

Monitavoiteoptimoinnin ja erityisesti NIMBUS-menetelmän
hyödyntäminen monitavoitteisessa päätöksenteossa.

Ellemari Teinilä

Pro gradu -tutkielma
Heinäkuu 2019

MATEMATIIKAN JA TILASTOTIETEEN LAITOS
TURUN YLIOPISTO

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

TURUN YLIOPISTO
Matematiikan ja tilastotieteen laitos

TEINILÄ, ELLEMARI
Pro gradu -tutkielma, 37 s.
Matematiikka
Kesäkuu 2019

Työssä on käsitelty monitavoiteoptimointia yleisesti, erilaisia monitavoiteoptimointiongelmien ratkaisumenetelmiä, sekä erityisesti NIMBUS-menetelmän hyödyntämistä ongelmanratkaisussa.

Työssä on käsitelty aluksi monitavoiteoptimoinnin keskeisiä käsitteitä sekä optimointiongelmien muotoa ja ratkaisua. Tämän jälkeen on esitelty erilaisia monitavoitteisten optimointiongelmien ratkaisumenetelmiä. Ratkaisumenetelmät on jaoteltu neljään osaan riippuen siitä, millaisessa roolissa päätöksentekijä ratkaisuprosessissa on. Jokaisesta menetelmäosasta on valittu yksi ratkaisumenetelmä, jonka toimintaperiaate on esitetty. Ratkaisumenetelmistä on nostettu esille erityisesti NIMBUS-menetelmä, jonka toimintaperiaate on esitetty tarkemmin ja jonka avulla on ratkaistu monitavoiteoptimointiongelma.

Työssä on lisäksi esitetty sovelluksia, joissa NIMBUS-menetelmää on hyödynnetty. Aiheeksi on valittu paperin valmistukseen liittyvät monitavoiteongelmat, joista toinen kuvaa paperikoneen vedenjakeluongelmaa ja toinen paperintuotantoongelmaa. Molempien monitavoiteoptimointiongelmien ratkaisussa päätöksentekijä on ollut keskeisessä roolissa.

Asiasanat: monitavoiteoptimointi, optimointimenetelmä, NIMBUS-menetelmä, paperikone, paperintuotanto.

Sisältö

| | | |
|----------|--|-----------|
| 1 | Johdanto | 3 |
| 2 | Monitavoiteoptimointitehtävistä | 4 |
| 2.1 | Peruskäsitteitä | 4 |
| 2.2 | Monitavoiteoptimointiongelma | 4 |
| 2.3 | Monitavoiteoptimointiongelman ratkaisu | 5 |
| 2.4 | Pareto-optimaalisuus | 6 |
| 3 | Monitavoiteoptimointitehtävien ratkaisumenetelmiä | 9 |
| 3.1 | Päätöksentekijän mielipidettä ei huomioida | 10 |
| 3.1.1 | Globaalin tavoitteen menetelmä | 10 |
| 3.2 | Päätöksentekijän mielipiteet huomioidaan etukäteen | 11 |
| 3.2.1 | Painokerroinmenetelmä | 11 |
| 3.3 | Päätöksentekijän mielipiteet otetaan huomioon jälkikäteen | 12 |
| 3.3.1 | Rajoiteyhtälömenetelmä | 12 |
| 3.4 | Interaktiiviset menetelmät: päätöksentekijän ja ratkaisumenetelmän välistä vuorovaikutusta | 13 |
| 3.4.1 | ISWT (Interactive Surrogate Worth Trade-Off) | 14 |
| 4 | NIMBUS | 15 |
| 4.1 | NIMBUS-menetelmä yleisesti | 15 |
| 4.1.1 | NIMBUS-menetelmän keskeisiä käsitteitä | 15 |
| 4.2 | NIMBUS-menetelmän aputehtävät | 16 |
| 4.3 | NIMBUS-algoritmi | 18 |
| 4.4 | Yksinkertaisen optimointiongelman ratkaisu | 20 |
| 4.4.1 | Optimointiongelman ratkaisu erilaisilla luokittelutiedoilla | 24 |
| 5 | Monitavoiteoptimoinnin hyödyntäminen paperin valmistuksessa | 28 |
| 5.1 | Paperikoneesta | 28 |
| 5.2 | Monitavoiteoptimointi ja paperikone | 29 |
| 6 | NIMBUS-menetelmä paperinvalmistuksessa | 30 |
| 6.1 | Vedenjakeluongelman asettelu | 30 |
| 6.1.1 | Ongelmaratkaisuprosessi | 31 |

| | | |
|----------|------------------------------------|-----------|
| 6.2 | Paperintuotanto-ongelma | 33 |
| 6.2.1 | Ongelmanratkaisuprosessi | 34 |
| 7 | Päätelmät | 36 |
| | Kirjallisuutta | 37 |

1 Johdanto

Useimmat todelliset päätöksentekotilanteet koostuvat monista eri tavoitteista, joiden väliltä halutaan löytää paras mahdollinen ratkaisu. Tavoitteet saattavat olla keskenään ristiriitaisia ja päätöksen syntymiseen vaikuttavat vahvasti päätöksentekijän omat mieltymykset ja arvotusjärjestelmät. Esimerkiksi yrityksen logistiikan suunnittelussa tulee ottaa huomioon sekä tuotteen kuljetuksen kustannukset että mahdollisimman ripeä aikataulu. Tuote halutaan toimittaa asiakkaalle mahdollisimman edullisesti, mutta samaan aikaan myös mahdollisimman nopeasti. Jos yritys valmistaa esimerkiksi pihakeinuja, tulee myös niiden valmistuksessa optimoida monia eri tavoitteita. Tavoitteena voi olla esimerkiksi tuotantokustannusten ja tuotantoajan minimointi, sekä laadukkuuden ja ulkonäön maksimointi. Tällaisien monista eri tavoitteista koostuvien ongelmien ratkaisua kutsutaan *monitavoiteoptimoinniksi*.

Monitavoiteoptimointitehtävissä kohdefunktioita on enemmän kuin yksi, jolloin optimiratkaisu ei ole yksikäsitteinen, kuten yhden tavoitefunktion tapauksissa. Monitavoiteoptimoinnissa myös esimerkiksi päätöksentekijän mielipide voi vaikuttaa lopputulokseen. Onkin olemassa eri tyyppisiä monitavoiteoptimointimenetelmiä sen mukaan, millaisessa roolissa päätöksentekijä niissä on. Joissakin menetelmissä päätöksentekijän mielipidettä ei oteta olenkaan huomioon, kun taas joissakin tapauksissa päätöksentekijä on ratkaisuprosessin kanssa jatkuvassa vuorovaikutuksessa. Erilaiset monitavoiteoptimointitehtävien ratkaisumenetelmät ovat keskeisiä yhteiskunnan toimivuuden ja kehityksen kannalta.

Tässä työssä käsitellään monitavoiteoptimointia yleisesti ja käydään läpi erilaisia monitavoiteoptimointiongelmiä ratkaisumenetelmiä. Eriteltyt menetelmät on valittu siten, että menetelmät ovat päätöksentekijän roolin kannalta erilaisia. Työssä syvennytään erityisesti NIMBUS-menetelmään, jossa päätöksentekijä on jatkuvasti mukana ratkaisuprosessissa. Menetelmän avulla ratkaistaan monitavoiteoptimointiongelma kahdessa eri tapauksessa, jolloin pystytään tuomaan esille päätöksentekijän vaikutus saatuihin ratkaisuihin. Lisäksi esitellään muutama paperinvalmistukseen liittyvä monitavoiteoptimointiongelma, joiden ratkaisussa on myös hyödynnetty NIMBUS-menetelmää.

2 Monitavoiteoptimointitehtävistä

2.1 Peruskäsitteitä

Aluksi esitellään muutamia optimoinnin keskeisiä termejä ja käsitteitä, jotka esiintyvät tekstissä myöhemmin, ja joiden pohjalta optimointiongelmat muodostetaan.

- *Muuttujien arvo* optimitilanteessa halutaan selvittää.
- *Kohdefunktio* kuvaa tavoitetta, jota halutaan minimoida tai maksimoida. Kohdefunktio ilmoittaa, kuinka optimoitava suure lasketaan muuttujien avulla.
- *Rajoitteet* määrittävät sallitun alueen S , jolta ratkaisun tulee löytyä.
- *Hyötyfunktio* on eri tavoitteiden reaaliarvoinen funktio.
- *Trade-Off* kuvaa kohdefunktioiden arvojen välistä suhdetta/ tasapainoa; kun jonkin kohdefunktion arvo laskee, se mahdollistaa jonkin toisen kohdefunktion arvon nousun (ja päinvastoin).
- *Päätöksentekijä* on henkilö, joka voi kertoa ristiriitaisiin tavoitteisiin liittyvää tietoa mieltymyksistään, ja voi siten vaikuttaa omien arvojensa, mielipiteidensä ja päätöstensä kautta ongelman ratkaisuun. Matemaattisesti kaikki saadut ratkaisut ovat yhtä hyviä, joten tällöin tarvitaan päätöksentekijä valitsemaan niiden joukosta parhaat ratkaisut.

2.2 Monitavoiteoptimointiongelma

Monitavoiteoptimointiongelma voidaan esittää seuraavasti:

$$\begin{array}{ll} \text{minimoi} & f(x) = \{f_1(x), f_2(x), \dots, f_k(x)\} \\ \text{rajoituksin} & x \in S \end{array}$$

missä

$x \in R^n$ on n :stä muuttujasta x_i koostuva muuttujavektori

f_i on kohdefunktio ($i = 1, \dots, k$)

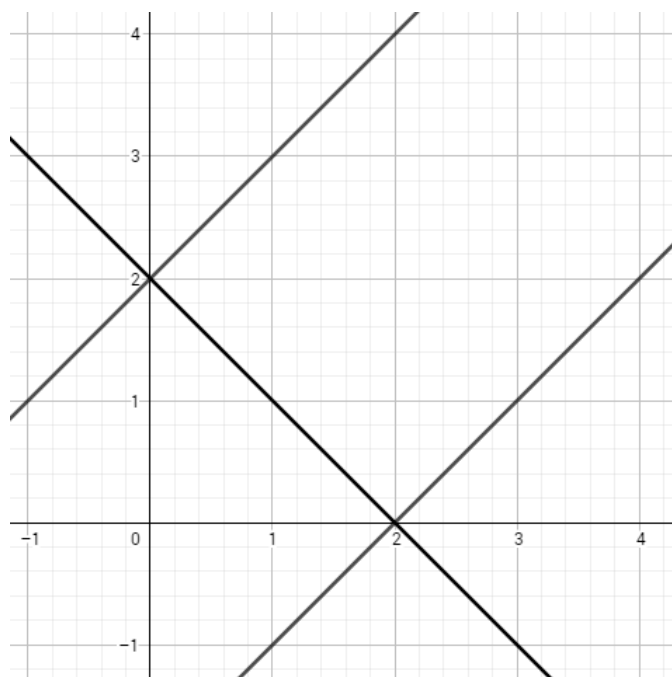
k on kohdefunktioiden lukumäärä (monitavoiteoptimoinnissa näitä on aina enemmän kuin yksi, $k \geq 2$)

S on sallittu alue.

Rajoitteet voidaan esittää esimerkiksi seuraavasti:

$$S = \{x \in R^n \mid g(x) \leq 0\},$$

missä $g(x) = (g_1(x), \dots, g_m(x))^T$ [5].



Kuva 1: Esimerkin 1 sallittu alue.

Esimerkki 1

$$\begin{array}{ll}
 \text{minimoi} & f_1 = x \\
 & f_2 = y \\
 \text{rajoituksin} & -x - y + 2 \leq 0 \\
 & x - y - 2 \leq 0 \\
 & y - x - 2 \leq 0, \\
 & x \geq 0, \quad y \geq 0
 \end{array}$$

Kuvasta 1. nähdään, että tavoite f_1 saavuttaa minimiarvonsa pisteessä $(0, 2)$. Tavoite f_2 taas saavuttaa miniminsä pisteessä $(2, 0)$.

Tällöin ongelmalle ei ole olemassa ratkaisua, jossa kumpikin kohdefunktio saavuttaisi minimiarvonsa. Tällaisessa tilanteessa voidaan hyödyntää esimerkiksi päätöksentekijän mieltymyksiä ja arvojärjestyksiä. [4]

2.3 Monitavoiteoptimointiongelman ratkaisu

Monitavoiteoptimointiongelmissa parasta ratkaisua ei yleensä löydetä vain optimointimenetelmien avulla, vaan lisäksi tarvitaan myös psykologisia keinoja. Optimoinnilla saadaan ratkaistua ongelmasta osa, ja loput jää päätök-

sentekijän ratkaistavaksi. Paras ratkaisu riippuu siis paljolti päätöksentekijän omista mieltymyksistä ja arvoista. Jotkut arvostavat esimerkiksi enemmän määrää ja toiset laatua. Tällöin saman ongelman paras ratkaisu saattaa vaihdella sen mukaan, millaisia mieltymyksiä päätöksentekijällä on. Optimoimintimenetelmillä voidaan kuitenkin määritellä ratkaisujen joukko, joista päätöksentekijä voi valita itselleen parhaan. Tätä joukkoa kutsutaan Pareto-optimaalisten ratkaisujen joukoksi.

Monitavoiteoptimoinnissa kohdefunktioita voidaan pitää optimaalisina, mikäli yhtäkään niiden komponenteista ei voida parantaa ilman, että ainakin yksi muista komponenteista huononee. Kutakin kohdefunktiota minimoivaa kohdevektoria kutsutaan *ideaaliseksi kohdevektoriksi*. Ideaalisen kohdevektorin komponentit $z_i^* \in R^k$, $i = 1, \dots, k$ saadaan minimoimalla jokaista kohdefunktiota erikseen rajoitteiden mukaan. Ideaalinen kohdevektori olisi monitavoiteoptimointiongelman ratkaisu, mutta koska tavoitteiden välillä on tavallisesti ristiriitoja, eivät komponenttien optimit yleensä löydy samasta pisteestä. Tällöin näille tehtäville on siis määriteltävä muita ratkaisujen määrittely- ja valintakeinoja. Ideaalista kohdevektoria voidaan kuitenkin pitää kiintopisteenä, jota kohti tavoitellaan. Lisäksi ideaalisen kohdevektorin komponenteista saadaan Pareto-optimaalisen joukon alarajat jokaiselle kohdefunktiolle. [8] Pareto-optimaalisuus on keskeinen käsite monitavoiteoptimoinnissa [6].

2.4 Pareto-optimaalisuus

Määritelmä 2.1. Piste $x^* \in S$ on Pareto-optimaalinen, jos ei ole olemassa toista sallittua pistettä $x \in S$ siten, että $f_i(x) \leq f_i(x^*)$ kaikilla $i = 1, \dots, k$ ja $f_j(x) < f_j(x^*)$ ainakin yhdellä indeksin j arvolla. Kohdevektori $z^* = f(x^*) \in Z = f(S)$ on Pareto-optimaalinen kohdefunktioavaruudessa, jos sitä vastaava piste $x^* \in S$ on Pareto-optimaalinen muuttuja-avaruudessa.

Pareto-optimaalisten pisteiden joukkoa kutsutaan Pareto-optimaaliseksi joukoksi. Kuvassa 2. on esitetty esimerkki muuttuja-avaruudesta S ja sitä vastaavasta kohdefunktioavaruudesta Z on esitetty esimerkki kuvassa 3. Määritelmän 2.1 mukaan kohdevektori $z^* \in Z$ on Pareto-optimaalinen kohdefunktioavaruudessa, jos sitä vastaava piste $x^* \in S$ on Pareto-optimaalinen muuttuja-avaruudessa. [5]

Pareto-optimaalisessa pisteessä mikään kohdefunktio ei voi saada pienempää arvoa, kuin ideaalisen kohdevektorin komponentti. On mielekäästä puhua pienemmistä arvoista, sillä Pareto-optimaalisten joukkojen ylärajojen

arviointi on paljon haastavampaa. [5]

Pareto-optimaalisen joukon alarajat ovat mahdollisia *ideaaliselle kohdevektorille* $z^* \in R^k$.

Määritelmä 2.2. [6] Ideaalisen kohdevektorin $z^* \in R^k$ komponentit z_i^* saadaan minimoimalla jokaista kohdefunktiota rajoitteiden mukaisesti.

$$\begin{array}{ll} \text{minimoi} & f_i(x) \\ \text{rajoituksin} & x \in S, \\ \text{kaikilla} & i = 1, \dots, k \end{array}$$

Vektoria, joka saa parempia arvoja kuin z^* , kutsutaan *utopistiseksi kohdevektoriksi* z^{**} .

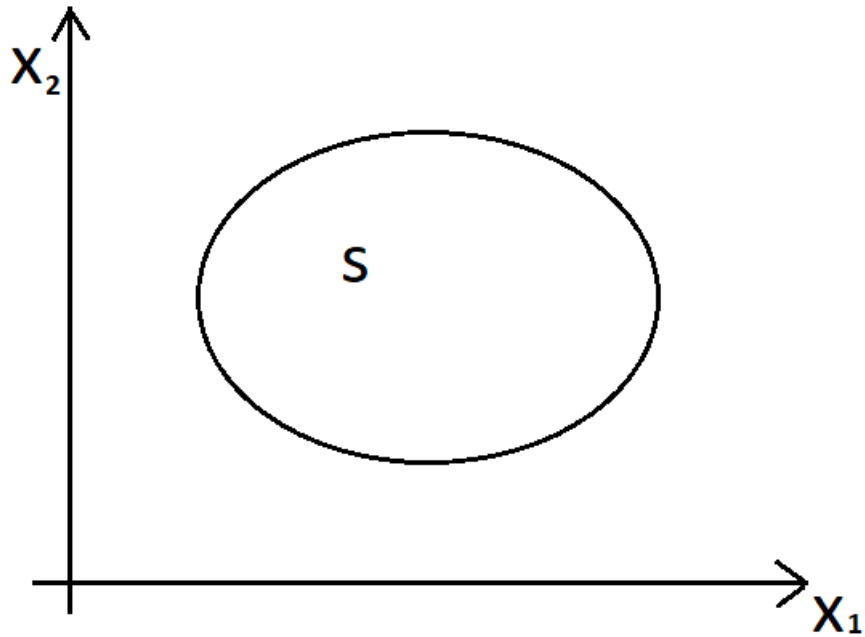
Määritelmä 2.3. [6] Utopistisen kohdevektorin $z^{**} \in R^k$ komponentit z_i^{**} muodostetaan ideaalisen kohdevektorin avulla:

$$z_i^{**} = z_i^* - \varepsilon_i$$

kaikilla $i = 1, \dots, k$. Muuttuja z_i^* on ideaalisen kohdevektorin komponentti ja $\varepsilon_i > 0$ on pieni skalaari.

Pareto-optimaalisten ratkaisujen joukon ylärajat muodostavat *nadir-kohdevektorin* z^{nad} . Todellista nadir-kohdevektoria on vaikeaa laskea, joten usein viitataan vektorin z^{nad} approksimoituun arvoon. [6] Tarkan laskumenetelmän sijaan nadir-kohdevektorin arvo voidaan arvioida käyttämällä niin sanottua *pay off -taulukkoa* [5].

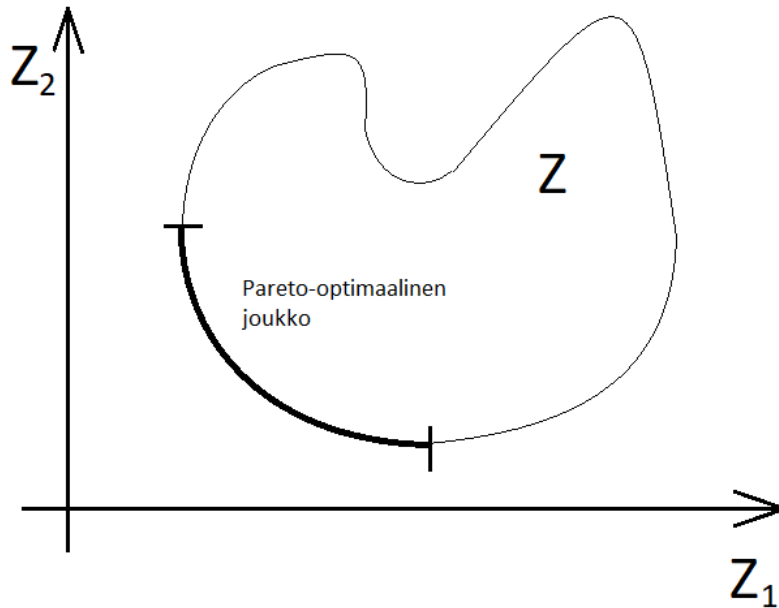
Pareto-optimaalisen joukon lisäksi on olemassa heikosti Pareto-optimaalisten pisteiden joukko. Kohdevektori on heikosti Pareto-optimaalinen, jos ei ole olemassa toista kohdevektoria, jonka kaikki komponentit saisivat parempia arvoja. Pareto-optimaalisten pisteiden joukko on heikosti Pareto-optimaalisten pisteiden osajoukko, joten heikosti Pareto-optimaalisten pisteiden joukko on yleensä liian suuri ollakseen kovin käytännöllinen. Heikosti Pareto-optimaalisia pisteitä on kuitenkin usein helpompi generoida kuin Pareto-optimaalisia pisteitä. [6]



Kuva 2: Muuttuja-avaruuden sallittu joukko S .

Määritelmä 2.4. [5] Piste $x^* \in S$ on heikosti Pareto-optimaalinen, jos ei ole olemassa toista sallittua pistettä $x \in S$ siten, että $f_i(x) < f_i(x^*)$ kaikilla $i = 1, \dots, k$. Kohdevektori $z^* \in Z$ on heikosti Pareto-optimaalinen, jos sitä vastaava sallitun alueen piste $x^* \in S$ on heikosti Pareto-optimaalinen.

Koska vektoreita ei voida järjestää, voidaan kaikkia Pareto-optimaalisen joukon ratkaisuja pitää matemaattisessa mielessä yhtä toivottuina. Tämän vuoksi tarvitaan päätöksentekijä tunnistamaan parhaat ratkaisut saadusta ratkaisujoukosta. [8]



Kuva 3: Kohdefunktioavaruus ja sallittu kohdealue Z . Tummennettu osa viivasta kuvaa Pareto-optimaalista joukkoa.

3 Monitavoiteoptimointitehtävien ratkaisumenetelmiä

Monitavoiteoptimointiongelmiä ratkaistaan yleensä yksitavoitteisen optimoinnin menetelmien hyödyntämiseen. Monitavoiteongelmat muokataan tehtävätyypistä riippuen erilaisin keinoin yksitavoiteongelmiksi. Lisäksi menetelmätyypit eroavat toisistaan siinä, millaisessa roolissa päätöksentekijä niissä on. Päätöksentekijän rooli monitavoiteongelmissa on merkittävä, sillä juuri päätöksentekijän arvioitavana on, miten hyvä matemaattinen ratkaisumenetelmä on. Pelkästään matemaatikon mielestä hyvä ratkaisumenetelmä ei suinkaan siis takaa, että menetelmä on hyvä ja käyttökelpoinen. Toki myös tilanne, jossa päätöksentekijä ja matemaatikko ovat sama henkilö, on mahdollinen.

Seuraavaksi esitellään neljä erilaista monitavoiteoptimoinnin menetelmäluokkaa, joissa päätöksentekijä osallistuu ratkaisuprosessiin erilaisin tavoin. Lisäksi esitellään kustakin menetelmästä esimerkki, jossa on käytetty jotakin menetelmäluokkaan kuuluvaa ratkaisumenetelmää.

3.1 Päätöksentekijän mielipidettä ei huomioida

Ensimmäisessä kategoriassa päätöksentekijän mielipidettä ei huomioida, eikä päätöksentekijää näin ollen tarvita. Tällaisissa tapauksissa eri tavoitteille ei siis anneta erilaisia painoarvoja, vaan jokainen kohdefunktio on yhtä tärkeä. Yritetään löytää siis ratkaisu, jossa kaikki tavoitefunktiot saavat mahdollisimman pieniä arvoja.

3.1.1 Globaalin tavoitteen menetelmä

Globaalin tavoitteen menetelmissä on ensin muodostettava jotenkin globaali tavoite, jota sitten minimoidaan. Useimmiten minimoidaan etäisyyttä referenssipisteen ja sallitun kohdealueen välillä. Kaikkia kohdefunktioita pidetään tässä menetelmässä saman arvoisina. Tehtävän ratkaisua varten on löydettävä sopiva yksitavoitteisen optimoinnin menetelmä. Seuraavassa ratkaisumenetelmässä ideaalista kohdevektoria käytetään referenssipisteenä ja mittaukseen käytetään L_p -metriikkaa:

$$\begin{array}{ll} \text{minimoi} & (\sum_{i=1}^k |f_i(x) - z_i^*|^p)^{\frac{1}{p}} \\ \text{rajoituksin} & x \in S, \end{array} \quad (1)$$

missä $1 \leq p \leq \infty$.

Erilaisilla p :n valinnoilla voidaan tuottaa erilaisia ratkaisuja. Kun $p = \infty$, saadaan *Tsebysevin metriikka*, jonka ratkaisut ovat heikosti Pareto-optimaalisia. [5]

Lause 3.1. [6] Tehtävän (1) ratkaisut ovat Pareto-optimaalisia, kun $1 \leq p < \infty$.

Todistus. Olkoon $x^* \in S$ ongelman (1) ratkaisu, kun $1 \leq p < \infty$.

Oletetaan, että x^* ei ole Pareto-optimaalinen. Tällöin on olemassa sellainen piste $x \in S$, jolla $f_i(x) \leq f_i(x^*)$ kaikilla $i = 1, \dots, k$ ja $f_j(x) < f_j(x^*)$ ainakin yhdellä indeksin j arvolla.

Tällöin $(f_i(x) - z_i^*)^p \leq (f_i(x^*) - z_i^*)^p$ kaikilla i ja $(f_j(x) - z_j^*)^p < (f_j(x^*) - z_j^*)^p$.

Nyt saadaan

$$\sum_{i=1}^k (f_i(x) - z_i^*)^p < \sum_{i=1}^k (f_i(x^*) - z_i^*)^p$$

Kun epäyhtälön molemmat puolet kohotetaan potenssiin $1/p$, saadaan epäyhtälö, jonka mukaan on olemassa ratkaisua x^* pienempi ratkaisu x . Saadaan ristiriita oletuksen kanssa, jonka mukaan x^* on ongelman (1) ratkaisu. Voidaan siis todeta, että ratkaisu x^* on Pareto-optimaalinen. \square

3.2 Päätöksentekijän mielipiteet huomioidaan etukäteen

Tässä kategoriassa päätöksentekijä miettii jo etukäteen eri kohdefunktioiden painoarvot, joiden avulla voidaan määrittää kohdefunktioiden tärkeysjärjestys. Sitten matemaatikko etsii annettujen tietojen ja arvojen pohjalta parhaan mahdollisen ratkaisun. Tässä menetelmässä tilanne on paras, kun päätöksentekijä on tietoinen paremmuuksista ja niiden vaikutuksista lopputulokseen. Tällöin päätöksentekijä ei joudu tekemään valintoja ikäänkuin sokkona.

3.2.1 Painokerroinmenetelmä

Painokerroinmenetelmässä kaikille kohdefunktioiden f_i määrätään painokerroin $w_i \geq 0$ ($i = 1, \dots, k$) ja ratkaistaan seuraava tehtävä:

$$\begin{array}{ll} \text{minimoi} & \sum_{i=1}^k w_i f_i(x) \\ \text{rajoituksin} & x \in S, \end{array} \quad (2)$$

missä $w_i \geq 0$ kaikilla $i = 1, \dots, k$ ja $\sum_{i=1}^k w_i = 1$.

Painokerrointen avulla saadaan määriteltyä kohdefunktioiden tärkeysjärjestys. [5]

Lause 3.2. [6] Painokerrointehtävän (3) ratkaisu on Pareto-optimaalinen, jos $w_i > 0$ kaikilla $i = 1, \dots, k$.

Todistus. Olkoon $x^* \in S$ vain positiivisia painokertoimia sisältävän painokerrointehtävän ratkaisu. Oletetaan, että ratkaisu ei ole Pareto-optimaalinen. Tällöin on olemassa ratkaisu $x \in S$ jolla $f_i(x) \leq f_i(x^*)$ kaikilla $i = 1, \dots, k$, ja epäyhtäsuuruus on aito ainakin yhdellä indeksin j arvolla.

Koska $w_i > 0$ kaikilla $i = 1, \dots, k$, saadaan että $\sum_{i=1}^k w_i f_i(x) < \sum_{i=1}^k w_i f_i(x^*)$. Tällöin saadaan epäyhtälö, jonka mukaan on olemassa ratkaisua x^* pienempi ratkaisu x . Saadaan ristiriita oletuksen kanssa, jonka mukaan x^* on tehtävän (3) ratkaisu. Ratkaisun x^* on siis oltava Pareto-optimaalinen.

□

Mikäli painokertoimia varioidaan, saadaan tuotettua vielä erilaisia Pareto-optimaalisia ratkaisuja. Painokertoimien suuruuksien lisäksi painokerrointehtävien ratkaisussa tulee muistaa kohdefunktioiden skaalaus, jotta painokertoimet toimivat halutusti. [5]

3.3 Päätöksentekijän mielipiteet otetaan huomioon jälkikäteen

Tätä kategoriaa voidaan kutsua myös Pareto-optimaalisten pisteiden määräämismenetelmäksi. Matemaatikko siis etsii Pareto-optimaalisten pisteiden joukon, josta päätöksentekijä saa valita mielestään parhaan. Mikäli Pareto-optimaalisia pisteitä on ääretön määrä, on parhaan pisteen valinta vaikeaa. Tällöin on etsittävä sopiva edustajisto pisteistä. Pareto-optimaalisten pisteiden etsiminen saattaa olla myös vaikeaa ja raskasta. Pareto-optimaalisten pisteiden määräämismenetelmien avulla tehtäviä voidaan kuitenkin palauttaa yksitavoitetehtäväksi, mihin monet monitavoitemenetelmät perustuvatkin.

3.3.1 Rajoiteyhtälömenetelmä

Rajoiteyhtälömenetelmässä valitaan yksi kohdefunktio f_1 optimoitavaksi. Sitten asetetaan muille kohdefunktiolle ylärajat ε_i ($i = 2, \dots, k$), jotka saadaan määriteltä esimerkiksi lausekkeen $\varepsilon_i = z_i^* + \delta_i$ avulla. Lausekkeessa z_i^* on i :nnen kohdefunktion ideaalinen arvo ja δ_i on muunneltavissa oleva positiivinen luku. Näin saadaan palautettua tehtävä yksitavoitteiseksi optimointitehtäväksi

$$\begin{array}{ll}
\text{minimoi} & f_1(x) \\
\text{rajoituksin} & f_i(x) \leq \varepsilon_i, \quad i = 2, \dots, k, \\
& x \in S.
\end{array} \tag{3}$$

Yksitavoitteinen optimointitehtävä ratkaistaan sallitun alueen S osajoukoissa. Rajoitteiden monimutkaiset funktiot voivat kuitenkin tehdä kuva-alueen hahmottamisesta haastavaa. Saatetaan joutua esimerkiksi tilanteeseen, jossa alkuperäisen alueen ja uusien rajoitteiden muodostaman alueen leikkaus on tyhjä. [5]

Lause 3.3. [5] Jos piste $x^* \in S$ on tehtävän (3) yksikäsitteinen ratkaisu, kun $\varepsilon_i = f_i(x^*)$, niin x^* on Pareto-optimaalinen.

Todistus. Olkoon $x^* \in S$ tehtävän (3) yksikäsitteinen ratkaisu minimoitaessa funktiota f_1 . Oletetaan, että ratkaisu ei ole Pareto-optimaalinen. Tällöin on olemassa toinen ratkaisu $x \in S$, jolla $f(x) \leq f_i(x^*)$ kaikilla i .

Pisteen x^* yksikäsitteisyyden mukaan kaikilla $\bar{x} \in S$, joilla $f_i(\bar{x}) \leq f_i(x^*)$ kun $i \neq 1$, on $f_1(x^*) < f_1(\bar{x})$.

Saadaan edellisten epäyhtälöiden kanssa aikaan ristiriita, joten voidaan todeta, että x^* on Pareto-optimalinen. \square

Pareto-optimaalisten pisteiden joukko saadaan muuttamalla tehtävän (3) ylärajoja ε_i ($i = 2, \dots, k$) tai minimoimalla eri kohdefunktiota.

3.4 Interaktiiviset menetelmät: päätöksentekijän ja ratkaisumenetelmän välistä vuorovaikutusta

Seuraava kategoria poikkeaa edellisistä menetelmistä siten, että tässä päätöksentekijä saa antaa mielipiteitä ja miettiä paremmuussuhteita sitä mukaa, kun matemaatikko antaa erilaisia vaihtoehtoja ja etenee tehtävän ratkaisussa. Päätöksentekijän ei siis tarvitse etukäteen tietää mahdollisia painotuksia, tai jälkikäteen valita ratkaisua isosta määrästä Pareto-optimaalisia pisteitä. Tällöin esimerkiksi paperikoneen toiminnan optimoinnissa päätöksentekijä saa olla mukana prosessissa, ja pystyy antamaan uutta tietoa tai mieltymyksiään päästessään paremmin perille tehtävän luonteesta ja mahdollisista ratkaisuista.

3.4.1 ISWT (Interactive Surrogate Worth Trade-Off)

ISWT -menetelmän perusajatuksena on maksimoida tuntematonta hyötyfunktioita. Tehtävä ratkaistaan ensin rajoiteyhtälömenetelmällä, jonka jälkeen saadusta ratkaisusta voidaan lähteä eteenpäin: selvitetään millaisia mieltymyksiä päätöksentekijällä on eri kohdefunktioiden arvojen muutoksista tässä pisteessä.

Jokaisen kohdefunktion kohdalla annetaan vaihtoehto, jolla kyseinen kohdefunktion arvo vähenee, mutta samaan aikaan jonkin toisen kohdefunktion arvo kasvaa. Päätöksentekijä saa omien arvojensa perusteella sanoa, kannattaako vai eikö kannata tämän kyseisen kohdefunktion arvon vähentämistä. Saatujen vastausten perusteella voidaan sitten muuttaa rajoiteyhtälötehtävää, ja saadaan ratkaistua uusi piste, jossa päätöksentekijä saa vastata samoihin kysymyksiin kuin edellä.

Lopulta tulisi päästä ratkaisupisteeseen, jossa päätöksentekijä ei halua muuttaa minkään kohdefunktion arvoa. Menetelmä tuottaa Pareto-optimaalisia ratkaisuja, mutta niiden löytäminen saattaa tällä menetelmällä kestää kauan, mikäli päätöksentekijä ei osaa antaa aivan johdonmukaisia vastauksia toistuvilla kierroksilla. Päätöksentekijä saa kuitenkin olla tiiviisti mukana ratkaisun löytämisessä ja saatu ratkaisu on lopulta varmasti päätöksentekijää miellyttävä. [5]

Seuraavaksi esitellään tarkemmin toista interaktiivista ratkaisumenetelmää; NIMBUS-menetelmää.

4 NIMBUS

NIMBUS-menetelmän esittely perustuu kolmeen eri lähteeseen: Kaisa Miettisen teoksiin *Epälineaarinen optimointi* [5] ja *Nonlinear multiobjective optimization* [6], sekä K. Miettisen ja M. M. Mäkelän artikkeliin *Synchronous approach in interactive multiobjective optimization* [8].

4.1 NIMBUS-menetelmä yleisesti

NIMBUS on monitavoiteoptimointimenetelmä, joka pyrkii tekemään ongelmanratkaisun päätöksentekijälle mahdollisimman miellyttäväksi. Menetelmän tavoitteena on helpottaa matemaatikon ja päätöksentekijän yhteistyötä ja tehdä ongelmanratkaisusta joustavampaa. Päätöksentekijä oppii ratkaisuprosessin aikana tuntemaan ongelman ja sen mahdolliset ratkaisumahdollisuudet paremmin, ja voi halutessaan muuttaa aiempia päätöksiään. Päätöksentekijä saa näin olla alusta asti mukana päätöksenteossa. Helppokäyttöisyyden lisäksi menetelmä painottaa laskennallista tehokkuutta.

4.1.1 NIMBUS-menetelmän keskeisiä käsitteitä

Ensimmäiseksi päätöksentekijän tulee luokitella viiteen eri luokkaan kohdefunktiot f_i , joiden arvot on laskettu Pareto-optimaalisessa pisteessä x^h .

Funktioluokat:

$I^<$ funktiot, joiden arvoja halutaan pienentää.

I^{\leq} funktiot, joiden arvoja halutaan pienentää tavoitetasolle $\hat{z}_i < f_i(x^h)$.

$I^=$ funktiot, joiden arvoja ei haluta tällä hetkellä muuttaa.

$I^>$ funktiot, joiden arvot saavat kasvaa ylärajaan $\varepsilon_i > f_i(x^h)$ asti.

I° funktiot, joiden arvot saavat vaihdella vapaasti.

Kaikkien funktioluokkajoukkojen yhdisteen on luokittelussa oltava joukko $\{1, \dots, k\}$ eli

$$I^< \cup I^{\leq} \cup I^= \cup I^> \cup I^\circ = \{1, \dots, k\}.$$

Luokittelussa joukkojen $I^< \cup I^{\leq}$ ja $I^> \cup I^\circ$ on lisäksi oltava epätyhjiä. Tämä seuraa Pareto-optimaalisuudesta, sillä jotta jonkin kohdefunktion arvoa voitaisiin kasvattaa, on jonkin kohdefunktion arvoa voitava pienentää.

Päätöksentekijän on luokittelua varten määrättävä myös seuraavat tiedot:

- luokan $i \in I^{\leq}$ funktioille tavoitetasot $\hat{z}_i < f_i(x^h)$

- luokan $i \in I^>$ funktioille ylärajat $\varepsilon_i > f_i(x^h)$.

Luokan $I^<$ funktioita minimoidaan niin paljon, kuin mahdollista.

Referenssipiste

Referenssipiste muodostuu tavoitetasoista \bar{z}_i , jotka päätöksentekijä haluaa kohdefunktioiden arvojen saavuttavan. Useimmiten ideaalisen kohdevektorin arvo on laskettu ja esitetty päätöksentekijälle jo ennen kohdefunktioiden luokittelua, jolloin päätöksentekijä saa tietoa tehtävän mahdollisuuksista ja rajoitteista. Tällöin voidaan olettaa, että päätöksentekijä ei määrittele toiveita ideaalisia arvoja paremmaksi, eli oletetaan, että $\bar{z}_i \geq z_i^*$ kaikilla $i = 1, \dots, k$.

Kun päätöksentekijä on antanut luokittelutiedot, voidaan niiden avulla muodostaa referenssipiste $\bar{z} \in R^k$ (mikäli kohdevektoreiden vaihteluvälit tunnetaan Pareto-optimaalisessa joukossa) seuraavasti: [8]

$$\begin{aligned}\bar{z}_i &= z_i^* \text{ kaikilla } i \in I^<, \\ \bar{z}_i &= \hat{z}_i \text{ kaikilla } i \in I^\leq, \\ \bar{z}_i &= f_i(x^h) \text{ kaikilla } i \in I^=, \\ \bar{z}_i &= \varepsilon_i \text{ kaikilla } i \in I^>, \\ \bar{z}_i &= z_i^{nad} \text{ kaikilla } i \in I^\circ.\end{aligned}$$

4.2 NIMBUS-menetelmän aputehtävät

Edellisten päätösten ja niistä saatujen tietojen perusteella voidaan muodostaa uusi tehtävä: [8]

$$\begin{aligned}\text{minimoi} \quad & \max_{i \in I^<, j \in I^\leq} \left[\frac{f_i(x) - z_i^*}{z_i^{nad} - z_i^{**}}, \frac{f_j(x) - \hat{z}_j}{z_j^{nad} - z_j^{**}} \right] + \rho \sum_{i=1}^k \frac{f_i(x)}{z_i^{nad} - z_i^{**}} \\ \text{rajoituksin} \quad & f_i(x) \leq f_i(x^c) \text{ kaikilla } i \in I^< \cup I^\leq \cup I^= \\ & f_i(x) \leq \varepsilon_i, \text{ kaikilla } i \in I^> \\ & x \in S,\end{aligned} \tag{4}$$

missä

$\rho > 0$ on täydennyskerroin (pieni reaaliluku).

z_i^* on ideaalisen kohdevektorin komponentti.

z_i^{**} on utopiavektorin komponentti.

z_i^{nad} on nadir-kohdevektorin komponentti.

$\frac{1}{z_i^{nad} - z_i^{**}}$ ovat painokertoimia, jotka skaalaaavat kohdefunktioiden arvoja ja lisäävät siten laskennallista tehokkuutta.

Aputehtävistä (4), (5), (6) ja (7) saadut ratkaisut on taattu olevan Pareto-optimaalisia täydennyskerrointermin avulla [8]. Pareto-optimaalisia pisteitä lähestytään yleensä kartion avulla, jonka sivut ovat koordinaattiakselien suuntaisia. Täydennyskerrointermin avulla tästä kartiosta saadaan muodostettua tylppä, jolloin saadut ratkaisut ovat varmasti Pareto-optimaalisia.

Pareto-optimaalisen ja heikosti Pareto-optimaalisen pisteen erottaa se, että heikosti Pareto-optimaalisen pisteen tapauksessa saattaa olla olemassa piste, jossa joidenkin kohdefunktioiden arvot parantuvat, mutta joidenkin kohdefunktioiden arvot pysyvät samoina. Pareto-optimaalisen pisteen tapauksessa taas kaikkien kohdefunktioiden arvojen tulee parantua. Heikosti Pareto-optimaalisia pisteitä saattaa esiintyä esimerkiksi koordinaattiakselien suuntaisilla viivoilla. Täydennyskerrointermin avulla saadaan siis tilanne, jossa kartio on tylppä, ja tällöin muodostuu vain Pareto-optimaalisia ratkaisuja.

Lause 4.1. [8] Tehtävän (4) ratkaisut ovat Pareto-optimaalisia.

Todistus. Olkoon tehtävän (4) ratkaisu $x^* \in S$. Oletetaan, että ratkaisu ei ole Pareto-optimaalinen. Tällöin on olemassa $x' \in S$ jolla $f_i(x') \leq f_i(x^*)$ kaikilla $i = 1, \dots, k$. Epäyhtäsuuruus on aito ainakin yhdellä indeksin j arvolla. Tästä seuraa, että x' on tehtävän (4) sallittu ratkaisu.

Kun $z_i^{nad} > z_i^{**}$, saadaan

$$\max_{i \in I^<, j \in I^{\leq}} \left[\frac{f_i(x') - z_i^*}{z_i^{nad} - z_i^{**}}, \frac{f_j(x') - \hat{z}_j}{z_j^{nad} - z_j^{**}} \right] \leq \max_{i \in I^<, j \in I^{\leq}} \left[\frac{f_i(x^*) - z_i^*}{z_i^{nad} - z_i^{**}}, \frac{f_j(x^*) - \hat{z}_j}{z_j^{nad} - z_j^{**}} \right]$$

ja

$$\rho \sum_{i=1}^k \frac{f_i(x')}{z_i^{nad} - z_i^{**}} < \rho \sum_{i=1}^k \frac{f_i(x^*)}{z_i^{nad} - z_i^{**}}$$

Täten x^* ei voi olla tehtävän (4) ratkaisu. Tästä ristiriidasta seuraa, että ratkaisu x^* on Pareto-optimaalinen. \square

Yleensä matemaatikko valitsee päätöksentekijän puolesta menetelmän, jolla monitavoitetehtävä muutetaan yksitavoitetehtäväksi. NIMBUS-

menetelmässä kuitenkin käytetään useampia aputehtäviä, jolloin päätöksentekijä saa valittavakseen useampia eri ratkaisuja. [5]

Seuraavaksi esitellään nämä yksitavoitteiset aputehtävät, jotka on esitelty K. Miittisen ja M. M. Mäkelän artikkelissa [8].

$$\begin{array}{ll} \text{minimoi} & \max_{i=1,\dots,k} \left[\frac{f_i(x) - z_i^{**}}{\bar{z}_i - z_i^{**}} \right] + \rho \sum_{i=1}^k \frac{f_i(x)}{\bar{z}_i - z_i^{**}} \\ \text{rajoittein} & x \in S, \end{array} \quad (5)$$

missä tavoitetasojen \bar{z}_i on oltava utopiavektoreita z_i^{**} aidosti suurempia.

$$\begin{array}{ll} \text{minimoi} & \max_{i=1,\dots,k} \left[\frac{f_i(x) - \bar{z}_i}{z_i^{nad} - z_i^{**}} \right] + \rho \sum_{i=1}^k \frac{f_i(x)}{z_i^{nad} - z_i^{**}} \\ \text{rajoittein} & x \in S \end{array} \quad (6)$$

$$\begin{array}{ll} \text{minimoi} & \max_{i \notin I^\circ} \left[\frac{f_i(x) - z_i^{nad}}{z_i^{nad} - \bar{z}_i} \right] + \rho \sum_{i=1}^k \frac{f_i(x)}{z_i^{nad} - \bar{z}_i} \\ \text{rajoittein} & x \in S. \end{array} \quad (7)$$

Yksitavoitteiset aputehtävät (5), (6) ja (7) perustuvat samoille päätöksentekijän antamille luokittelutiedoille kuin tehtävä (4). Samoista lähtötiedoista muodostetut aputehtävät tuottavat kuitenkin erilaisia Pareto-optimaalisia ratkaisuja, jolloin päätöksentekijä saa kattavamman valikoiman mahdollisia ratkaisuja. [5] Aputehtävät eroavat toisistaan siinä, onko niissä hyödynnetty ideaalisen kohdevektorin, utopistisen kohdevektorin, nadir-kohdevektorin vai referenssipisteen komponentteja. Tehtävää (5) hyödynnetään STOM-menetelmässä, tehtävää (6) referenssipiste-menetelmässä ja tehtävää (7) GUESS-menetelmässä [8].

4.3 NIMBUS-algoritmi

Seuraavaksi käydään vaihe vaiheelta läpi, miten NIMBUS-algoritmi toimii, ja millaisia tehtäviä matemaatikolla ja päätöksentekijällä optimointitehtävän ratkaisussa on. Kappale perustuu K. Miittisen teokseen [5] ja K. Miittisen ja M. M. Mäkelän artikkeliin [8].

1. Ratkaisuprosessin lähtöpiste voi tulla päätöksentekijältä, tai se voi olla jokin kompromissi kohdefunktioiden välillä. Generoidaan Pareto-optimaalinen lähtöpiste $x^1 \in S^n$, jota kutsutaan nykyratkaisuksi, minimoimalla jokaista kohdefunktiota rajoitteiden mukaisesti. Tällöin saadaan ideaalisen kohdevektorin z^* komponentit, jotka muodostavat Pareto-optimaalisen joukon alarajat.

2. Päätöksentekijän tulee luokitella kohdefunktiot nykyratkaisussa luokkiin $I^<$, I^{\leq} , $I^=$, $I^>$ ja I° , sekä määritellä tavoitetasot ja ylärajat.

3. Päätöksentekijän tulee kertoa, montako ratkaisua haluaa maksimissaan generoitavan väliltä yhdestä neljään. Matemaatikko tuottaa sitten yhtä monta ratkaisua ratkaisemalla edellä esiteltyjä yksitavoitteisia aputehtäviä (4),(5),(6),(7).

4. Matemaatikko esittelee saamansa erilaiset ratkaisut päätöksentekijälle.

5. Päätöksentekijä voi halutessaan tallettaa saadun/ saatuja ratkaisuja joukkoon A . Joukko A sisältää vain saatuja ratkaisuja, ja on siten aluksi tyhjä joukko. Päätöksentekijä voi jatkossa tarkastella uudelleen joukkoon lisättyjä ratkaisuja.

6. Päätöksentekijä voi valita, haluaako nähdä välivaihtoehtoja kahden aiemmin saadun ratkaisun välillä. Mikäli haluaa, tulee hänen valita kaksi ratkaisua uusista tai joukkoon A talletetuista ratkaisuksista, sekä päättää, montako ratkaisua haluaa generoitavan. Mikäli päätöksentekijä ei halua nähdä välivaihtoehtoja, voidaan mennä suoraan kohtaan 8.

7. Matemaatikko generoi väliratkaisuja halutun määrän ja projisoi ne Pareto-optimaaliseksi. Kahden pisteen x^h ja \hat{x}^h väliratkaisut saadaan muodostamalla hakusuunta $d^h = \hat{x}^h - x^h$ ja määräämällä askelpituus. Kohdevektoreita $f(x^h + td^h)$ lasketaan t :n arvoilla kun $0 \leq t \leq 1$. Ratkaisuksista saadaan Pareto-optimaalisia referenssipistepohjaisen aputehtävän (5), (6) tai (7) avulla. Sitten mennään kohtaan 4, eli esitellään saadut ratkaisut jälleen päätöksentekijälle.

8. Päätöksentekijän tulee valita mielestään paras ratkaisu uusien ratkaisujen, väliratkaisujen ja talletettujen ratkaisujen joukosta. Mikäli päätöksentekijä haluaa jatkaa, tehdään valitusta ratkaisusta nykyratkaisu ja mennään uudelleen kohtaan 2. Mikäli päätöksentekijä on tyytyväinen saatuun ratkaisuun, voidaan algoritmi lopettaa tähän.

Saatujen ratkaisujen jälkeen voidaan aina palata kohtaan 2, jolloin päätöksentekijä saa luokitella uudestaan kohdefunktiot nykyratkaisussa, sekä antaa tarvittavat ylärajat ja tavoitetasot. Päätöksentekijä voi siis aina palata takaisin päin, tai muuttaa valintojaan. Hänen on tällöin mahdollista

oppia ongelmasta lisää, sekä muuttaa käsityksiään saadessaan lisää tietoa. Tämä on juuri NIMBUS-algoritmin vahvuus. Pareto-optimaalisuudesta johtuen päätöksentekijän on tosin oltava valmis heikentämään jonkin kohdefunktion arvoa, mikäli hän haluaa parantaa jonkin toisen kohdefunktion arvoa. Algoritmin pyörittäminen ei siis sinänsä anna parempia arvoja, jos päätöksentekijä ei ole valmis heikentämään minkään funktion arvoa.

4.4 Yksinkertaisen optimointiongelman ratkaisu

Seuraavaksi esitellään yksinkertaisen optimointiongelman ratkaisu NIMBUS-menetelmän avulla. NIMBUS-menetelmää pystyy käyttämään nettiselaimella ilman, että tietokoneelle tarvitsee ladata erillistä ohjelmaa ongelman ratkaisua varten [8]. Tehtävä (8) on artikkelista [7] ja on nimeltään Chankong-Haimes. WWW-NIMBUS -menetelmän käyttöä on esitelty artikkelissa [8].

Esimerkki 4.2. Chankong-Haimes

$$\begin{aligned}
 \text{minimoi} \quad & f_1(x) = (x_1 - 1)^2 + (x_2 - 1)^2 \\
 & f_2(x) = (x_1 - 2)^2 + (x_2 - 3)^2 \\
 & f_3(x) = (x_1 - 4)^2 + (x_2 - 2)^2 \\
 \text{rajoittein} \quad & g_1(x) = x_1 + 2x_2 \leq 10 \\
 & 0 \leq x_1 \leq 10 \\
 & 0 \leq x_2 \leq 4
 \end{aligned} \tag{8}$$

Aloitetaan tehtävän ratkaisu syöttämällä kohdefunktiot, muuttujat ja rajoitteet WWW-NIMBUS -menetelmään [9].

Valitaan sitten aloituspisteet: $x_1 = 5$ ja $x_2 = 2$ (Kuva 4.).

NIMBUS-menetelmä laskee lähtötietojen avulla lähtöpisteen, jonka avulla pystytään suorittamaan luokittelu. Ensimmäisessä luokittelussa (Kuva 5.) kohdefunktion f_1 arvo pyritään saamaan tasolle 2.5 (Kuva 6.) ja kohdefunktion f_2 arvo taas ei saa nousta yli ylärajan 5.0. Kohdefunktion f_3 arvoa pyritään minimoimaan.

Aloitusratkaisun lisäksi saatiin kolme uutta ratkaisua, joista valittiin se, jossa kohdefunktioiden arvot olivat lähimpänä toisiaan: $f_1 = 2.679605$, $f_2 = 2.497952$ ja $f_3 = 2.32663$ (Kuva 7).

Valitut arvot luokiteltiin nyt toisen kerran (Kuva 8.). Kohdefunktion f_2 arvo saa edelleen nousta ylärajaan 5.0 asti ja funktion f_1 pyritään edelleen saamaan tasolle 2.5. Funktion f_3 arvolle asetetaan tavoitetaso 2.0 (Kuva 9.).

Input Problem

Fill in all the fields, please. Note that the function and the variable names are used when displaying results, and the variable names x1, x2, x3, etc. must be used when defining the problem.

Objective functions to be optimized

| name | expression | Lowest Value | Highest Value |
|----------|--|----------------------|----------------------|
| Min ▾ f1 | <input type="text" value="(x1-1)^2+(x2-1)^2"/> | <input type="text"/> | <input type="text"/> |
| Min ▾ f2 | <input type="text" value="(x1-2)^2+(x2-3)^2"/> | <input type="text"/> | <input type="text"/> |
| Min ▾ f3 | <input type="text" value="(x1-4)^2+(x2-2)^2"/> | <input type="text"/> | <input type="text"/> |

Use [Standard functions available](#)

Variables

Use [autofill](#)

| | lower bound | <= starting point | <= upper bound | Integer values |
|----|--------------------------------|--------------------------------|---------------------------------|--------------------------|
| x1 | <input type="text" value="0"/> | <input type="text" value="5"/> | <input type="text" value="10"/> | <input type="checkbox"/> |
| x2 | <input type="text" value="0"/> | <input type="text" value="2"/> | <input type="text" value="4"/> | <input type="checkbox"/> |

Linear Constraints

| | Ax | b |
|-------|---|--------------------------------------|
| No1 : | <input type="text" value="1"/> *x1 + <input type="text" value="2"/> *x2 | <input type="text" value="10"/> <= ▾ |

Kuva 4: Lähtötietojen syöttäminen WWW-NIMBUS -menetelmään.

Classify Functions

Specify the direction of change from the current criterion vector by selecting an appropriate class for classification instead, select the appropriate operation below.

| Function | Current solution | Class: | < | <= | = | >= | > |
|----------|------------------|--------|----------------------------------|----------------------------------|-----------------------|-----------------------|----------------------------------|
| Min f1 | 2.679605 | | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Min f2 | 1.339437 | | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> |
| Min f3 | 2.679605 | | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

| Function | Lowest Value | Highest Value |
|----------|--------------|---------------|
| Min f1 | 0.0 | 10.0 |
| Min f2 | 0.0 | 5.0 |
| Min f3 | 0.0 | 10.0 |

Next optimization: ▾

Maximum number of new solutions to be generated: ▾

Select operation

- Another problem
- Specify classification parameters if necessary (continue)
- Use graphical classification instead
- Go to solution database
- Correct Highest or Lowest values
- Show the whole problem
- Modify this problem
- Stop

Kuva 5: Ensimmäinen luokittelu.

Enter Classification Parameters

Aspiration functions

Enter desired level(s)
Min f1 0.0 <= <= 2.679605

Select operation

- Another problem Calculate a new solution (continue)
 Correct Highest or Lowest values Show the whole problem

Kuva 6: Määritetään funktiolle f_1 tavoitetaso 2.5.

Analyse Results

Select one alternative for continuation (classification), generate intermediate solutions alternatives visually or stop.

- Alternative1** : f1 : 2.679605
f2 : 1.339437
f3 : 2.679605
- Alternative2** : f1 : 2.679605
f2 : 2.497952
f3 : 2.32663
- Alternative3** : f1 : 9.664362
f2 : 4.83357
f3 : 2.864599E-3
- Alternative4** : f1 : 3.90626
f2 : 2.646301
f3 : 1.40626

Kuva 7: Ensimmäisen luokittelun jälkeen saadut kolme uutta ratkaisua.

Classify Functions

Specify the direction of change from the current criterion vector by selecting an appropriate class for classification instead, select the appropriate operation below.

| Function | Current solution | Class: | < | <= | = | >= | > |
|----------|------------------|--------|-----------------------|----------------------------------|-----------------------|----------------------------------|-----------------------|
| Min f1 | 2.679605 | | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Min f2 | 2.497952 | | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> |
| Min f3 | 2.32663 | | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

| Function | Lowest Value | Highest Value |
|----------|--------------|---------------|
| Min f1 | 0.0 | 10.0 |
| Min f2 | 0.0 | 5.0 |
| Min f3 | 0.0 | 10.0 |

Next optimization:

Maximum number of new solutions to be generated:

Select operation

- Another problem
- Specify classification parameters if necessary (continue)
- Use graphical classification instead
- Go to solution database
- Correct Highest or Lowest values
- Show the whole problem
- Modify this problem
- Stop

Kuva 8: Toinen luokittelu.

Enter Classification Parameters

Aspiration functions

Enter desired level(s)

| | | | | | |
|--------|-----|----|----------------------------------|----|----------|
| Min f1 | 0.0 | <= | <input type="text" value="2.5"/> | <= | 2.679605 |
| Min f3 | 0.0 | <= | <input type="text" value="2.0"/> | <= | 2.32663 |

Select operation

- Another problem
- Calculate a new solution (continue)
- Correct Highest or Lowest values
- Show the whole problem

Kuva 9: Tavoitetasot funktioille f_1 ja f_3 .

Analyse Results

Select one alternative for continuation (classification), generate intermediate solutions¹ alternatives visually or stop.

- Alternative1** : f_1 : 2.679605
 f_2 : 2.497952
 f_3 : 2.32663
- Alternative2** : f_1 : 2.786392
 f_2 : 2.505533
 f_3 : 2.229136
- Alternative3** : f_1 : 2.75626
 f_2 : 2.496301
 f_3 : 2.25626

Number of new alternatives : <= 15

Select operation

- Another problem
- Generate new alternatives
- Go to solution database
- Modify this problem
- Select one alternative (continue)
- Visualize the alternatives
- Show the whole problem
- Stop

Kuva 10: Toisen luokittelun jälkeen saadut uudet ratkaisut.

Toisen luokittelun ja tavoitetasojen määräämisen jälkeen saadaan generoitua vain kaksi uutta ratkaisua (Kuva 10.). Yhdessäkin ratkaisussa funktio f_1 ei saavuta tavoitetasoa 2.5, joten lopulliseksi ratkaisuksi valittiin ratkaisu, jossa funktion f_3 arvo saatiin lähemmäs tavoitetasoa 2.0: $f_1 = 2.786392$, $f_2 = 2.505533$ ja $f_3 = 2.229136$. Muuttujien arvoiksi saadaan tällöin $x_1 = 2.583365$ ja $x_2 = 1.528532$ (Kuva 11.).

4.4.1 Optimointiongelman ratkaisu erilaisilla luokittelutiedoilla

Ratkaistaan seuraavaksi sama ongelma kuin edellä (8), mutta erilaisilla luokittelutiedoilla. Tällöin nähdään, miten monitavoiteoptimointiongelman ratkaisut voivat muuttua, kun päätöksentekijä vaihtuu. Tehtävän ratkaisu aloitetaan jälleen syöttämällä kohdefunktiot, muuttujat ja rajoitteet WWW-NIMBUS -menetelmään [9].

Ensimmäisellä ratkaisukerralla aloituspisteet valittiin sallittujen välien keskivaiheilta. Valitaan nyt aloituspisteet, jotka ovat lähempänä sallittujen välien ylärajoja: $x_1 = 9$ ja $x_2 = 3$ (kuva 12.). Seuraavaksi tehdään ensimmäinen luokittelu (kuva 13.), jossa kohdefunktion f_2 arvo pyritään saamaan tasolle 1.0 ja kohdefunktioiden f_1 ja f_3 arvot taas saavat kasvaa ylärajaan 10.0 asti. Tilanne on siis toisella ratkaisukerralla päinvastainen kuin ensimmäisellä ratkaisukerralla, jolloin funktioiden f_1 ja f_3 arvoja pyrittiin minimoimaan

Closing

Thank you for solving the problem Chankong-Haines with NIMBUS.

The final solution is

| Function name | Value |
|---------------|----------|
| f1 | 2.786392 |
| f2 | 2.505533 |
| f3 | 2.229136 |

| Variable name | Value |
|---------------|----------|
| x1 | 2.583365 |
| x2 | 1.528532 |

Kuva 11: Lopullisen ratkaisun funktioiden ja muuttujien arvot.

Input Problem

Fill in all the fields, please. Note that the function and the variable names are used when displaying results, and the variable name

Objective functions to be optimized

| name | expression | Lowest Value | Highest Value |
|----------|---------------------|----------------------|----------------------|
| Min ▾ f1 | $(x1-1)^2+(x2-1)^2$ | <input type="text"/> | <input type="text"/> |
| Min ▾ f2 | $(x1-2)^2+(x2-3)^2$ | <input type="text"/> | <input type="text"/> |
| Min ▾ f3 | $(x1-4)^2+(x2-2)^2$ | <input type="text"/> | <input type="text"/> |

Use [Standard functions available](#)

Variables

Use [autofill](#)

| | lower bound | <= starting point | <= upper bound | Integer values |
|----|--------------------------------|-----------------------------------|------------------------------------|--------------------------|
| x1 | <input type="text" value="0"/> | <= <input type="text" value="9"/> | <= <input type="text" value="10"/> | <input type="checkbox"/> |
| x2 | <input type="text" value="0"/> | <= <input type="text" value="3"/> | <= <input type="text" value="4"/> | <input type="checkbox"/> |

Linear Constraints

| | Ax | b |
|-----|---|------------------------------------|
| No1 | <input type="text" value="1"/> *x1 + <input type="text" value="2"/> *x2 | <= <input type="text" value="10"/> |

Kuva 12: Lähtötietojen syöttäminen WWW-NIMBUS -menetelmään toisella ratkaisukerralla.

Classify Functions

Specify the direction of change from the current criterion vector by selecting an appropriate class classification instead, select the appropriate operation below.

| Function | Current solution | Class: | < | <= | = | >= | > |
|----------|------------------|--------|-----------------------|----------------------------------|-----------------------|-----------------------|----------------------------------|
| Min f1 | 2.679605 | | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> |
| Min f2 | 1.339437 | | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Min f3 | 2.679605 | | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> |

| Function | Lowest Value | Highest Value |
|----------|--------------|---------------|
| Min f1 | 0.0 | 10.0 |
| Min f2 | 0.0 | 5.0 |
| Min f3 | 0.0 | 10.0 |

Next optimization:

Maximum number of new solutions to be generated:

Kuva 13: Ensimmäinen luokittelu toisella ratkaisukerralla.

Analyse Results

Select one alternative for continuation (classification), generate intermediate solutions alternatives visually or stop.

- Alternative1** : f1 : 2.679605
f2 : 1.339437
f3 : 2.679605
- Alternative2** : f1 : 4.995012
f2 : 2.490535E-6
f3 : 4.995012
- Alternative3** : f1 : 3.483005
f2 : 0.347714
f3 : 3.483005

Kuva 14: Ensimmäisen luokittelun jälkeen saadut kaksi uutta ratkaisua toisella ratkaisukerralla.

ja funktion f_2 arvo taas sai nousta ylärajaan asti.

Näillä luokittelutiedoilla pystyttiin generoimaan vain kaksi uutta ratkaisua aloitusratkaisun lisäksi (kuva 14.). Ratkaisusta valittiin se, jossa funktiot f_1 ja f_3 saivat suurimmat arvot, ja funktion f_2 arvo oli pienin: $f_1 = 4.995012$, $f_2 = 2.490535 \cdot 10^{-6}$ ja $f_3 = 4.995012$.

Suoritetaan sitten toinen luokittelu, jossa pyritään vielä minimoimaan funktion f_2 arvoa. Funktioiden f_1 ja f_3 arvot saavat jälleen kasvaa ylärajaan 10.0 asti. Saadaan generoitua yksi uusi ratkaisu, jossa funktion f_2 arvo on saatu minimoitua edelleen pienemmäksi ja funktioiden f_1 ja f_3 arvot eivät ole kasvaneet merkittävästi: $f_1 = 4.999999$, $f_2 = 9.980651 \cdot 10^{-14}$ ja $f_3 = 4.999999$. Päätöksentekijä on tähän ratkaisuun tyytyväinen ja muuttujien arvoiksi saadaan tällöin $x_1 = 2.0$ ja $x_2 = 3.0$ (kuva 15).

Verrattaessa kuvissa 11. ja 15. esitettyjä ratkaisuja monitavoiteoptimoin-

Closing

Thank you for solving the problem chankong-haimes2 with NIMBUS.

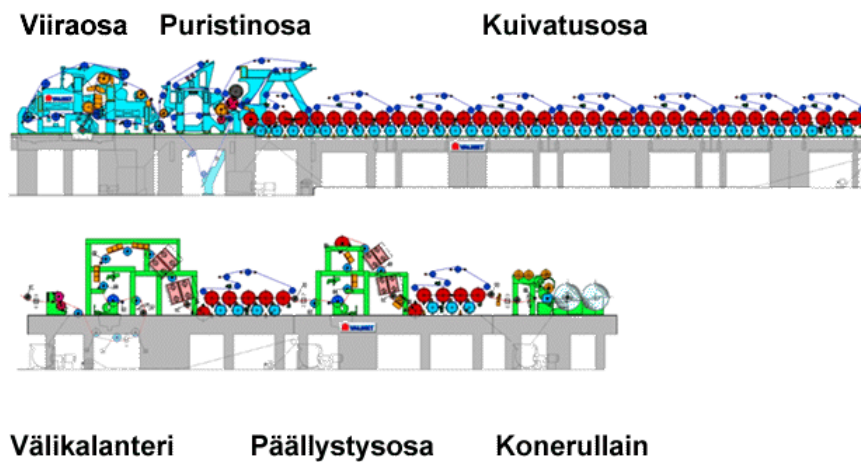
The final solution is

| Function name | Value |
|---------------|--------------|
| f1 | 4.999999 |
| f2 | 9.980651E-14 |
| f3 | 4.999999 |

| Variable name | Value |
|---------------|-------|
| x1 | 2.0 |
| x2 | 3.0 |

Kuva 15: Lopullisen ratkaisun funktioiden ja muuttujien arvot toisella ratkaisukerralla.

tiongelmalle (8) voidaan todeta, että saadut ratkaisut muuttuvat, kun päätöksentekijän antamat luokittelutiedot muuttuvat. Ensimmäisellä ratkaisukerralla päätöksentekijä pyrki löytämään ratkaisun, jossa kohdefunktioiden arvot olisivat mahdollisimman lähellä toisiaan (kuva 11.). Toisella ratkaisukerralla taas päätöksentekijä pyrki löytämään ratkaisun, jossa tavoitefunktio f_2 saisi mahdollisimman pienen arvon (kuva 15.). Lisäksi kummallakin ratkaisukerralla päätöksentekijät tekivät uusia luokitteluja sen mukaan, millaisia ratkaisuja WWW-NIMBUS -menetelmän avulla saatiin. NIMBUS-menetelmän avulla päätöksentekijät pystyivät siis oppimaan ongelmasta ja muuttamaan luokittelutietoja ratkaisuprosessin edetessä, mutta lisäksi erilaisilla luokittelutiedoilla ja tavoitteilla voitiin muodostaa samasta tehtävästä (8) kahteen erilaiseen päämäärään sopivia ratkaisuja.



Kuva 16: Paperikoneen laitteisto

5 Monitavoiteoptimoinnin hyödyntäminen paperin valmistuksessa

5.1 Paperikoneesta

Tyypillinen paperintekoprosessi sisältää kaksi perusosaa: paperin tuottamisen ja paperin viimeistelyn. Viimeistely voi olla myös erillinen prosessi. Paperin tuottamiseen tarvittava koneisto (kuva 16.) muodostuu useista eri osista, joista ensimmäinen on *perälaatikko*. Perälaatikon tehtävänä on sekoittaa paperimassa mahdollisimman tasaiseksi ja syöttää sitä sitten tasaisesti seuraavaan koneiston osaan, joka on *viira*. Viirat kuljettavat paperimassan edelleen kohti puristusosaa ja matkan aikana massasta haihtuu pois noin 20 prosenttia vettä. *Puristin* muistuttaa mankelointia ja se puristaa edelleen vettä pois massasta. Lopun kosteuden paperimassasta poistaa *kuivatusosa*, joka vie koneistossa eniten energiaa. Sitten paperi *päällystetään* sen mukaan, millaista paperia halutaan saada. Lopuksi *kalanteri* kiilottaa paperin ja muokkaa tarvittaessa sen paksuutta. Valmis paperi rullataan isoiksi konerulliksi, jotka saattavat painaa kymmeniä tuhansia kiloja. [3]

5.2 Monitavoiteoptimointi ja paperikone

Paperikoneen suunnittelussa ja toiminnassa tulee ottaa huomioon useita eri tekijöitä: paperin laatu, tehokkuus, tuotannon suuruus, kustannukset ja ympäristövaikutukset. Eri tavoitteet ovat yleensä ristiriidassa keskenään, jolloin sopivan tasapainon löytämiseksi voidaan käyttää monitavoiteoptimointia.

Tavoitteena on muodostaa virtuaalinen paperikone, jonka toimintaa pystytään simuloimaan. Virtuaalinen paperikone koostuu paperikoneen eri osien matemaattisten mallien simulaatiosta. Simulaatioon sisältyy niin paperin laatuun, kuin koneen ajettavuuteen ja tehokkuuteenkin vaikuttavat koneistot. Kun toteutettu virtuaalimalli on tarpeeksi luotettava, tarkka ja ennustettava, voidaan sen toimintaperiaate siirtää käytäntöön. Tällöin myös paperikoneeseen aiottuja muutoksia voidaan ensin kokeilla simulaation avulla ja sen jälkeen vasta oikealla paperikoneella. Seuraavaksi esitellään virtuaalisen paperikoneen tärkeimpiä ominaisuuksia.

Virtuaalisen paperikoneen tärkeimpiä vaatimuksia:

- Kokonaisen paperikoneen mallintamiseksi prosessin eri osien väliset suhteet ja reaktioketjut tulee saada toimimaan oikein.
- Virtuaalisen paperikoneen tulisi antaa nopeaa ja luotettavaa tietoa paperintekoprosessista, kuten esimerkiksi parametrien vuorovaikutuksen ja syy-seuraus -suhteiden toimivuudesta. Virtuaalisesta paperikoneesta tulee sen vuoksi löytyä laaja valikoima erilaisia virtuaalisia toimintamalleja.
- Virtuaalisen paperikoneen avulla tulisi voida joustavasti kokeilla ja analysoida erilaisia asetuksia ja käyttöolosuhteita. Kokeilu ja analysointi pitää lisäksi pystyä toteuttamaan niin koko koneen, kuin vain yksittäisen laiteosankin kohdalla.
- Virtuaalisen systeemin avulla tulisi pystyä automatisoimaan ainakin osa paperintekolaitteiston suunnitteluprosessista.
- Virtuaalisen paperikoneen tulee pysyä ajan tasalla uusien prosessien ja mallien kehittyessä; kun uusi simulaatiomalli on kehitetty, olisi sen sitten oltava myös virtuaalisesti saatavilla.
- Virtuaalinen paperikone tulee kehittää paperin tekoon erikoistuneelle ammattilaiselle, eikä esimerkiksi vain matemaatikolle sopivaksi. Virtuaalisen paperikoneen tulee siis olla myös suhteellisen helppokäyttöinen. [3]

6 NIMBUS-menetelmä paperinvalmistuksessa

Interaktiiviset monitavoiteoptimointimenetelmät sallivat päätöksentekijän oppia eri tavoitteiden välisistä riippuvuuksista, sekä erilaisten mieltymysten ja valintojen toimivuudesta ja merkityksestä kyseisen tehtävän ratkaisussa. Tällöin saadaan todennäköisesti ratkaisu, joka lopulta miellyttää päätöksentekijää. Interaktiivisista menetelmistä erityisesti NIMBUS-menetelmä on käyttäjäystävällinen ja ottaa erityisen hyvin huomioon päätöksentekijän mahdollisen oppimisen ja mielenmuutokset. Tällöin ei myöskään ole tarvetta kartoittaa päätöksentekijän arvojärjestyksiä etukäteen, kuten esimerkiksi painokertoimia käytettäessä. [8] Seuraavaksi esitellään kaksi erilaista tutkimusta [2] ja [3], joissa on NIMBUS-menetelmän avulla optimoitu paperikoneen toimintaa.

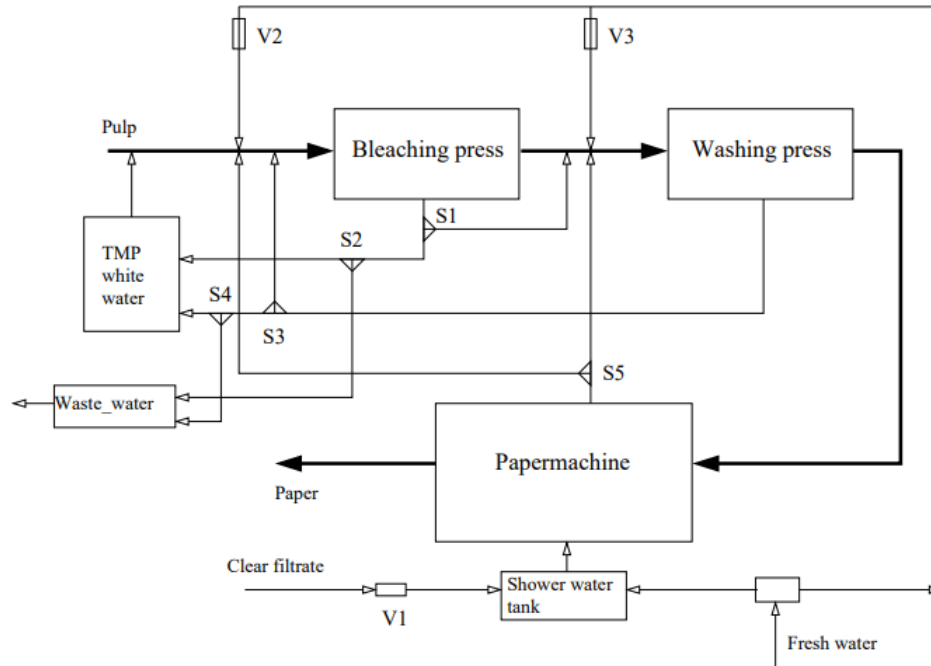
6.1 Vedenjakeluongelman asettelu

Artikkelissa [2] esitellään veden jakeluun liittyvä ongelma, joka ilmenee paperikoneen käytössä ja jonka ratkaisuun käytetään NIMBUS-menetelmää. Vedenkäsittelyä paperituotannossa ohjaa tavoite valmistaa paperia mahdollisimman tehokkaasti. Tuotantoprosessi vaatii vettä, jonka avulla hallinnoidaan paperin laatua ja koneiston ajettavuutta. Paperinteon alussa puuaines prosessoidaan selluloosaksi, mikä edellyttää jauhantaa ja valkaisua. Vettä tarvitaan muun muassa ennen ja jälkeen valkaisun. Sitten selluloosa johdetaan paperikoneeseen, jossa siitä valmistetaan paperia.

Ratkaistava ongelma sisältää kolme kohdefunktiota, joita tulee minimoida. Ensimmäinen funktio WW kuvaa liuenneen orgaanisen aineen pitoisuutta paperikoneen johdinvedessä. Toinen kohdefunktio B kuvaa liuenneen orgaanisen aineen pitoisuutta valkaisu prosessiin menevässä massassa (kg/1000 kg). Kolmas kohdefunktio FW kuvaa prosessissa tarvittavan tuoreen veden määrä (m³/ 1000 kg paperia).

Ongelma sisältää kahdeksan muuttujaa, jotka voidaan jakaa kahteen eri tyyppiin: jakajiin ja venttiileihin. Jakaja jakaa tulevan virran kahdeksi ulostuloksi. Sen arvo on nollan ja yhden välillä ja se kuvaa virtauksen osaa, joka menee ensimmäiseen ulostuloon. Venttiili taas kontrolloi veden määrää, joka tarvitaan prosessiin ja jonka yksikkö on kg/s.

Ensimmäiset kaksi muuttujaa ovat jakajia $S1$ ja $S2$, jotka kuvaavat valkaisu prosessista tulevan veden reittiä (Kuva 17.). Myös seuraavat kaksi muuttujaa $S3$ ja $S4$ ovat jakajia, jotka taas kuvaavat pesupuristimesta tulevan veden reittiä. Viides muuttuja $V1$ on venttiili, joka kontrolloi suodatteen määrää, joka otetaan paperikoneen suihkuvesitankkiin. Kuudes ja seitsemäs muuttuja



Kuva 17: Vedenjakeluongelmaa kuvaava kaavio. Tummat viivat kuvaavat paperimassan kulkureittiä ja ohuet viivat veden kulkureittiä.

$V2$ ja $V3$ ovat venttiilejä, jotka kontrolloivat veden määrää, joka otetaan valkaisuprosessiin ja pesupuristimeen. Viimeinen muuttuja $S5$ on jakaja, joka jakaa valkaisuvedettä valkaisun ja pesun välillä. [2]

6.1.1 Ongelmaratkaisuprosessi

Tutkimuksen [2] vedenjakeluongelman ratkaisussa käytettiin NIMBUS-menetelmää, jolloin pystyttiin myös hyödyntämään päätöksentekijän tietoa paperikoneen toiminnasta. Kohdefunktioiden ala- ja ylärajojen Pareto-optimaalisiksi ratkaisuksi arvioitiin $(261, 5.0, 4.9)^T$ ja $(2718, 21.0, 28.8)^T$. Pareto-optimaalinen lähtöpiste x^1 muodostettiin sitten syöttämällä ratkaisuarviot NIMBUS-algoritmiin, jolloin saatiin kohdefunktiolle arvot: funktiolle WW arvo 820, funktiolle B arvo 6.6 ja funktiolle FW arvo 12.1.

Päätöksentekijä halusi tutkia kohdefunktioiden WW ja B vaikutuksia vedenjakeluun, joten ensimmäinen luokittelu tehtiin veden kulutuksen minimoimiseksi: orgaanisen aineen osuutta johdinvessä lisättiin tasolle 1000

| Ratkaisu | WW (f_1)[mg/l] | B (f_2) [kg/1000 kg] | FW (f_3) [m^3 /tonni paperia] |
|-------------|--------------------|--------------------------|--------------------------------------|
| $f(x^1)$ | 820 | 6.6 | 12.1 |
| $f(x^2)$ | 1024 | 9.1 | 9.4 |
| $f(x^3)$ | 1693 | 10.0 | 7.5 |
| $f(x^4)$ | 1079 | 9.0 | 9.1 |
| $f(x^5)$ | 1143 | 9.0 | 8.9 |
| $f(x^6)$ | 1217 | 9.1 | 8.6 |
| $f(x^7)$ | 1305 | 9.2 | 8.3 |
| $f(x^8)$ | 1410 | 9.4 | 8.1 |
| $f(x^9)$ | 1536 | 9.7 | 7.8 |
| $f(x^{10})$ | 1444 | 13.6 | 7.3 |
| $f(x^{11})$ | 1328 | 10.0 | 8.1 |
| $f(x^{12})$ | 1359 | 11.1 | 7.8 |
| $f(x^{13})$ | 1397 | 12.2 | 7.5 |

Taulukko 1: NIMBUS-menetelmällä tuotetut kohdefunktioiden Pareto-optimaaliset arvot.

mg/l. Kohdefunktion B arvo oli sillä hetkellä tyydyttävä.

Tällöin saatiin luokittelu:

$I^<=3$, $I^-=2$, $I^>=1$ ja $\varepsilon_1=1000$.

NIMBUS-menetelmä tuotti neljä eri ratkaisua kohdefunktioille, joista päätöksentekijä valitsi seuraavan luokittelun lähtöpisteeksi ratkaisun $f(x^2)=(1024, 9.1, 9.4)^T$. Saatu ratkaisu on tuotettu aputehtävän (5), (6) tai (7) avulla, jolloin asetettu yläraja ε_1 ei ole niin kova, kuin esimerkiksi tehtävän (4) avulla ratkaistaessa.

Seuraavaksi päätöksentekijä halusi tutkia kohdefunktion WW vaikutuksia veden kulutukseen. Kohdefunktion B arvo pidettiin tasolla 9.1 ja funktion WW arvo nostettiin arvosta 1024 arvoon 1600 mg/l. Toinen luokittelu oli tällöin:

$I^<=3$, $I^-=2$, $I^>=1$ ja $\varepsilon_1=1600$.

Menetelmän avulla saatiin jälleen neljä eri ratkaisua, joista päätöksentekijä

valitsi ratkaisun $f(x^3)=(1693, 10.0, 7.5)^T$. Myös tämä ratkaisu on saatu apu-tehtävän (5), (6) tai (7) avulla.

Tämän jälkeen päätöksentekijä halusi generoida kuusi uutta ratkaisua kahden edellisen ratkaisun ($f(x^2)$ ja $f(x^3)$) väliltä. Saadut ratkaisut $f(x^4) - f(x^9)$ on esitetty taulukossa 1. Seuraavaksi lähtöpisteeksi päätöksentekijä valitsi ratkaisun $f(x^7) = (1305, 9.2, 8.3)^T$.

Päätöksentekijä halusi tutkia myös kohdefunktion B vaikutuksia vedenkäyttöön, jolloin kolmas luokittelu oli seuraava:

$I^<=3$, $I^-=1$, $I^>=2$ ja $\varepsilon_2=13.0$.

Tässä on uusi yläraja 13.0 funktiolle B. Saatiin kolme uutta ratkaisua, joista päätöksentekijä valitsi ratkaisun $f(x^{10}) = (1444, 13.6, 7.3)^T$. Tämä ratkaisu vähensi vedenkulutusta.

Lopuksi päätöksentekijä halusi generoida vielä kolme uutta ratkaisua arvojen $f(x^7)$ ja $f(x^{10})$ väliltä. Saadut ratkaisut $f(x^{11}) - f(x^{13})$ on esitetty taulukossa 1. Tämän jälkeen päätöksentekijä oli tyytyväinen saatuihin ratkaisuihin, eikä halunnut tehdä enää luokitteluja. Hän valitsi lopulliseksi ratkaisuksi ratkaisun $f(x^{11})$. Tällä ratkaisulla vedenkulutus on $8.1 \text{ m}^3/1000 \text{ kg}$ paperia ja orgaanisen aineen pitoisuus on 1328 mg/l johdinväessä ja $10.0 \text{ kg}/1000 \text{ kg}$ valkaisuun menevässä massassa.

Tutkimuksessa selvitettiin pääasiassa kohdefunktioiden WW ja B vaikutuksia vedenkulutuksen minimointiin ja NIMBUS-menetelmä tarjosi tähän hyvän mahdollisuuden. [2]

6.2 Paperintuotanto-ongelma

Tutkimuksessa [3] esitetty ongelma kuvaa esimerkkiä todellisesta optimointitehtävästä, jossa kohdefunktioiden arvot on tuotettu virtuaalisen paperikoneen avulla. Näin monimutkaista ongelmaa on mahdoton saada mallinnettua alusta loppuun vain yhdellä mallilla, joten on luotava useita osamalleja. Esitetyn ongelman ratkaisua varten luotiin yhteensä 23 eriliasta osaongelmaa.

Optimointiongelma sisältää kokonaisuudessaan yhdeksän optimoitavaa kohdefunktiota. Kolmea niistä maksimoidaan, ja kuutta minimoidaan. Nämä kohdefunktiot kuvaavat paperin tuotantoon ja laatuun vaikuttavia tekijöitä, kuten kuiva-ainepitoisuutta, ominaispainoa, β -formaatiota, vetolujuutta, kiiltoa ja PPS 10 -ominaisuuksia. Kaikki kohdefunktioiden arvot on normalisoitu, jotta niitä pystytään vertaamaan toisiinsa.

Optimointiongelma sisälsi kaikkiaan 23 muuttujaa, jotka ovat yleisesti tyypillisiä paperikoneen toimintaa ohjaavia tekijöitä. Paperin tuottamiseen erikoistunut ammattilainen osallistui päätöksentekoon olemalla päätöksentekijä, jolle optimointiongelman muuttujat olivat siis jo ennalta tuttuja. [3]

6.2.1 Ongelmanratkaisuprosessi

Tutkimuksessa [3] on ongelmanratkaisuprosessi esitetty seuraavasti: Päätöksentekijällä oli läpi ratkaisuprosessin selvät ajatukset siitä, millaisia arvoja hän halusi saavuttaa. Kuiva-ainepitoisuuden tulisi olla niin suuri kuin mahdollista, tavoitearvona 1.0. Ominaispainon tulisi olla lähellä arvoa 1.0 ja β -formaation ja vetolujuuden arvojen tulisi molempien kohdalla olla korkeintaan 1.0. Päätöksentekijä halusi lisäksi, että PPS 10 -arvojen (jotka kuvaavat paperin karkeutta) tulee olla 1.0 ja kiillon arvojen tulee olla vähintään 1.0. Päätöksentekijän mielestä paperin ylä- ja alareunan arvojen tulisi lisäksi olla lähellä toisiaan karkeuden ja kiillon osalta.

NIMBUS-menetelmällä toteutettu ratkaisuprosessi sisälsi kaikkiaan kahdeksan eri luokittelua ja kaksi väliratkaisujen generointia. Aluksi päätöksentekijä halusi kontrolloida erityisesti paperin laatuun liittyviä kohdefunktioita. Lisäksi hän piti tärkeimpänä tavoitteena ominaispainon arvon optimointia.

Ensimmäisen luokittelun avulla saavutettiin todella hyvä arvo ominaispainolle, sekä lisäksi β -formaation arvo parani. Sen jälkeen päätöksentekijä halusi pitää näiden muuttujien arvot samoina ja lisäksi parantaa vetolujuuden arvoa. Tämä ei kuitenkaan ollut mahdollista, jolloin todettiin, että β -formaation ja vetolujuuden arvojen välillä on ristiriita. Molempien muuttujien arvoja on siis mahdotonta kasvattaa samaan aikaan. Paperin laatuun vaikuttavien ominaisuuksien arvot, vetolujuutta lukuunottamatta, olivat toivotuilla tasoilla, joten päätöksentekijä siirtyi seuraavaksi optimoimaan paperin viimeistelyyn liittyviä tavoitteita.

Ensimmäiseksi hän halusi parantaa alareunan PPS 10 -arvoa, jotta se olisi lähempänä yläreunan arvoa. Lisäksi hän halusi parantaa myös kuiva-ainepitoisuutta. Muutamia toivottuja parannuksia pystyttiin saavuttamaan, mutta samaan aikaan ominaispainon arvo huononi. Päätöksentekijä valitsi ratkaisun, jossa ominaispainon arvo on erittäin hyvä, mutta kuiva-ainepitoisuus tarvitsi parannusta. Tämän jälkeen jatkettiin parantamalla kuiva-ainepitoisuutta ja pitämällä ominaispainon arvo nykyisellään. Täten löydettiin ratkaisu, jossa ominaispainon ja kuiva-ainepitoisuuden arvot olivat hyviä ja sen lisäksi β -formaation arvo parani ja kiillon arvo oli hyväksyttävällä tasolla. Tämä kuitenkin johti siihen, että vetolujuuden arvo huononi huomattavasti.

Tähän mennessä oli siis saatu kaksi hyvin erilaista ratkaisua ja päätöksentekijä päättikin generoida ratkaisun, joka olisi näiden kahden ratkaisun välistä. Toisen ratkaisun generoinnin jälkeen hän saavutti ratkaisun, jonka valitsi parannettavaksi luokittelun avulla. Tavoitteena oli saavuttaa pieniä parannuksia ilman, että tarvitsisi huonontaa saavutettuja hyviä arvoja.

Kahden luokittelun avulla päätöksentekijä pystyi parantamaan ominais-

| | Dry solids cont.(base paper) | Basis weight | β -formation | Tensile str. ratio | PPS 10 top | PPS 10 bottom | Gloss top | Gloss bottom | Dry solids cont. |
|-------|------------------------------|--------------|--------------------|--------------------|------------|---------------|-----------|--------------|------------------|
| | f1 | f2 | f3 | f4 | f5 | f6 | f7 | f8 | f9 |
| | Max | Min | Min | Min | Min | Min | Max | Max | Min |
| Start | 1.0091 | 1.0521 | 1.0556 | 1.0313 | 1.0275 | 1.1623 | 0.8666 | 0.9586 | 0.9979 |
| Cl. 1 | 1.0096 | 1.0003 | 1.0010 | 1.0975 | 0.9987 | 1.0600 | 1.1098 | 1.1543 | 1.0332 |
| Cl. 2 | 1.0096 | 1.0003 | 1.0010 | 1.0975 | 0.9987 | 1.0600 | 1.1098 | 1.1543 | 1.0332 |
| Cl. 3 | 1.0096 | 1.0000 | 1.0008 | 1.0976 | 0.9986 | 1.0600 | 1.1098 | 1.1543 | 1.0331 |
| Cl. 4 | 1.0081 | 1.0000 | 0.8716 | 1.1108 | 1.0513 | 1.0485 | 1.0902 | 1.0771 | 1.0074 |
| Cl. 5 | 1.0094 | 1.0589 | 0.9590 | 1.1199 | 1.0289 | 1.0495 | 1.0902 | 1.0771 | 1.0122 |
| Cl. 6 | 1.0095 | 1.0003 | 0.5243 | 1.3429 | 1.0836 | 1.1671 | 1.0897 | 1.0763 | 0.9996 |
| Ge. 1 | 1.0094 | 1.0139 | 0.5628 | 1.3102 | 1.0848 | 1.1634 | 1.0750 | 1.0690 | 0.9989 |
| Ge. 2 | 1.0095 | 1.0028 | 0.5319 | 1.3362 | 1.0838 | 1.1663 | 1.0868 | 1.0749 | 0.9995 |
| Cl. 7 | 1.0095 | 1.0026 | 0.8919 | 1.0470 | 1.0836 | 1.1156 | 1.0868 | 1.0749 | 1.0035 |
| Cl. 8 | 1.0096 | 1.0010 | 0.9788 | 0.9979 | 1.0826 | 1.1175 | 1.0868 | 1.0752 | 1.0034 |

Taulukko 2: Taulukko optimointitehtävän ratkaisun etenemisestä.

painoa, β -formaatiota ja vetolujuutta hyväksyttävälle tasolle asti. Tällöin kuitenkin kuiva-ainepitoisuuden ja PPS 10 arvot eivät saavuttaneet ihanteellista tasoa. Päätöksentekijä piti tilannetta kuitenkin tyydyttävänä, sillä kiillon saavuttamat arvot olivat jopa toivottua paremmat. Lisäksi sekä kiillon että karkeuden kohdalla paperin ylä- ja alareunan arvot olivat lähellä toisiinsa. Saadut ratkaisut on esitetty taulukossa 2, jossa lyhenne "Cl" tarkoittaa luokittelua ja "Ge" generointia. [3]

7 Päätelmät

Tutkimuksen perusteella monitavoiteoptimointi antaa uuden näkökulman monimutkaisiin ja ristiriitaisiin ilmiöihin ja auttaa päätöksentekijää löytämään parhaan mahdollisen ratkaisun. Pienetkin muutokset paperinvalmistusprosessissa saattavat vaikuttaa merkittävästi taloudelliseen tuottoon. Esimerkiksi yhden prosentin kasvu paperikoneen tuotossa tarkoittaa miljoonan euron arvoa myynnissä. Taloudellisesta näkökulmasta katsoen on loppujen lopuksi vain yksi tavoite, jota halutaan maksimoida: tuotto ja raha.

Mallinnuksen näkökulmasta katsoen ei ole aivan yksiselitteistä, miten kaikki virtuaalisen paperikoneiston osat ja niiden vuorovaikutus saadaan taloudellisen tavoitteen alle. Tämän vuoksi ratkaisun löytämiseen käytetään monitavoitteista optimointia interaktiivista mallinnusmenetelmää hyödyntäen. [3]

Molemmissa NIMBUS-menetelmän sovelluksissa päätöksentekijät ovat ratkaisuprosessin edetessä kokeilleet erilaisia vaihtoehtoja ja muuttaneet mieltään kohdefunktioiden luokitteluiden suhteen, kun ovat nähneet, miten funktioiden arvot vaikuttavat toisiinsa. Voidaan siis todeta, että NIMBUS-menetelmä on erityisen joustava monitavoiteoptimointiongelman ratkaisumenetelmä ja sen avulla pystytään valitsemaan päätöksentekijää eniten miellyttävä ratkaisu.

Kirjallisuutta

- [1] M. Emmerichl, A. Deutzi: *A tutorial on multiobjective optimization: fundamentals and evolutionary methods*
<https://link.springer.com/article/10.1007/s11047-018-9685-y>
(Luettu 20.1.2019)
- [2] J. Hakanen, K. Miettinen, M. M. Mäkelä: *An application of multiobjective optimization to process simulation*, (eds. P. Neittaanmäki et al.)
Proceedings of ECCOMAS 2004, Jyväskylä, Finland , 2004.
- [3] J. Hämäläinen, K. Miettinen, E. Madetoja, M. M. Mäkelä, P. Tarvainen:
Multiobjective decision making for papermaking, (ed. W. C. Wedley)
Proceedings of MCDM 2004, Whistler, British Columbia, Canada, 2004.
- [4] P. Korhonen: *Monitavoiteoptimointi*, luentomoniste, Helsingin kauppa-
korkeakoulu, Helsinki, 1980.
- [5] K. Miettinen: *Epälineaarinen optimointi*, luentomoniste, Jyväskylän yli-
opisto, Jyväskylä, 2003.
- [6] K. Miettinen: *Nonlinear multiobjective optimization*. Kluwer Academic
Publishers, Boston, 1999.
- [7] K. Miettinen, K. Kaario, M. M. Mäkelä: *Experiments with classification-
based scalarizing functions in interactive multiobjective optimization*,
European Journal of Operational Research 175(2) (2006), 931-947.
- [8] K. Miettinen, M. M. Mäkelä: *Synchronous approach in interactive mul-
tiobjective optimization*, European Journal of Operational Research
170(3) (2006), 909-922.
- [9] WWW-NIMBUS
<https://wwwnimbus.it.jyu.fi/N4/index.html>
(Luettu 23.4.2019)