteiden avulla hallitaan optimointitehtävän lopputulosta.

Edellä mainitut kaksi ominaisuutta yhdistyvät tavoitefunktiossa (engl. *Objective function*). Tavoitefunktio on funktio, jonka tarkoituksena on päätösmuuttujien sekä annettujen rajoitteiden avulla antaa optimaalinen, tavoitteen mukainen tulos (Oulasvirta ym. 2018). Optimaalinen tulos on optimointitehtävästä riippuen joko minimaalinen tai maksimaalinen numeerinen arvo, joka saadaan laskettua rajoitteiden ja päätösmuuttujien avulla.

Lähestulkoon kaikissa käyttöliittymäsuunnittelun optimointitehtävissä täytyy ottaa huomioon useampi kuin yksi optimointitavoite, sillä suurimman osan käyttöliittymistä on toteutettava enemmän kuin vain yksi käyttötarkoitus. Tavoitteet voivat olla ristiriidassa keskenään, jonka takia tavoitteita ei saada usein määriteltyä yksiselitteisesti, vaan joudutaan tyytymään kompromisseihin (Oulasvirta ym. 2018). Monitavoitteisella optimoinnilla (engl. *Multi-objective optimization*) tarkoitetaan useamman tavoitteen tiivistämistä yhteen tavoitefunktioon.

Optimaalisella suunnittelutuloksella tarkoitetaan täysin optimaalista ratkaisua — yksiselitteisesti parasta vaihtoehtoa rajattujen käyttöliittymäehdotusten joukosta. Jos ratkaisu on likimääräisesti optimaalinen, sillä tarkoitetaan että sen tavoitearvo ylittää tietyn rajan, ja se voidaan täten hyväksyä optimaalisten ratkaisujen joukkoon (Oulasvirta ym. 2018). Suunnitteluongelman huolellinen määrittely on tärkeää, sillä ilman huolellista määrittelyä optimointimetodien antamia tuloksia ei voida pitää valideina. Lopullisen optimaalisuuden käyttöliittymälle kuitenkin määrittää käyttäjäkokemus, toisin sanoen kuinka käyttäjät kokevat järjestelmän käyttämisen hyödyt. Olennaista optimaalisen suunnittelutuloksen saavuttamiseen on optimointimenetelmien lisäksi siis myös suunnittelijan osallisuus sekä empiiriset kokeilut oikeilla käyttäjillä.

4.2 Tyypillisimmät suunnitteluongelmat

Tämän alaluvun tarkoituksen on esitellä edellä käsiteltyyn määrittelyyn vaikuttavia käyttöliittymäoptimoinnin tavallisimpia suunnitteluongelmia, joita myös Oulasvirta ja Karrenbauer (Oulasvirta ym. 2018) ovat käsitelleet suunnitteluongelmien neljänä tärkeimpänä tyyppinä. Käyttöliittymäoptimoinnissa keskitytään olennaisesti aina yksittäisten suunnitteluongelmien ratkaisemiseen.

Valintaongelmat (engl. *Selection problems*) ovat ongelmia, joissa tavoitteena on valita ennalta määrätyn valikoiman joukosta oikea yhdistelmä elementtejä. Ongelma, jossa suunnittelijan täytyy valita paras yhdistelmä erilaisia käyttöliittymäelementtejä, kuten alavetovalikoita ym. on esimerkki tällaisesta suunnitteluongelmasta. Elementtejä voidaan arvioida vertaamalla niiden attribuutteja ennalta määritettyihin vaatimuksiin — esimerkiksi käytettävyys tai opittavuus voivat olla tietyn elementin eri arvoja saavia attribuutteja, joita voidaan verrata suunnittelijan antamiin vaatimuksiin.

Järjestysongelmilla (engl. *Ordering problems*) tarkoitetaan tilannetta, jossa tavoitteena on löytää annetuille elementeille optimaalinen järjestys. Tällaisesta ongelmasta kenties tunnetuin variaatio on tutkielmassa aikaisemmin esitelty kauppamatkustajan ongelma (Oulasvirta ym. 2018). Käyttöliittymän valikkorakenteen elementtien sijoittaminen kaikkein nopeimmin luettavaan järjestykseen on myös esimerkki järjestysongelmasta.

Sijoitusongelmia (engl. *Assignment problems*) ovat ongelmat, joissa on n määrä sijoitettavia elementtejä, sekä n määrä sijainteja joihin elementit voidaan sijoittaa. Tavoitteena on löytää elementeille optimaaliset paikat vaatimusten mukaisesti.

Aikataulutusetongelmat (engl. *Scheduling problems*) ovat ongelmia, joissa tehtäviä määrätään tietyille resursseille, tietyille ajoille (Oulasvirta ym. 2018). Esimerkkinä aikataulutusetongelmasta toimivat työntekijöiden työaikataulujen optimointi mahdollisimman tehokkaiksi, ja käyttöliittymäsuunnittelun yhteydessä taas esimerkiksi käyttäjälle näytettävien notifikaatioiden aikataulutus.

4.3 Vuorovaikutuksen optimoinnin hyödyt

Käyttöliittymäoptimoinnin tarkoituksena on vähentää suunnittelijan manuaalista työtä, ehkäistä ihmissuunnittelijan tekemiä virheitä sekä antaa matemaattisesti perusteltavia, faktatietoon pohjautuvia suunnitteluratkaisuja vaativiin suunnitteluongelmiin. Vuorovaikutus- tai käyttöliittymäsuunnittelijan työ sisältää tavallisesti paljon iterointia. Käyttöliittymäsuunnitelmia joudutaan usein muuttamaan ja muokkaamaan kehityksen aikana, ja hyvin usein myös järjestelmän julkaisun jälkeenkin. Virheiden minimointi on suunnittelijan työssä myös erityisen tärkeää — esimerkiksi terveydenhuollon järjestelmissä tai muissa mahdollisesti ihmisten turvallisuuteen liittyvissä käyttöliittymissä suoritus aika sekä virheidenhallinta täytyisi olla optimaalista.

Optimoinnin avulla on siis mahdollista saavuttaa nopeammin parempia tuloksia. On kuitenkin tärkeä pitää suunnittelija optimointiprosessissa mukana, jotta suunnittelija voi muokata päätösmuuttujia sekä rajoitteita tarvittaessa. Muun muassa Salemin geneettisten algoritmien avulla tuotetussa optimoinnissa (Salem, P. 2017) optimointialgoritmi saattoi välillä ehdottaa hyvien ratkaisujen lisäksi myös huonoja ratkaisuja mutaatiomenetelmien takia. Salem painottaakin tutkimuksessaan suunnittelijan vastuuta optimointiprosessin aikana. Virheitä optimointiprosessissa voitaisiinkin välttää tuomalla optimointiprosessiin jo peruselementeiltään valmis, suunnittelijan ehdottama käyttöliittymä, jota optimoinnin avulla kehitettäisiin yksityiskohdiltaan paremmaksi.

Vielä ainakin nykyaikana vuorovaikutuksen ja käyttöliittymien optimointi onkin jo olemassa olevien, suunnittelijan tekemien suunnittelupäätösten parantamista (Oulasvirta ym. 2018). Suunnitteluongelmien kombinatorisella formuloinnilla voidaan löytää uusia näkökulmia sekä suuntaa antavia suosituksia suunnittelupäätöksien tueksi — optimoinnin avulla pystyttäisiin esimerkiksi määrittämään kuinka kaukana nykyinen suunniteltu käyttöliittymä tai ominaisuus on tavoitearvoista, ja minkälaisen askeleiden kautta optimaalisempaan ratkaisuun voitaisiin päätyä.

Oulasvirta (Oulasvirta, A. 2017) mainitsee tekstissään myös vuorovaikutuksen op-

timoinnin haasteita. Suunnitteluongelmien määrittely ja muuttaminen matemaattiseen muotoon on ainakin toistaiseksi vielä hankalaa, ja ratkaistavien suunnitteluongelmien joukko rajoittuukin vielä vain yleisimpiin suunnitteluongelmiin, kuten aikaisemmin mainittuihin järjestys- tai sijoitusongelmiin. Prediktiivisten mallien avulla on voitu saada paljon tietoa käyttäjän vuorovaikutuksesta, mutta esimerkiksi sosiaalisten, kulttuuristen tai muiden abstraktimpien tekijöiden vaikutusta vuorovaikutukseen on ollut toistaiseksi vaikeaa mallintaa. Myös kysymys siitä, kuinka paljon suunnittelijalle tai algoritmille pitäisi antaa päätösvaltaa optimointiprosessissa, on aiheellinen.

Muun muassa valikkohierarkioita on kuitenkin onnistuttu optimoimaan onnistuneesti kombinatorisen optimoinnin menetelmiä hyödyntäen. Tästä esimerkkinä on Bailey'n ym. MenuOptimizer, graafisia käyttöliittymien suunnitteluun tarkoitettuun QtDesigner-työkalun tueksi kehitetty lisäosa. (Bailly, G., Oulasvirta, A., Kötzing, T. & Hoppe, S. 2013). MenuOptimizer koettiin tehokkaaksi välineeksi valikkohierarkioiden luomiseen — tutkimuksen yhteydessä tehty käyttäjätutkimus osoitti, että optimointityökalun avulla luotuja toimivia valikkohierarkioita saatiin aikaan 38% nopeamassa ajassa, kuin suunnitteluprosessissa jossa työkalua ei ollut käytössä. Yhtenä tärkeänä sovellutuksena onkin esitetty optimoinnin yhdistämistä jo valmiiksi olemassa oleviin suunnittelutyökaluihin.

5 Yhteenveto

Tässä tutkielmassa selvitettiin minkälaisia kombinatorisen optimoinnin menetelmiä vuorovaikutussuunnittelun optimoinnissa on tutkittu, ja minkälaisia hyötyjä käyttöliittymien algoritmisesta optimoinnista on saatu. Tutkielmassa on käyty läpi hyvän vuorovaikutussuunnittelun periaatteita, sekä esimerkkejä käyttöliittymäoptimoinnin tukena hyödynnettävistä prediktivisistä malleista. Myös käyttöliittymäoptimoinnin kannalta tärkeimpiä kombinatorisen optimoinnin menetelmiä, geneettisiä algoritmeja on esitelty yleisellä tasolla sekä avattu tutkielman aiheeseen soveltuvien menetelmien toimintaa.

Tutkielmassa selvisi, että aiheesta on tehty vielä vähäisesti tutkimusta. Vuorovaikutussuunnittelun optimoinnin suurimpina haasteina on esiintynyt suunnitteluongelmien formaalisen määrittelyn hankaluus, jonka takia suunnitteluongelmien implementointi ohjelmallisesti on ollut vaikeaa. Tutkielmassa esitellyt toteutukset keskittyvät yksinkertaisimpien suunnitteluongelmien ratkaisemiseen, eikä aineistoa monimutkaisempien järjestelmien optimoinnin hyödyistä ole esitetty. Myös prediktivisten mallien empiirinen validiteetti sekä pitkäikäisyys ovat nousseet tärkeinä kysymyksinä esiin, ja ne on otettava huomioon menetelmien luotettavuutta arvioitaessa. Tästä voidaan päätellä, että ihmisen ja tietokoneen välistä vuorovaikutusta kuvaavia malleja on kehitettävä eteenpäin sekä nykyajan digiyhteiskuntaa enemmän vastaaviksi.

Haasteista sekä vaikeuksista huolimatta huomattavaa edistystä digitaalisten käyttöliittymien käytettävyyden parantamiseen optimointimenetelmiä käyttäen on kuitenkin tapahtunut. Kombinatorisen optimoinnin hyödyntämistä vuorovaikutussuunnittelussa on pääosin tutkittu vasta viime vuosien aikana, mutta tähän saakka aikaisempi tutkimus on keskittynyt fyysisten käyttöliittymien, kuten näppäimistöjen ulkoasun sekä kirjoitusnopeuden optimointiin. Optimoinnilla oli saavutettu huomattavia etuja muun muassa valikkohierarkioiden suunnittelussa. Optimoinnin hyötyjä oltiin myös pystytty todistamaan käyttäjätutkimuksen avulla kerätyillä tiedoilla. Abstraktien käytettävyyssominaisuuksien esittämiseen oli esitetty myös ratkaisua

interaktiivisten geneettisten algoritmien hyödyntämisestä, jossa vuorovaikutuksesta sekä preferensseistä saadaan tietoa suoraan käyttäjältä. Geneettisten algoritmien menetelmät ovatkin luultavasti kiinnostavia jatkokehityksen kohteita käyttöliittymäoptimoinnin tulevissa tutkimuksissa.

Vuorovaikutus- ja käyttöliittymäsuunnittelun optimoinnissa on kuitenkin yhä paljon tutkittavaa, vaikka selvää kehitystä tutkimuksen alalla onkin huomattavissa. Tärkeäksi keskustelunaiheeksi jäävät myös perinteisten suunnittelumetodien ongelmat, sekä se, miten ihmisen tekemää suunnittelua voitaisiin muokata algoritmisten optimointimenetelmien avulla paremmaksi. Ideaalitapauksessa optimointimenetelmiä voitaisiin integroida suoraan suunnittelijan työkaluihin, antamaan suunnittelijalle suosituksia sekä opastusta käyttöliittymien mahdollisimman suoraviivaiseen sekä tehokkaaseen suunnitteluun.

Kirjallisuutta

- Bailly, G., Oulasvirta, A., Kötzing, T., Hoppe, S. 2013. *MenuOptimizer: interactive optimization of menu systems*. UIST '13 Proceedings of the 26th annual ACM symposium on User interface software and technology, s. 331–342.
- Bi, X., Smith, A. & Zhai, S. *Quasi-qwerty soft keyboard optimization*. CHI , 2010.
- Bonnie, J. & Kieras, D. 1996. *The GOMS Family of User Interface Analysis Techniques: Comparison and Contrast*. ACM Transactions on Computer-Human Interaction, Vol. 3, No. 4, December 1996.
- Carroll, J. 2003. *HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science*. SF, USA: Elsevier Science.
- Dix, A., Finlay, J., Abowd, G. & Beale, R. 2004. *Human-Computer Interaction*. Harlow, England: Pearson Education Limited.
- Haataja, J. 2004. *Optimointitehtävien ratkaiseminen*. Helsinki: Tieteen tietotekniikan keskus CSC.
- Haataja, J., Heikonen, J., Leino, Y., Rahola, J., Ruokolainen, J. & Savolainen, V. 2002. *Numeeriset menetelmät käytännössä*. Helsinki: Tieteen tietotekniikan keskus CSC.
- MacKenzie, S. 1992. *Fitts' law as a research and design tool in Human-Computer Interaction*. Human-computer interaction, Volume 7, s. 91–139.
- Matsui, S. & Yamada, S. *Genetic Algorithm Can Optimize Hierarchical Menus*. CHI 2008 Proceedings. Menu and Command Selection April 5-10, 2008.
- Nielsen, J. 1995. *10 Usability Heuristics for User Interface Design*.
- Nielsen, J. 1999. *Designing Web Usability: The Practice of Simplicity*. CA, USA: New Riders Publishing Thousand Oaks.
- Nielsen, J. 1990. *Heuristic evaluation of user interfaces*. CHI '90 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, s. 249–256.
- Nielsen, J. 1994. *Usability Engineering*. Cambridge, MA: AP Professional.
- Oulasvirta, A., Kristensson, P., Bi, X. & Howes, A. 2018. *Computational Interaction*. Oxford: Oxford University Press.
- Oulasvirta, A. 2017. *User Interface Design with Combinatorial Optimization*. Computer, s. 40–47.

- Salem, P. 2005. *User Interface Optimization using Genetic Programming with an Application to Landing Pages*. Proc. ACM Hum.-Comput. Interact., Vol. 1, No. 1, Article 12.
- Seow, S. 2005. *Information Theoretic Models of HCI: A Comparison of the Hick-Hyman Law and Fitts' Law*. Human-computer interaction, Volume 20, s. 315–352.
- Quiroz, J., Louis, S., Shankar, A. & Dascalu, S. 2007. *Interactive Genetic Algorithms for User Interface Design*. 2007 IEEE Congress on Evolutionary Computation, Singapore, 2007, s. 1366–1373