



**TURUN
YLIOPISTO**

SYNKRONOITU LÄHESTYMISTAPA INTERAKTIIVISEEN
MONITAVOITEOPTIMOINTIIN

Julia Vehkaoja

LuK -tutkielma
Huhtikuu 2024

Ohjaaja: Dos. Yury Nikulin

MATEMATIIKAN JA TILASTOTIETEEN LAITOS

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

TURUN YLIOPISTO
Matematiikan ja tilastotieteen laitos

JULIA VEHKAOJA: Synkronoitu lähestymistapa interaktiiviseen monitavoiteoptimointiin

LuK -tutkielma, 23 s.

Sovellettu Matematiikka

Huhtikuu 2024

Tässä työssä käsitellään uutta lähestymistapaa interaktiiviseen monitavoiteoptimointiin. Menetelmää kutsutaan nimellä NIMBUS, ja se perustuu kohdefunktioiden luokitteluun. Lisäksi työssä esitellään skalaarifunktioita, joiden perusteella päätöksentekijä saa helpommin mieltymystensä mukaisen ratkaisun. Muista menetelmistä poiketen tässä hyödynnetään skalaarifunktioita synkronoidusti. Tämä tarkoittaa sitä, että päätöksentekijä valitsee eri laskettujen skalaarifunktioiden arvojen väliltä sopivimmat. Päätös ei siis kuulu menetelmää käyttävälle vaan päätöksentekijälle. Päätöksentekijän osallistuessa hän saa paremman kuvan ratkaisun mahdollisuuksista jokaisella iteraatiolla.

Työssä esitellään myös synkroninen muunnos NIMBUS-menetelmästä ja uusi versio WWW-NIMBUS, jonka toteutus on internetissä. WWW-NIMBUS on ohjelmistojärjestelmä, joka pystyy ratkaisemaan vaativia epälineaarisia ongelmia. Järjestelmän uusi versio on käyttäjäystävällisempi ja pystyy käsittelemään monipuolisesti monitavoiteoptimointiongelmia. Lisäksi käydään läpi esimerkki menetelmän käytöstä ja tehdään siihen kaksi uutta iteraatiota.

Uusi NIMBUS-menetelmä on siis tehokkaampi ja vastaa aikaisempaa paremmin päätöksentekijän tarpeisiin.

Asiasanat: Monitavoiteoptimointi; interaktiiviset menetelmät; NIMBUS; epälineaarinen ohjelmointi.

Sisällys

| | | |
|----------|--|-----------|
| 1 | Johdanto | 1 |
| 2 | Menetelmät | 2 |
| 3 | Skalaarifunktiot ja niiden käyttö | 2 |
| 3.1 | Skalarisoidut funktiot | 2 |
| 3.2 | Aliongelmat | 4 |
| 4 | NIMBUS-algoritmi | 5 |
| 4.1 | Synkronoitu NIMBUS-algoritmi | 6 |
| 4.2 | WWW-NIMBUS | 7 |
| 5 | Numeerinen esimerkki | 9 |
| 5.1 | Uudet iteraatiot | 17 |
| 6 | Yhteenveto | 22 |
| | Viiteet | 22 |

1 Johdanto

Optimointi on parhaan eli optimaalisen tuloksen etsimistä tarkasteltavalle ongelmalle matemaattisia menetelmiä hyödyntämällä. Kohdefunktio on funktio, jota minimoidaan tai maksimoidaan ongelmasta riippuen. Optimoinnissa on tärkeää, että pysytään annettujen rajoitteiden sallimassa alueessa. Rajoitteet siis rajoittavat kohdefunktion arvoa, ja kohdefunktiolta voidaan lisäksi vaatia tiettyjä vähimmäisarvoja rajoitteiden avulla.

Tässä tutkielmassa käsitellään monitavoiteoptimointia, jossa huomioidaan erityisesti interaktiivisuus eli päätöksentekijän mieltymykset optimaalista ratkaisua etsittäessä. Käytännössä päätöksentekijälle annetaan siis Pareto-optimaalisten pisteiden joukko, joiden avulla päätös tehdään. Tavoitteiden ristiriitaisuuden takia joudutaan tyytymään kompromisseihin eli Pareto-optimaalisiin pisteisiin, jotka toteuttavat jokaisen tavoitteen mahdollisimman hyvin. Kohdefunktiossa sallittu piste on Pareto-optimaalinen, mikäli ei ole yhtään sallittua pistettä, joka on ainakin yhtä hyvä kaikkien kriteerien suhteen ja aidosti parempi yhden kriteerin suhteen [1].

Interaktiivisessa toimintatavassa päätöksentekijä oppii tavoitteiden välisistä ristiriidoista ja pystyy suhtautumaan realistisesti niiden toteuttamismahdollisuuksiin. Haasteena on kuitenkin sama asia eli se, että päätöksentekijän täytyy tuntea käytettävissä olevat menetelmät sekä niiden toimintatavat ja mahdollisuudet. Lisäksi parasta tapaa määrittää ratkaisua ei ole, jos päätöksentekijä ei osallistu prosessiin. Päätöksentekijän osallistuminen prosessiin on siis välttämätöntä.

Työssä esitellään interaktiivinen menetelmä, joka perustuu luokitteluun ja kiintopisteisiin. Lisäksi skalaarifunktiot perustuvat samoihin päätöksentekijältä saatuihin alkutietoihin. Menetelmän avulla halutaan useita Pareto-optimaalisia ratkaisuja, joten käytetään myös useita skalaarifunktioita. Menetelmän avulla pyritään muodostamaan Pareto-optimaalisten ratkaisujen joukkoja, joista päätöksentekijä voi valita omien mieltymystensä mukaisesti parhaan.

Luvussa 2 perehdytään menetelmän teoriaan. Sen jälkeen esitellään skalaarifunktiota ja niiden hyödyntämistä optimointiongelmissa. Uusi skalaarifunktio, joka esitellään luvussa 3, on laskennallisesti tehokkaampi. Luvussa 4 esitellään NIMBUS-menetelmä ja sen synkronoitu algoritmi. NIMBUS-menetelmä pystyy arvioimaan ja ratkaisemaan ongelmia joustavammin kuin muut menetelmät. Menetelmän nimi tulee sanoista *Nondifferentiable Interactive Multiobjective BUndle-based optimization System*, ja se perustuu kohdefunktioiden luokitteluun [2].

Luvussa 4 perehdytään myös WWW-NIMBUS-menetelmään, jonka idea perustuu keskitettyyn laskentaan ja hajautettuun käyttöliittymään. Erillistä sovellusta ei siis tarvita, ja uusien versio on aina käytettävissä verkkosivuilla. Tämä ohjelmisto sisältää neljä erilaista skalarisointifunktiota, jotka perustuvat päätöksentekijän kerran ilmaisemiin luokittelutietoihin. Luvussa 5 esitetään numeerinen esimerkki ja tehdään siihen uusi iteraatio.

2 Menetelmät

Lähteiden [2, 3] mukaan monitavoiteoptimointiongelmat voidaan määritellä muodossa

$$\min \{f_1(x), f_2(x), \dots, f_k(x)\}, \quad (1)$$

jossa $x \in S$. Samalla, jos arvo k on suurempi kuin 2, on ristiriitaisia kohdefunktioita $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, joita halutaan minimoida samanaikaisesti.

Päätösvektorit $x = (x_1, x_2, \dots, x_n)^T$ kuuluvat epätyhjään joukkoon, jonka sallittu alue on $S \subset \mathbb{R}^n$. Nämä vektorit sisältävät myös kohdefunktioiden arvot

$$f(x) = (f_1(x), f_2(x), \dots, f_k(x))^T$$

ja nämä yhdessä muodostavat sallitun alueen kuvan. Päätösvektoreita $x' \in S$ kutsutaan Pareto-optimaaliseksi [4], jos ei esiinny toista päätösvektoria $x \in S$ siten, että $f_i(x) \leq f_i(x')$ kaikilla $i = \{1, \dots, k\}$ ja $f_j(x) < f_j(x')$ vähintään yhdelle indeksille j . Lisäksi kohdevektori on Pareto-optimaalinen, jos vastaava päätösvektori on myös Pareto-optimaalinen.

Yllä määriteltyjen ehtojen perusteella tiedetään, että Pareto-optimaalinen ratkaisu on olemassa. Sallittujen Pareto-optimaalisten ratkaisujen joukko kertoo, jos halutut tavoitteet ovat sallitun alueen ulkopuolella. Pareto-optimaalisen joukon alarajat ovat ihanteellisessa kohdevektorissa $z^* \in \mathbb{R}^k$. Sen komponentit z_i^* saavutetaan minimoimalla kohdefunktiot sallitulla alueella. Tästä voidaan muodostaa vielä tarkempi versio, joka on utopistinen kohdevektori z^{**} . Asetetaan $z_i^{**} = z_i^* - \varepsilon$ siten, että $i = 1, \dots, k$ ja $\varepsilon > 0$ on mahdollisimman pieni.

Pareto-optimaalisen joukon ylärajat merkitään z^{nad} , ja niitä on usein hankala saavuttaa erityisesti epälineaarisissa tapauksissa. Jatkossa oletetaan, että on saatavilla globaalit arviot Pareto-optimaalisten ratkaisujen vaihteluväleistä.

Pareto-optimaalisia ratkaisuja ei siis matemaattisesti voida asettaa täydellisesti paremmuusjärjestykseen, ja tämän takia päätöksentekijän rooli on merkittävä sopivaa ratkaisua etsittäessä. Päätöksentekijän vastuulle jää siis valita, mikä tai mitkä ratkaisuista palvelevat hänen tarkoituksiaan parhaiten.

3 Skalaarifunktiot ja niiden käyttö

Tässä luvussa esitellään luokittelu- ja kiintopistepohjaisia skalarisointifunktioita. Nämä reaaliarvoiset funktiot muodostetaan kohdefunktioiden avulla, missä otetaan huomioon päätöksentekijän antamat tiedot. Näiden avulla saadaan Pareto-optimaaliset ratkaisut alkuperäiseen ongelmaan. Skalaarifunktio tarkoittaa kuvausta $\mathbf{R}^n \rightarrow \mathbf{R}$, jossa skalaarifunktion derivaatta on vektorifunktio ja sitä kutsutaan gradientiksi [5].

3.1 Skalarisoidut funktiot

Luokittelussa päätöksentekijä ohjaa ratkaisun löytymistä Pareto-optimaalisten ratkaisujen joukosta. Interaktiivisen prosessin aikana päätöksentekijä voi vaikuttaa ratkaisun muodostumiseen jatkuvasti, mikä helpottaa sopivan ratkaisun löytymistä. Kohdefunktion arvot lasketaan nykyisen Pareto-optimaalisen päätösvektorin $x^c \in S$

avulla ja esitetään päätöksentekijälle. Tämän jälkeen päätöksentekijä voi ilmaista millaisia muutoksia hän haluaa, luokittelemalla kohdefunktiot eri luokkiin. Esitellään seuraavaksi lähteessä [2] esiteltyt viisi luokkaa, joita myös NIMBUS-menetelmässä käytetään:

- $I^<$, joiden arvoja pyritään parantamaan nykyisestä,
- I^{\leq} , joiden arvoja pyritään parantamaan haluttuun tavoitetasoon asti $\hat{z}_i < f_i(x^c)$,
- $I^=$, joiden arvot ovat hyväksytyjä nykyisessä ratkaisussa,
- I^{\geq} , joiden arvot voivat heikentyä johonkin ylärajaan asti $\varepsilon_i > f_i(x^c)$ ja
- I° , jonka arvot saavat väliaikaisesti muuttua vapaasti.

Tavoitetasot ja ylärajat luokitteluun saadaan päätöksentekijältä. Kiintopistemene-
telmässä päätöksentekijä määrittää kiintopisteet $\bar{z} \in \mathbf{R}^k$, ja ne koostuvat halutuista
tavoitetasoista \bar{z}_i jokaiselle funktiolle. Tämän kiintopisteen ei tarvitse riippua ny-
kyisestä ratkaisusta, vaan se osoittaa, millaista kohdefunktion arvoa päätöksentekijä
suosii.

Yleensä ideaali kohdevektori lasketaan ja näytetään päätöksentekijälle ennen luo-
kittelua tai kiintopisteiden valintaa. Lasketun kohdevektorin avulla päätöksenteki-
jällä on realistisempi käsitys ongelman mahdollisuuksista ja rajoituksista. Voidaan
siis olettaa, että päätöksentekijä valitsee vain sallittuja arvoja, joissa $\bar{z}_i \geq z_i^*$ ja
 $i = 1, \dots, k$.

Kiintopisteet voidaan luokitella annettujen tietojen perusteella, jos Pareto-optimaalinen
joukko tiedetään. Tässä tapauksessa asetetaan

$$\begin{aligned}\bar{z}_i &= z_i^* , \text{ jossa } i \in I^< , \\ \bar{z}_i &= z_i , \text{ jossa } i \in I^{\leq} , \\ \bar{z}_i &= f_i(x^c) , \text{ jossa } i \in I^= , \\ \bar{z}_i &= \varepsilon_i , \text{ jossa } i \in I^{\geq} \text{ ja} \\ \bar{z}_i &= z_i^{nad} , \text{ jossa } i \in I^{\circ} .\end{aligned}$$

Päätöksentekijän kannalta luotettava ja ymmärrettävä tapa kerätä mieltymystie-
toja on tärkeä osa interaktiivista menetelmää. Painokertoimien käyttö taas on esi-
merkki epäsuorasta ja epävarmasta tavasta ohjata ratkaisuprosessia. Käyttämällä
painokertoimia ei voida ennustaa, miten tulos muuttuu, eikä painotetulla summalla
ole ymmärrettävää merkitystä.

Toisaalta pyytämällä luokittelutietoja ja kiintopisteitä päätöksentekijältä saa-
daan suoraviivaisesti ensisijaisen tärkeitä tietoja. Päätöksentekijä käsittelee halut-
tuja kohdefunktioita ja näin ollen tiedonkäsittely on konkreettista ja selkeästi ym-
märrettävissä. Päätöksentekijän ei tarvitse hallita monimutkaisia käsitteitä, ja luo-
kittelun avulla on helppo verrata kohdefunktion arvojen muutoksia.

Seuraavassa luvussa esiteltävä NIMBUS-menetelmä perustuu luokitteluun, jos-
sa päätöksentekijä luokittelee jokaisen iteraation jälkeen kohdefunktiot korkeintaan

viiteen yllä määriteltyyn luokkaan $I^<$, I^{\leq} , $I^=$, I^{\geq} ja I° . Toimimalla näin pyritään löytämään päätöksentekijälle mieluisin Pareto-optimaalinen ratkaisu mahdollisimman vähillä iteraatiokerroilla. Luokittelu on siis mahdollista vain jos vähintään yhtä kohdefunktion arvoa halutaan vielä parantaa nykyisestä tasosta, jolloin $I^< \cup I^{\leq} \neq \emptyset$ ja $I^{\geq} \cup I^{\circ} \neq \emptyset$.

NIMBUS-menetelmässä aliongelman muodostaminen perustuu luokitteluun sekä samojen ylärajojen ja tavoitetasojen pitämiseen. Uudenlainen aliongelman muotoilu takaa Pareto-optimaaliset ratkaisut, ja se voidaan lähteen [2] mukaan muotoilla seuraavasti:

$$\min \max_{\substack{i \in I^< \\ j \in I^{\leq}}} \left[\frac{f_i(x) - z_i^*}{z_i^{nad} - z_i^{**}}, \frac{f_j(x) - \bar{z}_j}{z_j^{nad} - z_j^{**}} \right] + \rho \sum_{i=1}^k \frac{f_i(x)}{z_i^{nad} - z_i^{**}} \quad (2)$$

siten, että $f_i(x) \leq f_i(x^c)$ kaikilla $i \in I^< \cup I^{\leq} \cup I^=$, $f_i(x) \leq \varepsilon_i$ kaikilla $i \in I^{\geq}$ ja $x \in S$, jossa niin sanottu painokerroin $\rho > 0$ on suhteellisen pieni. Painokertoimen $\frac{1}{z_j^{nad} - z_j^{**}}$ käytön on osoitettu auttavan ymmärtämään päätöksentekijän mieltymyksiä hyvin ja se lisää myös laskennan tehokkuutta.

3.2 Aliongelmat

Esitellään seuraavaksi kolme aliongelmaa, joissa skalaarifunktiot perustuvat vertailupisteisiin. Vertailukohtat voidaan muodostaa myös luokittelun avulla ilman päätöksentekijän osallistumista ratkaisuprosessiin. Lisäksi näitä aliongelmiä voidaan käyttää NIMBUS-menetelmässä ja sen aliongelmissa.

Ensimmäinen aliongelmissa käytettävä skalaarifunktio on nimeltään STOM. Se käyttää kompromissimenetelmää

$$\min \max_{i=1, \dots, k} \left[\frac{f_i(x) - z_i^{**}}{\bar{z}_i - z_i^{**}} \right] + \rho \sum_{i=1}^k \frac{f_i(x)}{\bar{z}_i - z_i^{**}}, \quad (3)$$

jossa tavoitetasojen \bar{z}_i täytyy olla aidosti korkeampia kuin vastaavat komponentit utopistisessa vektorissa z_i^{**} .

Toinen aliongelmissa käytettävä skalaarifunktio on perusmuodossa

$$\min \max_{i=1, \dots, k} \left[\frac{f_i(x) - \bar{z}_i}{z_i^{nad} - z_i^{**}} \right] + \rho \sum_{i=1}^k \frac{f_i(x)}{z_i^{nad} - z_i^{**}}, \quad (4)$$

jossa $x \in S$. Jos nimittäjä asetetaan yhtä suureksi kuin $\bar{z}_i - z_i^{**}$, saadaan tulokseksi (3) yhtälö.

Viimeinen aliongelmissa käytettyä skalaarifunktiota kutsutaan nimellä GUESS, ja se on muotoa

$$\min \max_{i \notin I^{\circ}} \left[\frac{f_i(x) - z_i^{nad}}{z_i^{nad} - \bar{z}_i} \right] + \rho \sum_{i=1}^k \frac{f_i(x)}{z_i^{nad} - \bar{z}_i}, \quad (5)$$

jossa $x \in S$.

Pareto-optimaalisuuden takaamiseksi käytetään painokerrointa, jota alkuperäisessä muotoilussa ei ollut. Lisäksi skalaarifunktioon täytyy tehdä muutoksia, sillä vertailupisteet ovat peräisin luokittelusta, ja luokittelussa esiintyvällä nolllalla jakamista on vältettävä. Tämän takia korvataan nimittäjä summalausekkeessa termillä $z_i^{nad} - z_i^{**}$, kun $i \in I^\circ$. Tällöin myös tavoitetasojen \bar{z}_i täytyy olla aidosti pienempiä kuin kohdevektorin z_i^{nadir} komponentit. Aliongelma perustuu nadir-kohdevektoriin, joka täytyy arvioida. Menetelmän toimivuus ei kuitenkaan ole riippuvainen arvioinnin oikeellisuudesta.

Merkittävin ero verrattaessa esiteltyjä aliongelmia esiintyy kohdan (2) rajoitteissa. Näillä rajoitteilla taataan kohdefunktioiden arvojen väheneminen sekä se, että määritellyt ylärajoja ei ylitetä. Toisaalta muut aliongelmat käyttävät annettujen tietoja ja vertailupisteitä enemmän ehdotusten kaltaisesti ilman lisärajoituksia. Tämä ero tiedon käsittelyssä johtuu siitä, että luokittelutiedot perustuvat nykyiseen ratkaisuun, jossa vertailupisteet voidaan muodostaa itsenäisesti nykyisestä ratkaisusta. Lisäksi esiteltyt aliongelmat ovat Pareto-optimaalisia, sillä painokerroin takaa sen.

Lause 1. *Oletetaan, että kohdan (2) ratkaisu on Pareto-optimaalinen määritelmän (1) perusteella.*

Todistus. Olkoon $x^* \in S$ ratkaisu kohtaan (2). Todistetaan tämä vastaoletuksen avulla. Oletetaan siis, että ratkaisu ei ole Pareto-optimaalinen. Tässä tapauksessa on olemassa toinen $x' \in S$ siten, että $f_i(x') \leq f_i(x^*)$ kaikilla $i = 1, \dots, k$. Lisäksi epäyhtälö pätee ainakin yhdelle indeksille j . Tämä tarkoittaa, että x' on mahdollinen kohdassa (2). Nyt saadaan:

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

ja

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

Tällöin x^* ei voi olla kohdan (2) ratkaisu. Päädytään ristiriitaan, jolloin x^* on Pareto-optimaalinen. \square

4 NIMBUS-algoritmi

Tässä luvussa perehdytään NIMBUS-algoritmiin ja sen interaktiivisiin mahdollisuuksiin. Lisäksi esitellään WWW-NIMBUS-menetelmä ja sen toimintaperiaate.

Synkronoitua NIMBUS-menetelmää on käytetty monimutkaisten ongelmien ratkaisemiseen kuten paperikoneiden ja paperinvalmistusprosessin suunnitteluun. Menetelmää hyödyntäneiden asiantuntijoiden mukaan nykyinen synkronoitu versio on monipuolisempi kuin aiemmat versiot. He ovat saaneet paremman käsityksen siitä, miten ongelma voidaan ratkaista antamatta liikaa mieltymystietoja.

Lisäksi lähteen [6] mukaan NIMBUS-menetelmää voidaan hyödyntää myös jätevedenpuhdistamoissa, sillä jätevedenkäsittelyongelma on laskennallisesti kallis. Yritykselle aiheutuvia kustannukset pyritään siis minimoimaan. Esimerkkinä käytetään yritystä Pöyry Finland Oy. Tässä kuitenkin lisätään NIMBUS-menetelmään PAINT-menetelmä, jonka avulla iteraatioita voidaan tehdä entistä nopeammin. PAINT-menetelmä interpoloi Pareto-optimaalisten ratkaisujen ja annetun joukon välillä. Tämän jälkeen menetelmä muodostaa korvaavan ja laskennallisesti edullisen sekalukuratkaisun alkuperäiselle ongelmalle.

4.1 Synkronoitu NIMBUS-algoritmi

Päätöksentekijä saattaa haluta ratkaisuja Pareto-optimaalisten ratkaisujen väliltä. Tässä tapauksessa otetaan yhtä pitkät askelmat päätösavaruudessa ja niitä vastaavat kohdevektorit annetaan kiintopisteiksi kohtaan (4). Tämän avulla saadaan Pareto-optimaaliset ratkaisut.

Mahdollisten aliongelmiin ja keskiarvoisten ratkaisujen takia ratkaisujen lukumäärä saattaa olla suuri. Esitellään seuraavaksi keinoja, joilla päätöksentekijä voi välttää suuria ratkaisumääriä tietokannassa.

Muodostetaan perusaskeleet synkronoidulle NIMBUS-algoritmille. Merkitään tallennettujen ratkaisujen joukkoa kirjaimella A ja asetetaan se alussa tyhjäksi $A = \emptyset$. Ratkaisuprosessin aloituskohta voi tulla päätöksentekijältä tai se voi olla luonnollinen kompromissi kahden tavoitteen välillä.

1. Määritä Pareto-optimaalinen aloituspiste.
2. Pyydä päätöksentekijää luokittelemaan kohdefunktiot nykyisessä ratkaisussa ja tarkentamaan mahdolliset tavoitetasot sekä ylärajat.
3. Pyydä päätöksentekijää valitsemaan ratkaisujen maksimimäärä ja ratkaise niin monta aliongelmaa.
4. Esitä uudet ratkaisut päätöksentekijälle.
5. Lisää yksi tai useampi ratkaisu joukkoon A , jos päätöksentekijä hyväksyy ne.
6. Siirry kohtaan 8, jos päätöksentekijä ei halua keskiarvoratkaisuja kahden ratkaisun väliltä. Mikäli haluaa, pyydä päätöksentekijää valitsemaan kaksi ratkaisua, joko uusista ratkaisuista tai joukosta A .
7. Luo haluttu määrä keskiarvoratkaisuja ja vertaa niitä Pareto-optimaaliseen joukkoon. Sen jälkeen siirry vaiheeseen 4.
8. Pyydä päätöksentekijää valitsemaan paras vaihtoehtoista. Tämä saa olla joko uusi ratkaisu ja/tai keskiarvoratkaisu tai ratkaisu, joka on joukossa A . Merkitse sitä nykyisenä ratkaisuna. Jos päätöksentekijä haluaa jatkaa, siirry vaiheeseen 2. Muussa tapauksessa lopeta.

Algoritmi lopetetaan, jos päätöksentekijä ei halua enää parantaa minkään kohdefunktion arvoa tai hän ei halua, että minkään kohdefunktion arvo pienenee. Muuten

parhaan ratkaisun etsiminen jatkuu iteratiivisesti liikkumalla Pareto-optimaalisessa joukossa.

Useampia aliongelmiä hyödynnetään synkronoidussa NIMBUS-algoritmissa kolmesta eri syystä. Ensimmäiseksi menetelmän kehittäjien ei tarvitse korjata yhtä skalaarifunktiota, eikä heidän tarvitse päättää, mikä ratkaisusta on paras, vaan päätös on päätöksentekijän. Toinen syy on se, että useamman ratkaisun avulla saadaan parempi kuva Pareto-optimaalisesta joukosta, joka on päätöksentekijän mieltymysten mukainen.

Kolmas hyöty useamman aliongelman hyödyntämisessä on se, että samaa mieltymystietoa voidaan hyödyntää eri tavoilla. Esimerkiksi aliongelmassa (2) noudatetaan tarkasti annettua ylärajaa I^{\geq} , kun taas muut aliongelmat voivat ylittää rajat. Jos uutta ratkaisua ei ole mahdollista luoda nykyisten rajoitusten takia, päätöksentekijä saa mahdollisuuden muuttaa rajoitteita, jotta halutut parannukset voidaan toteuttaa. Ratkaisuprosessia voidaan siis jatkaa palaamatta alkuun, kokeilla uutta luokittelua ja relaksoida ylärajoja.

NIMBUS-algoritmin paras ominaisuus on monipuolisuus ja päätöksentekijällä on useita mahdollisia strategioita ohjata ratkaisuprosessia. Jos päätöksentekijä ei halua vertailla ratkaisuja, hän voi käyttää pelkästään luokittelua ja yhtä aliongelmaa sekä päättää jokaisen iteraation kohdalla jatketaanko nykyisellä vai uudella ratkaisulla. Vaihtoehtoisesti päätöksentekijä voi valita mieluisen ratkaisun Pareto-optimaalisesta joukosta tai käyttää useita aliongelmiä ja keskiarvoratkaisuja.

Algoritmin joustavuus tarkoittaa myös sitä, että päätöksentekijä voi muuttaa strategiaansa iteraatioiden välissä. Aikaisemmat valinnat eivät siis rajoita tulevia valintoja. Synkroninen lähestymistapa on sopiva sellaiselle päätöksentekijälle, joka on valmis näkemään erilaisia ratkaisuja lisäämättä uusia tavoitteita ja preferenssejä.

Ratkaisut eivät kuitenkaan välttämättä ole erilaiset, vaikka ratkaisuissa olisi käytetty eri aliongelmiä tai keskiarvoratkaisuja. Tässä tapauksessa esitetään vain aiemmin saaduista ratkaisuista eroavat ratkaisut. On myös mahdollista, että kaikki aliongelmat johtavat samaan ratkaisuun. Tätä voidaan pitää osoituksena aliongelmiä kestävydestä, mikä on myös arvokasta tietoa.

Tutkimalla eri skalaarifunktioiden avulla saatuja ratkaisuja päätöksentekijä voi oppia ongelmasta ja verrata millaisia ratkaisuja saadaan, kun luokittelutietoja käytetään vaihtelevalla tarkkuudella. Tämä nopeuttaa interaktiivista ratkaisuprosessia, kun jokaisen iteraation aikana voidaan oppia mieltymyksistä. Tällöin iteraatioita tarvitaan vähemmän kuin muita menetelmiä hyödyntäen.

4.2 WWW-NIMBUS

NIMBUS-algoritmin interaktiivisuus ei rajoita sen monipuolisuutta monitavoiteoptimoinnissa. Ainoa rajoittava tekijä on perustana oleva yksitavoitteisen tehtävän ratkaisumalli, jonka täytyy käsitellä myös aliongelmat ja mahdolliset erityistapaukset.

WWW-NIMBUS perustuu keskitettyyn tietojenkäsittelyyn ja hajautettuun käyttöliittymään. Koska tämä toimii internetissä, se ei aseta vaatimuksia käyttöjärjestelmälle tai käytössä olevalle kääntäjälle. WWW-NIMBUS luo html-sivuja, jotka perustuvat käyttäjän antamaan syötteeseen.

Epälineaaristen monitavoiteoptimointiongelmiä ratkaisemiseen on tarvittu ohjelmistoa, ja sitä tarvetta vastaamaan suunniteltiin WWW-NIMBUS. Kohdefunktioita minimoidaan tai maksimoidaan, ja sallittu alue S saattaa sisältää epälineaarisia sekä lineaarisia epäyhtälöitä. Lisäksi yhtäsuuruusrajoitukset sekä ylä- ja alarajat voivat olla lineaarisia tai epälineaarisia.

Optimointitehtävä voidaan ratkaista joko syöttämällä tiedot verkkosivuilla tai käyttämällä aliohjelmia. Aliohjelman käyttö mahdollistaa vaikeampien ongelmien ratkaisemisen ilman analyttisten funktioiden muodostamista. Ratkaisussa, jossa käyttäjä luokittelee kohdefunktiot ensimmäisen kerran, käytetään seuraavaa aliongelmaa

$$\bar{z}_i = \frac{z_i^{nad} + z_i^{**}}{2}, \text{ jossa tavoitetasot ovat } i = 1, \dots, k.$$

Tästä tuloksena saadaan neutraali kompromissi, joka on hyvä aloituspiste, kun päätöksentekijän mieltymyksistä ei ole vielä saatavilla tietoa. Seuraavaksi luokitellaan kohdefunktiot joko symbolisesti tai graafisesti tietokoneen hiiren avulla. Näkymän palkkien avulla saa tietoa nykyisistä kohdefunktion arvoista sekä jokaisen kohdefunktion Pareto-optimaalisen joukon.

Luokittelun jälkeen käyttäjä voi valita, kuinka monta ratkaisua hän haluaa ja kuinka monta vastaavaa aliongelmaa muodostetaan luokittelua varten annetun informaation perusteella. Uusimmassa WWW-NIMBUS versiossa lokaalit ja globaalit ratkaisut ovat saatavilla, ja päätöksentekijä voi jokaisen luokittelun jälkeen valita mitä optimointialgoritmia käytetään. Lokaalia ratkaisua etsittäessä voidaan käyttää proksimaattista bundle-menetelmää, jonka avulla voidaan ratkaista myös ei-differentioituvia ongelmia ja joka olettaa että kohde- ja rajoitefunktiot ovat määrittelyjoukossaan Lipschitz-jatkuvia. Käyttämällä bundle-menetelmää tarvitaan myös tietoja gradientista, mutta verkkosivuja käytettäessä näitä tietoja ei tarvita.

Käyttäjä voi valita geneettisen algoritmin globaaliksi ratkaisijaksi. Geneettinen algoritmi pyrkii löytämään optimaalisen ratkaisun todennäköisyydellä 1, ja sen avulla voidaan saada hieman optimiratkaisua heikompia ratkaisuja lyhyemmällä ajoajolla eli pienemmällä iteraatioiden toistomäärällä [7].

Tässä tapauksessa ratkaisu saattaa sisältää kokonaislukumuuttujia. Geneettiset algoritmit yleisessä muodossa eivät kykene ratkaisemaan lineaarisia ja epälineaarisia rajoitteita tehokkaasti. Toistaiseksi tätä ratkaisukykyä kuitenkin vaaditaan aliohjelmien toiminnan kannalta. Tämän takia WWW-NIMBUS sisältää kaksi erilaista geneettistä algoritmia eri rajoitteidenkäsittelymenetelmillä. Ensimmäinen menetelmä hyödyntää mukautuvia sakkofunktioita ja toisessa käytetään parametrittomia sakkoja. Kaikki optimointialgoritmit sisältävät teknisiä parametreja, ja käyttäjä voi tarvittaessa vaihtaa oletusarvoja.

Käyttäjää suositellaan usein valitsemaan nopeampi lokaali ratkaisutapa, jos hän ei ole varma käyttäisikö lokaalia vai globaalialgoritmia. Silloin tällöin globaalilla ratkaisijalla voidaan tarkistaa, onko epäkonveksisuus aiheuttanut sen, että ratkaisu on pysähtynyt lokaaliin optimiin. Geneettisistä algoritmeista kumpaakin voi käyttää, mutta parametritonta menetelmää suositellaan, koska se ei sisällä ylimääräisiä parametreja.

Jokaisen luokittelun jälkeen systeemi antaa halutun määrän ratkaisuja ja vaih-

toehtoisten ratkaisujen määrä nousee, jos käyttäjä haluaa keskiarvoisia ratkaisuja. Tämän takia eri ratkaisujen vertailu on tärkeää ja käyttäjä voi missä tahansa vaiheessa tallentaa mielenkiintoisia ratkaisuja ratkaisutietokantaan. Uusien ja tallennettujen ratkaisujen vertailua helpottaa visuaalinen ratkaisu. Käyttäjä voi valita muun muassa erilaisia pylväskaaviota sekä terälehtikaavion. Visuaalisessa esitystavassa voidaan käyttää sekä absoluuttisia että suhteellisia asteikkoja. Käyttäjä voi jättää osan vaihtoehdoista pois ennen siirtymistä seuraavaan vaiheeseen.

WWW-NIMBUS-järjestelmässä käyttäjä voi tallentaa, ladata ja muokata ongelmia. Lisäksi joustava tietokanta mahdollistaa aina valittujen ratkaisujen tallentamisen. Tämä lisää käyttäjystävällisyyttä, sillä edellisiin ratkaisuihin on helppo palata, jos ne muodostuvatkin mielenkiintoisiksi. Käyttäjä voi myös helposti graafisesti havainnollistaa joukon valittuja ratkaisuja ja luoda keskiarvoratkaisun minkä tahansa kahden eri ratkaisun välille. Sovelluksen rakenteen vuoksi kaikki käsitellyt ratkaisut ovat Pareto-optimaalisia.

5 Numeerinen esimerkki

Tässä luvussa esitellään NIMBUS-algoritmin avulla tehty esimerkki, joka on ratkaistu käyttämällä WWW-NIMBUS-järjestelmää. Tässä esimerkissä on kuusi kohdefunktiota ja rajat kahdelle muuttujalle, kuten kuvasta 5 voidaan nähdä. Ensimmäistä kohdefunktiota maksimoidaan ja muita minimoidaan. Lisäksi kaikki muuttujat ovat jatkuvia.

Kun ongelma on määritelty WWW-NIMBUS avulla, se muodostaa kuvassa 5 näkyvät tiedot. Nämä tiedot sisältävät Pareto-optimaalisen joukon ja arvioidut vaihteluvälit. WWW-NIMBUS-menetelmä antaa kaksi erilaista luokittelumahdollisuutta. Ensimmäisessä luokittelussa symbolien avulla nähdään, että toisen, viidennen ja kuudennen kohdefunktion arvot laskivat sekä muiden kohdefunktioiden rajoitteita löysennettiin.

Järjestelmää pyydettiin luomaan korkeintaan kaksi uutta ratkaisua ja luokitteluparametrin määrittely näkyy kuvassa 3. Tavoitetaso tulee olla nykyisen ja ideaalin kohdearvon välissä, ja nämä arvot voidaan nähdä kuvassa 3 toiselle kohdefunktiolle. Vastaavasti nykyiset ja tavoitearvot muodostavat rajat funktioille, joiden arvot saavat heiketä.

Kuvasta 4 voidaan nähdä luokittelun lähtökohta sekä sen perusteella saadut kaksi uutta ratkaisua. Huomataan, että viidennen ja kuudennen kohdefunktion arvoa ei voida pienentää samanaikaisesti. Valitaan siis vaihtoehdot 2 ja 3. Luodaan niiden avulla kaksi uutta keskiarvoratkaisua, jotka ovat

$$(4.6, 7.2, -4.3, 0.2, 1450.0, 1805.6)$$

ja

$$(4.5, 7.2, -4.3, 0.2, 1434.7, 1822.5).$$

Kyseiset ratkaisut nähdään myös kuvassa 5. Ensimmäinen näistä eli vaihtoehto 2 kuvassa 5 valittiin lähtöpisteeksi seuraavaa vaihetta varten.

Valinnan jälkeen asetetaan $I^< = \{6\}$, $I^{\leq} = \{2\}$, $I^{\geq} = \{3,4,5\}$ ja $I^{\circ} = \{1\}$ siten, että $z_2 = 4.0$, $\varepsilon_3 = -3.0$, $\varepsilon_4 = 1.0$ ja $\varepsilon_5 = 1600.0$. Nämä näkyvät kuvissa 6 ja 7.

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 x_1 , x_2 , x_3 , etc. must be used when defining the problem.

Objective functions to be optimized

| | name | : expression | Lowest Value | Highest Value |
|-------|------|--|--------------|---------------|
| Max ▾ | f1 | : $x_1 \cdot x_2$ | | |
| Min ▾ | f2 | : $(x_1 - 4)^2 + x_2^2$ | | |
| Min ▾ | f3 | : $-x_1 - x_2$ | | |
| Min ▾ | f4 | : $x_1 - x_2$ | | |
| Min ▾ | f5 | : $50 \cdot x_1^4 + 10 \cdot x_2^4$ | | |
| Min ▾ | f6 | : $30 \cdot (x_1 - 5)^4 + 100 \cdot (x_2 - 3)^4$ | | |

Use [Standard functions available](#)

Variables

Use [autofill](#)

| | lower bound | <= starting point | <= upper bound | Integer values |
|----|-------------|-------------------|----------------|--------------------------|
| x1 | : 1 | <= 1 | <= 3 | <input type="checkbox"/> |
| x2 | : 1 | <= 1 | <= 3 | <input type="checkbox"/> |

Next optimization: Global 1 + Local (PB) ▾

Kuva 1: Ongelman kuvaus WWW-NIMBUS sivustolla.

Hybrid solver used: Global 1 + Local (PB)

Classify Functions

Specify the direction of change from the current criterion vector by selecting an appropriate class for each function. **To use graphical classification instead, select the appropriate operation below.**

| Function | Current solution | Class: | < | <= | = | >= | > |
|----------|------------------|--------|----------------------------------|----------------------------------|-----------------------|----------------------------------|-----------------------|
| Max f1 | 5.437906 | | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Min f2 | 9.124742 | | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Min f3 | -4.669013 | | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> |
| Min f4 | -0.2192304 | | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> |
| Min f5 | 1582.05 | | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Min f6 | 1788.815 | | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

| Function | Lowest Value | Highest Value |
|----------|--------------|---------------|
| Max f1 | 1.0 | 9.0 |
| Min f2 | 2.0 | 18.0 |
| Min f3 | -6.0 | -2.0 |
| Min f4 | -2.0 | 2.0 |
| Min f5 | 60.0 | 4860.0 |
| Min f6 | 480.0 | 9280.0 |

Next optimization:

Maximum number of new solutions to be generated:

Kuva 2: Symbolinen luokittelu.

Enter Classification Parameters

Aspiration functions

Enter desired level(s)

Min f2 2.0 <= <= 9.124742

Upper and lower bound functions

Enter boundary values

Max f1 1.0 <= <= 5.437906

Min f3 -4.669013 <= <= -2.0

Min f4 -0.2192304 <= <= 2.0

Kuva 3: Määritellään luokitteluparametrit.

Hybrid solver used: Global 1 + Local (PB)

Analyse Results

Select one alternative for continuation (classification), generate intermediate solutions between two selected alternatives, compare the alternatives visually or stop.

- Alternative1** : f1 : 5.437906
f2 : 9.124742
f3 : -4.669013
f4 : -0.2192304
f5 : 1582.05
f6 : 1788.815
- Alternative2** : f1 : 4.587083
f2 : 7.177962
f3 : -4.289251
f4 : 0.2221359
f5 : 1465.477
f6 : 1788.815
- Alternative3** : f1 : 4.523969
f2 : 7.197285
f3 : -4.259347
f4 : 0.2148569
f5 : 1419.552
f6 : 1839.552

Number of new alternatives : <= 15

Kuva 4: Tulokset ensimmäisen luokittelun jälkeen.

Analyse Results

Select one alternative for continuation (classification), generate intermediate solutions between two selected alternatives, compare the alternatives visually or stop.

- Alternative1** : f1 : 4.587083
f2 : 7.177962
f3 : -4.289251
f4 : 0.2221359
f5 : 1465.477
f6 : 1788.815
- Alternative2** : f1 : 4.565998
f2 : 7.184297
f3 : -4.279283
f4 : 0.2197096
f5 : 1450.045
f6 : 1805.607
- Alternative3** : f1 : 4.54496
f2 : 7.190739
f3 : -4.269315
f4 : 0.2172832
f5 : 1434.737
f6 : 1822.519
- Alternative4** : f1 : 4.523969
f2 : 7.197285
f3 : -4.259347
f4 : 0.2148569
f5 : 1419.552
f6 : 1839.552

Kuva 5: Luodut keskiarvoratkaisut.

Toisin sanoen toisessa luokittelussa toisen ja kuudennen kohdefunktion arvoa pienennetään ja ensimmäisen kohdefunktion arvo saa muuttua vapaasti. Lisäksi muille kohdefunktioille asetettiin yläraja.

| Function | Current solution | Class: | < | <= | = | >= | > |
|----------|------------------|--------|----------------------------------|----------------------------------|-----------------------|----------------------------------|----------------------------------|
| Max f1 | 4.565998 | | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> |
| Min f2 | 7.184297 | | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Min f3 | -4.279283 | | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> |
| Min f4 | 0.2197096 | | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> |
| Min f5 | 1450.045 | | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> |
| Min f6 | 1805.607 | | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

| Function | Lowest Value | Highest Value |
|----------|--------------|---------------|
| Max f1 | 1.0 | 9.0 |
| Min f2 | 2.0 | 18.0 |
| Min f3 | -6.0 | -2.0 |
| Min f4 | -2.0 | 2.0 |
| Min f5 | 60.0 | 4860.0 |
| Min f6 | 480.0 | 9280.0 |

Next optimization:

Maximum number of new solutions to be generated:

Kuva 6: Toisen luokittelun arvojen sallittu muutos.

Järjestelmää pyydettiin näiden avulla muodostamaan neljä uutta ratkaisua. Luokittelun aloituspiste ja neljä uutta ratkaisua näkyvät kuvassa (8). Tämä kuva auttaa myös hahmottamaan ratkaisun visuaalisesti. Saaduista ratkaisuista voidaan valita mikä vain tai vaihtoehtoisesti luokittelun aloituspiste.

Aspiration functions

Enter desired level(s)

Min f2 2.0 \leq \leq 7.184297

Upper and lower bound functions

Enter boundary values

Min f3 -4.279283 \leq \leq -2.0

Min f4 0.2197096 \leq \leq 2.0

Min f5 1450.045 \leq \leq 4860.0

Select operation

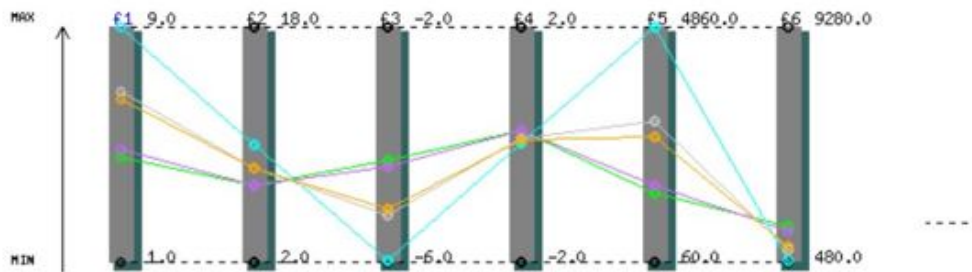
- Another problem Calculate a new solution (continue)
 Correct Highest or Lowest values Show the whole problem

Kuva 7: Toisen luokittelun rajoitteet.

PETAL DIAGRAM OF THE ALTERNATIVES



VALUE PATHS IN THE RELATIVE RANGE OF VALUES
Different alternatives have different colours



| Alternative: | 1 | 2 | 3 | 4 | 5 |
|--------------|-----------|-----------|-----------|-----------|-----------|
| f 1 : | 4.565998 | 4.789932 | 8.997004 | 6.807204 | 6.502192 |
| f 2 : | 7.184297 | 7.184297 | 9.99401 | 8.385318 | 8.37093 |
| f 3 : | -4.279283 | -4.383159 | -5.999001 | -5.218907 | -5.100221 |
| f 4 : | 0.21970 | 0.22882 | 0.00099 | 0.09040 | 0.05901 |
| f 5 : | 1450.045 | 1600.0 | 4858.922 | 2915.513 | 2617.729 |
| f 6 : | 1805.606 | 1652.745 | 480.0 | 911.3129 | 1034.856 |

Colour: ■ ■ ■ ■ ■

-- : Current classification

Kuva 8: Vaihtoehtojen visualisointi.

5.1 Uudet iteraatiot

Tehdään kaksi uutta iteraatiota käyttämällä samaa aloituspistettä toisella iteraatio-
kierroksella. Valitaan siis aloituspisteeksi

$$(4.6, 7.2, -4.3, 0.2, 1450.0, 1805.6).$$

Ensimmäisessä valitaan, että $I^= = \{1,2\}$, $I^{\leq} = \{3,4\}$, $I^{\geq} = \{5,6\}$. Tämä näkyy
kuvassa 9. Annetaan lisäksi rajat, joiden välillä funktioiden arvot saavat vaihdella.
Nämä rajoitteet näkyvät kuvassa 10.

| Function Current solution Class: | | < | <= | = | >= | > |
|----------------------------------|-----------|-----------------------|----------------------------------|----------------------------------|----------------------------------|-----------------------|
| Max f1 | 4.565998 | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Min f2 | 7.184297 | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Min f3 | -4.279283 | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Min f4 | 0.2197096 | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Min f5 | 1450.045 | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> |
| Min f6 | 1805.607 | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> |

| Function | Lowest Value | Highest Value |
|----------|--------------|---------------|
| Max f1 | 1.0 | 9.0 |
| Min f2 | 2.0 | 18.0 |
| Min f3 | -6.0 | -2.0 |
| Min f4 | -2.0 | 2.0 |
| Min f5 | 60.0 | 4860.0 |
| Min f6 | 480.0 | 9280.0 |

Next optimization:

Maximum number of new solutions to be generated:

Kuva 9: Ensimmäisen uuden iteraation arvojen sallittu vaihtelu.

Näiden valintojen kautta saadaan ratkaisu, joka näkyy kuvassa 11.

Toisessa uudessa iteraatiossa valitaan myös luoduista keskiarvoratkaisuista vaihto-
toehto 2, joka näkyy kuvassa 5. Luodaan sen avulla uudet ratkaisut. Valitaan, että
 $I^{\circ} = \{1,2,3,4,5,6\}$, jolloin kaikki arvot saavat kasvaa vapaasti. Tällaisen valinnan
avulla saadaan muodostettua vain yksi uusi ratkaisu, joten ratkaisut ovat

$$(4.6, 7.2, -4.3, 0.2, 1450.0, 1805.6)$$

Aspiration functions

Enter desired level(s)

Min f3 -6.0 <= <= -4.669013

Min f4 -2.0 <= <= -0.2192304

Upper and lower bound functions

Enter boundary values

Min f5 1582.05 <= <= 4860.0

Min f6 1788.815 <= <= 9280.0

Select operation

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

Kuva 10: Ensimmäisen uuden iteraation rajoitteet.

ja

$(9.0, 10.0, -6.0, 0.0, 4860.0, 480.0)$.

Esitetään sama vielä kuvien avulla.

Vertaillaan seuraavaksi eri rajoitteilla saatuja uusia ratkaisuja. Valitaan ensimmäiseksi tarkasteltavaksi ratkaisuksi kuvan 11 vaihtoehto 4, joka saa arvot

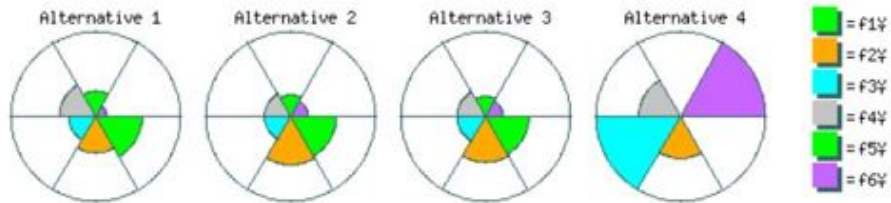
$(1.0, 10.0, -2.0, 0.0, 60.0, 9280.0)$.

Verrataan tätä ratkaisua kuvan 14 vaihtoehtoon 2, joka saa arvot

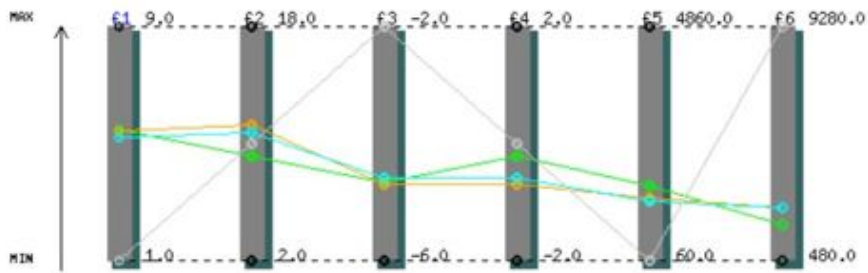
$(9.0, 10.0, -6.0, 0.0, 4860.0, 480.0)$.

Toinen ja neljäs kohdefunktio saavat samat arvot, mutta huomataan, että muut saavat rajoitteiden sallimissa rajoissa täysin päinvastaiset arvot. Alkuperäisessä tehtävässä pyritään maksimoimaan ensimmäistä kohdefunktiota ja minimoimaan kaikkia muita. Päätöksentekijän vastuulla on valita, kumpi ratkaisusta on hänelle mieluisempi. Tämä kuitenkin osoittaa, kuinka erilaisia ratkaisuja menetelmän avulla voidaan saada.

PETAL DIAGRAM OF THE ALTERNATIVES



VALUE PATHS IN THE RELATIVE RANGE OF VALUES
Different alternatives have different colours



| Alternative: | 1 | 2 | 3 | 4 |
|--------------|-----------|----------|-----------|--------|
| f 1 : | 5.437906 | 5.395259 | 5.192266 | 1.0 |
| f 2 : | 9.124742 | 11.2772 | 10.7399 | 10.0 |
| f 3 : | -4.669013 | -4.69763 | -4.596133 | -2.0 |
| f 4 : | -0.21923 | -0.69762 | -0.59613 | 0.0 |
| f 5 : | 1582.05 | 1329.577 | 1254.263 | 60.0 |
| f 6 : | 1788.815 | 2430.835 | 2432.66 | 9280.0 |

Colour: ■ ■ ■ ■

-- : Current classification

Kuva 11: Ensimmäisen uuden iteraation ratkaisu.

| | Function | Current solution | Class: | < | <= | = | >= | > |
|-----|-----------------|-------------------------|---------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------------------------|
| Max | f1 | 4.565998 | | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> |
| Min | f2 | 7.184297 | | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> |
| Min | f3 | -4.279283 | | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> |
| Min | f4 | 0.2197096 | | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> |
| Min | f5 | 1450.045 | | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> |
| Min | f6 | 1805.607 | | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> |

Kuva 12: Toisen uuden iteraation edellistä vapaammat rajoitteet.

Hybrid solver used: Global 1 + Local (PB)
 Warning: Notice that only 1 new alternatives could be generated.

Analyse Results

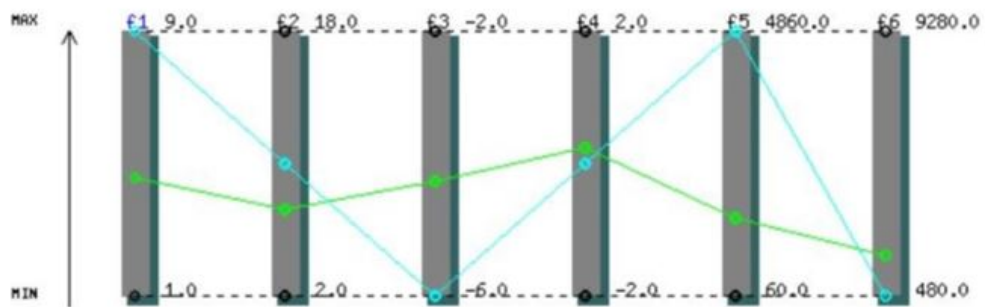
Select one alternative for continuation (classification), generate intermediate solutions between two selected alternatives, compare the alternatives visually or stop.

- Alternative1** : f1 : 4.565998
 f2 : 7.184297
 f3 : -4.279283
 f4 : 0.2197096
 f5 : 1450.045
 f6 : 1805.607
- Alternative2** : f1 : 9.0
 f2 : 10.0
 f3 : -6.0
 f4 : 0.0
 f5 : 4860.0
 f6 : 480.0

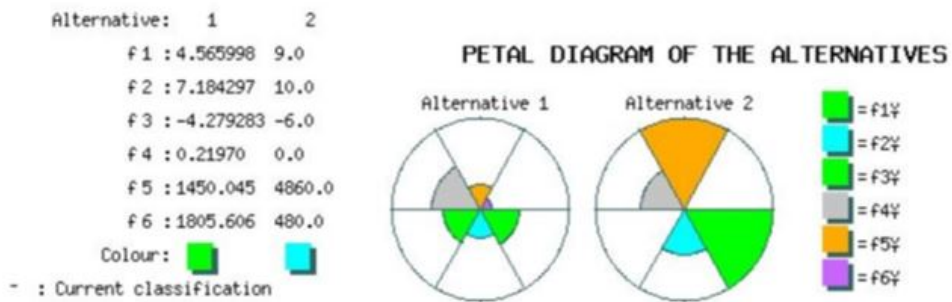
Kuva 13: Toisen uuden iteraation ratkaisut.

VALUE PATHS IN THE RELATIVE RANGE OF VALUES

Different alternatives have different colours



Visualizations



Kuva 14: Toisen uuden iteraation ratkaisun visualisointi.

6 Yhteenveto

NIMBUS-menetelmä [8] on aikaisemmin käytössä olleita menetelmiä monipuolisempi. Menetelmässä tavoitetasot ja ratkaisujen luokittelu nopeuttaa päätöksentekijän mieltymysten mukaisen ratkaisun löytymistä. Menetelmää käyttämällä ratkaisu saadaan usein siis myös pienemmillä iteraatiokerroilla. Lisäksi menetelmän avulla pystytään arvioimaan ja ratkaisemaan ongelmia joustavammin kuin muilla menetelmillä.

Interaktiivisuus päätöksentekijän kanssa on myös merkittävä osa menetelmää, sillä ratkaisuprosessin aikana päätöksentekijä voi vaikuttaa ratkaisun suuntaan. Lisäksi strategiaa voidaan muuttaa missä tahansa vaiheessa, joka lisää käyttäjäystävällisyyttä. Strategian muuttaminen ei kuitenkaan vie ratkaisuprosessia alkuun, mikä nopeuttaa myös lopullisen ratkaisun saamista.

Päätöksentekijälle käyttäjäystävällisyyttä lisää myös mahdollisuus tallentaa tuloksia ja palata niihin, jos tulokset muodostuvat kiinnostaviksi. Laskettu kohdevektori antaa myös realistisemmän käsityksen ongelman ratkaisumahdollisuuksista. Menetelmän avulla saadaan siis parempi käsitys ongelman mahdollisista ratkaisuisista antamatta liikaa tietoa menetelmää käyttävälle. Lisäksi graafiset ratkaisut ovat selkeitä ja niistä on helppo tulkita ratkaisu.

Monitavoiteoptimoinnissa käytetyillä skalaarifunktiolla ja aliongelmienvälillä on todella tärkeä rooli. Menetelmä on akateemisessa käytössä ja tosielämän ongelmassa todettu hyväksi ja vastaanotto on ollut positiivinen.

Viitteet

- [1] Marko M. Mäkelä: *Matemaattinen optimointi I*. Luentomoniste, luettu 20.2.2024.
- [2] Kaisa Miettinen, Marko M. Mäkelä: *Synchronous approach in interactive multiobjective optimization*. European Journal of Operational Research, 2004.
- [3] Kaisa Miettinen: *Nonlinear Multiobjective Optimization*. Kluwer Academic, Boston, 1999.
- [4] Vilfredo Pareto: *The Mind and Society*. Harcourt, Brace, 1935.
- [5] Jan Thompson, Thomas Martinsson: *Matematiikan käsikirja*. Helsinki, Tammi, 1994.
- [6] Markus Hartikainen, Vesa Ojalehto, Kristian Sahlstedt: *Applying the approximation method PAINTE and the interactive method NIMBUS to the multiobjective optimization of operating a wastewater treatment plant*. Engineering Optimization, 3/2015.
- [7] Atidel Ben Hadj-Alouane, James C. Bean: *A Genetic Algorithm for the Multiple-Choice Integer Program*. Operations research, 2/1997.
- [8] Kaisa Miettinen, Marko M. Mäkelä: *Interactive Multiobjective Optimization System WWW-NIMBUS on the Internet, Computers & Operations Research*. Vol. 27 (7-8), s. 709-728, 2000.