

# **Mat-2.177**

## **Operaatiotutkimuksen projektityöseminaari**

### **Loppuraportti**

**19.4.2004**

Projekti	Monitavoitteinen portfolio-optimointi tiestön päällystämishankkeiden valinnassa
Asiakas	Inframan Oy
Yhteyshenkilö	Jaakko Dietrich, Jaakko.Dietrich@inframan.fi
Projektiryhmä	Juuso Liesiö, PP, Juuso.Liesio@hut.fi Pekka Mild, Pekka.Mild@hut.fi Lauri Salminen, Lauri.Salminen@iki.fi Liisa Ikonen, Liisa.Ikonen@hut.fi

## Tiivistelmä

Tiehallinnon tehtävänä on muodostaa vuotuinen päällystysohjelma valitsemalla laajasta hanke-ehdokasjoukosta toteutettava osajoukko rajallisen budjetin puitteissa. Päätöksentekovaiheessa hanke-ehdokkaista on käytettävissä kuntotietodataa usean kriteerin suhteen. Nykyisellään päällystysohjelma laaditaan ensisijaisesti hyödyntäen yhtä näistä kriteereistä. Tässä työssä päällystysohjelman valintaa tarkastellaan monitavoitteisen päätöksenteon menetelmin, pyrkimyksenä muodostaa ehdokasjoukosta ohjelma, jonka vaikutukset ovat mahdollisimman suotuisat kaikkien valittujen kriteerien suhteen.

Työssä rakennettu malli ratkaisee (approksimatiivisesti) ehdokasjoukosta kaikki tehokkaat päällystysohjelmat. Ohjelma on tehokas, jos annetuilla resursseilla ei voida muodostaa toista sellaista ohjelmaa, joka on kaikkien kriteerien suhteen vertailuohjelmaa parempi. Malli huomioi päätöksentekijältä saatavissa olevan preferenssi-informaation, joka voi olla myös epätäydellistä, esimerkiksi ainoastaan kriteerien tärkeysjärjestys. Malli ei nykyisellään ehdota, mikä (mahdollisesti tuhansista) tehokkaista ohjelmista tulisi valita, vaan esittää koko tehokkaan ohjelmajoukon tuottamat suoritustasot eri kriteerien suhteen. Lisäksi malli liittää kuhunkin hanke-ehdokkaaseen indeksin välillä 0-100 %, joka kuvaa, kuinka suureen osaan tehokkaista ohjelmista kyseinen hanke kuuluu.

Rakennettua mallia testattiin kahdella todellisella hanke-ehdokasaineistoilla. Tietokoneajoilla tarkasteltiin mallin avulla muodostettujen tehokkaiden päällystysohjelmien tuloksia eri kriteerien suhteen nykyisen valintakäytännön mukaisen heuristiikan tuottamaan ohjelmaan. Toisesta ehdokasaineistosta oli käytettävissä myös todellinen, toteutettu päällystysohjelma, johon mallin generoimia ohjelmia voitiin verrata. Ohjelmat muodostettiin kolmen kriteerin perusteella (peittopros., ajokust. ja kestoikä)

Testien tulokset ovat varsin rohkaisevia. Malli tuottaa lukuisia tehokkaita ohjelmia, jotka ovat kaikkien valittujen kriteerien suhteen sekä heuristiikan tuottamaa että todellista ohjelmaa parempia. Toisaalta, todellisen vertailuohjelman (kustannus 16,3 M€) mukaiset suoritustasot valittujen kriteerien suhteen kyettiin saavuttamaan jo 15 M€ hintaisella ohjelmalla. Mallin tuottamia ohjelmia ei kuitenkaan ole esitetty päätöksentekijöiden arvioitaviksi niiden kokonaisvaltaisen subjektiivisen mielekkyyden arvioimiseksi.

# Sisällysluettelo

1	Johdanto .....	1
1.1	Taustaa .....	1
1.2	Tavoitteet .....	1
1.3	Rajaukset.....	2
1.4	Raportin rakenne.....	2
2	Päällystysohjelman valinta.....	3
2.1	Tieverkon kuntotiedot ja niiden kerääminen .....	3
2.2	Päällystysohjelman valinta tiepiireissä .....	4
3	Monitavoitteinen optimointi .....	7
4	Mallin rakentaminen .....	15
5	Testidatan mallintaminen.....	20
5.1	Koesuunnitelma .....	20
5.2	Tulokset.....	22
6	Johtopäätökset.....	33
7	Viitteet.....	35

# **1 Johdanto**

## **1.1 Taustaa**

Liikenteen aiheuttama tiestön rasitus on kasvanut 1990-luvulla voimakkaasti, vaikka kuljetussuorite ei ole juurikaan lisääntynyt. Tämä johtuu siitä, että kuormakoko on kasvanut ja kuljetuksissa käytetään entistä raskaampaa kalustoa. Jos tulevaisuudessa kuljetussuorite lisääntyy ja siirtyminen yhä raskaampaan kalustoon jatkuu, tulee tiestön rasitus lisääntymään erittäin voimakkaasti (Belt et. al., 2002).

Tiestön nykyisen kunnan säilyttäminen nykyisellä tasolla vaatii lisäämään ylläpito- ja korjauskustannuksia tulevaisuudessa (Lämsä, 2001). Samalla teiden ylläpitoon käytettävien resurssien tehokas käyttö muuttuu yhä tärkeämmäksi. Tieverkon kuntotiedon kerääminen ja sen tehokkaampi hyväksikäyttö ovatkin olleet voimakkaan kehitystyön kohteena viimeisen kymmenen vuoden aikana.

Tämän projektityön toimeksiantajana on väyläomaisuuden hallintaan erikoistunut konsulttiyritys Inframan. Vuonna 1993 perustettu yritys keskittyy infrastruktuurin kunnossapidon konsultointiin sekä kehittää apuvälineitä ylläpidon päätöksenteon tueksi.

Inframanin keskeisin asiakas on Tiehallinto. Tiehallinnon alaiset tiepiirit laativat vuosittaisen tiestön kunnossapito-ohjelman, josta yhden osan muodostaa tässä työssä käsiteltävä päällystysohjelma. Tiehallinto kerää monipuolista tietoa päällystetyn tiestön kunnosta. Kuntotietojen perusteella tiestöstä muodostetaan kunnostusta vaativat tieosuudet eli hanke-ehdokkaat, joita on tiepiiriä kohden muutamia satoja. Näistä hankkeista valitaan kunnostettavat kohteet eli päällystysohjelma vuotuisen budjetin rajoissa. Tyypillisen vuotuisen päällystysohjelman kustannus on 15-20 miljoonaa euroa.

## **1.2 Tavoitteet**

Työn tavoitteena on kehittää Tiehallinnon päällystysohjelman valintamenetelmää. Tällä hetkellä Tiehallinto hyödyntää olemassa olevaa dataa päällystysohjelman valinnassa varsin rajoitetusti. Hankejoukosta valitaan toteutettavat hankkeet lähinnä yhden kuntotietoaineistosta johdetun, huonokuntoisen tien osuutta kuvaavan, mittarin perusteella.

Työssä tutkitaan monitavoitteisen päätöksenteon, erityisesti monitavoitteisen portfolio-optimoinnin käyttömahdollisuuksia vuosittaisen päällystysohjelman valinnassa. Työssä rakennetaan malli, joka tukee hankkeiden monitavoitteista analysointia ja esitetään algoritmi mallin ratkaisemiseksi. Lisäksi hahmotellaan tämän mallin käyttöönottoa osana laajempaa päätöstukiprosessia. Mallin avulla analysoidaan kahta asiakkaan toimittamaa todellista hankejoukkoa. Toinen hankejoukko sisältää todellisen valitun päällystysohjelman, jota vertaamme mallin antamiin tuloksiin. Näin voidaan analysoida mallin tuomaa lisähyötyä päätöksentekoon.

### **1.3 Rajaukset**

Työssä ei puututa Tiehallinnon tiedonkeruumenetelmiin tai kerätyn datan luonteeseen, vaan malli rakennetaan tehostamaan jo olemassa olevan datan käyttöä. Malli rakennetaan tukemaan ainoastaan yhden vuoden päällystysohjelman valintaa, koska projektin puitteissa ei ole mahdollista mallintaa hanke-ehdokkaiden kunnossa tapahtuvaa kehitystä. Mallilla analysoidaan ainoastaan asiakkaan toimittamia hankejoukkoja eikä implementoida päätöstukiohjelmistoa, jolla mallia voitaisiin soveltaa muihin aineistoihin.

### **1.4 Raportin rakenne**

Raportissa esitellään aluksi Tiehallinnon keräämää dataa ja sen käyttöä. Kappaleessa 2 esitellään tiepiirikohtainen päällystysohjelman valintaongelma ja tämänhetkinen valintakäytäntö. Kappaleessa 3 käsitellään monitavoitteisen optimoinnin teoriaa ja sen hyödyntämismahdollisuuksia päällystysohjelman valinnassa. Teoriaan pohjautuen rakennetaan tarpeisiin sopiva malli kappaleessa 4. Lopuksi kuvataan testiaineistolla tehtyjä esimerkkiajoja ja esitellään tulokset. Testiajojen tuloksia verrataan nykyisen käytännön mukaisiin menetelmiin ja arvioidaan mallin avulla saavutettavaa lisähyötyä. Lopussa pohditaan työn asiakkaalle tuomaa lisähyötyä sekä hahmotellaan mahdollisia jatkotoimenpiteitä.

## **2 Päälystysohjelman valinta**

### **2.1 Tieverkon kuntotiedot ja niiden kerääminen**

Tiehallinto kerää tietoa tiestön kunnosta monilla eri mittausmenetelmillä. Mittauksia tehdään esimerkiksi tien pituusprofiilista, poikkiprofiilista, vaurioiden määrästä tiellä sekä tien kantavuudesta. Nykyisin tiestön kuntorekisteriin talletetaan mittausten perusteella tiedot 100 metrin jaksoilta seuraavien kuntomuuttujien osalta (Tielaitos, 1999):

#### **Tasaisuus (mm/m)**

Tasaisuutta kuvataan kansainvälisellä tasaisuusindeksillä, IRI (International Roughness Index). IRI kuvaa 80 km/h liikkuvan matemaattisella mallilla lasketun neljännesajoneuvon korin pystysuoraa liikettä pyörän suhteen. Tasaisuus mitataan palvelutasomittausautolla (PTM) lasertekniikan avulla.

#### **Urasyvyys (mm)**

Tien poikkiprofiili mitataan PTM-auton 17 ultraäänianturilla kahden metrin välisinä otoksina. Kullekin 100 millimetrin matkalle määritetään maksimiarvo, joista lasketaan urasyvyyden keskiarvo 100 metrin osuudelle. Lisäksi urasyvyyden kehityksestä muodostetaan ennuste historiatietojen perusteella. Tätä ennustetta käytetään mm. mahdollisten korjaus- tai parannustoimenpiteiden kestoikää arvioitaessa.

#### **Vauriosumma (m<sup>2</sup>)**

Päälysteessä esiintyvät vauriot inventoidaan visuaalisesti päälysteen vaurioinventointilaitteella (PVI). Yksittäiset vauriot (verkko-, pituus-, poikki- ja saumahalkeamat, reiät, purkautumat, reunapainumat sekä paikat) inventoidaan erikseen metreinä, neliömetreinä tai kappaleina. Eri vauriotyypit luokitellaan vakavuusasteen perusteella yhteen tai kahteen luokkaan. Inventoinnin tuloksista lasketaan painotettu vauriosumma. Myös vauriosumman muutosnopeudesta laaditaan ennuste, jota käytetään mahdollisten parannus- tai korjaustoimenpiteiden arvioinnissa.

#### **Kantavuusaste (%)**

Päällystettyjen teiden kantavuudet mitataan pudotuspainolaitteella (PPL). Laite mittaa tiehen syntyvää taipumaa kuudella eri etäisyyksillä olevilla antureilla, kun tiehen kohdistetaan 50 kN kuormitus. Mitattu maksimitaipuma lämpötilakorjataan ensin 20 °C lämpötilaan, jonka jälkeen määritetään kevätkantavuuskertoimella korjattu kevätkantavuus. Saatua kevätkantavuutta verrataan kuormituskertaluvun ja suunnitteluohjeiden perusteella laskettuun tavoitekantavuuteen. Kantavuusaste on kevätkantavuuden ja tavoitekantavuuden välinen suhde.

Urasyvyys ja tasaisuus mitataan pääteiltä joka vuosi ja muilta teiltä joka toinen vuosi. Vaurioituvilta teiltä inventoidaan päällystevauriot kolmen vuoden kierrolla. Kantavuusmittaus suoritetaan keskimäärin 400-500 metrin välein. Kantavuusasteen laskennassa tarvittavat taipumat mitataan 4-5 vuoden välein. Muille 100 metrin jaksoille lasketaan kantavuusaste interpoloimalla. Mitattujen tietojen lisäksi käytetään kuntoennustemalleja, joilla ennustetaan 1-3 vuoden ikäinen tasaisuus, urasyvyys tai vauriosumma ensin nykytilaan ja siitä tarvittaessa 1-3 vuotta eteenpäin (Belt & al, 2000).

## **2.2 Päällystysohjelman valinta tiepiireissä**

Tiehallinnon keskipitkän ja pitkän tähtäimen toiminta- ja taloussuunnitelma antaa strategiset suuntaviivat tiepiireille. Se laaditaan neljän vuoden tähtäimellä ja sitä päivitetään vuosittain. Tiepiirit vastaavat itsenäisesti oman alueensa korjaus- ja parannushankkeiden valinnasta. Tätä valittua korjaus- ja parannushankelistaa kutsutaan päällystysohjelmaksi (Ruotoistenmäki 2003).

Päällystysohjelmat muodostetaan kaikissa yhdeksässä tiepiirissä 1-3 vuoden aikavälille. Päällystysohjelman muodostaminen aloitetaan kokoamalla kuntomittausaineiston perusteella suurehko joukko potentiaalisia parantamis- ja korjaushankkeita. Näin saatavista sadan metrin pituisista tiejaksoista muodostetaan sopivan mittaisia hanke-ehdokkaita kahden eri vaiheen avulla: Ensimmäin suoritetaan automaattinen yhdistely, jolla yhdistetään vierekkäisiä 100 metrin pätkiä pidemmiksi osuuksiksi niille suositeltujen toimenpiteiden perusteella. Automaattisen yhdistelyn jälkeen päällystysohjelman tekijä yhdistää manuaalisesti haluamiaan osuuksia ja tekee niihin muita tarvittavia muutoksia.

Lopullisten hanke-ehdokkaiden määrä on tyypillisesti 200-1500. Näistä hanke-ehdokkaista valitaan annetun budjetin rajoissa korjattavat tai parannettavat tiet. Päällystysohjelman valitsijalla – päätöksentekijällä – on käytössään useita eri tietoja hanke-ehdokkaista. Tiedot on saatu yhdistämällä mitatut kuntotiedot tien käyttäjämääriin sekä tien rakennetta koskeviin perustietoihin. Kunkin hanke-ehdokkaan kohdalla käytössä on nykyisin seuraavat tiedot (kriteerit):

### **Peittoprosentti (%)**

Tiehankeiden arviointia varten lasketaan tien kuntotietoihin sekä käyttäjämääriin perustuvat kantavuuden, vaurioiden, tasaisuuden sekä urasyvyyden peittoprosentit, jotka kuvaavat kyseisellä tavalla vaurioituneen tien osuutta hanke-ehdokkaasta. Edellisten osuuksien unionina saadaan peittoprosentti, joka kuvaa vaurioituneen tien osuutta hanke-ehdokkaan pituudesta.

### **Elinkaarikustannukset (t€/km/vuosi)**

Tien elinkaarikustannuksiin lasketaan pitkällä tarkastelujaksolla tien korjauksiin ja parannuksiin liittyvät investointi- ja ylläpitokustannukset, tien käyttäjien kustannukset sekä tien rakenteiden jäännösarvo tarkastelujakson lopulla. (Petäjä & Spoof 2001).

### **Kestoikä (vuotta)**

Kestoikä on korjaus- tai parannustoimien odotettavissa oleva kestoikä kyseiselle hankkeelle. Ennusteen pohjana käytetään urasyvyyden sekä vauriosumman kehitysnopeutta.

### **Ajokustannussäästö (mitaton suure)**

Ajokustannukset muodostuvat ajoneuvo-, aika-, onnettomuus- sekä ympäristökustannuksista (Tiehallinto 2001). Tässä yhteydessä ajokustannussäästö kuvaa hankkeen toteuttamisen aiheuttamia säästöjä tien käyttäjille ensimmäisen vuoden aikana.

### **Toimenpidekustannus (t€)**

Toimenpidekustannus on arvio hankkeen ennustetuista toteutuskustannuksista. Päällystysohjelmaa valitessa hankkeiden toteuttamiskustannus toimii rajoittavana tekijänä.



Nykyisin päällystysohjelman valinnassa ei järjestelmällisesti hyödynnetä kaikkia kriteereitä, vaan käytännössä valinta suoritetaan pääasiassa yhden kriteerin, esimerkiksi peittoprosentin, mukaan valitsemalla hankkeita suuruusjärjestyksessä kunnes koko budjetti on käytetty. Usean eri kriteerin samanaikainen huomiointi hankkeita valittaessa voidaan mallintaa monitavoitteisella optimointimallilla.

### 3 Monitavoitteinen optimointi

Monitavoiteoptimointi ja yleisemmin monitavoitteinen päätöksenteko on perinteinen operaatiotutkimuksen ongelmakenttä. Aihetta on tutkittu 1950-luvulta lähtien (Charnes *et. al.*, 1955), ja voimakasta kehitystä tapahtui 1970- ja 1980-luvuilla (esim. Keeney ja Raiffa, 1976; Zionts ja Wallenius, 1976; Saaty, 1980; Weber, 1987; French, 1988; Korhonen ja Wallenius, 1988; Arbel, 1989). Hyviä koosteteoksia 1970- ja 1980-lukujen monitavoiteoptimoinnin kehityksestä ovat mm. Steuer (1986) ja Wallenius (1991). Alan tutkimus on jatkunut voimakkaana myös 1990- ja 2000-luvuilla, keskittyen etenkin yksittäisen kiinteän vaihtoehdon valintaa tukevien käyttäjäystävällisten uusien menetelmien kehittämiseen (Salo ja Hämäläinen, 1992; 1995; 2001; 2003; Lahdelma *et. al.*, 1998; Hammond *et. al.*, 1999).

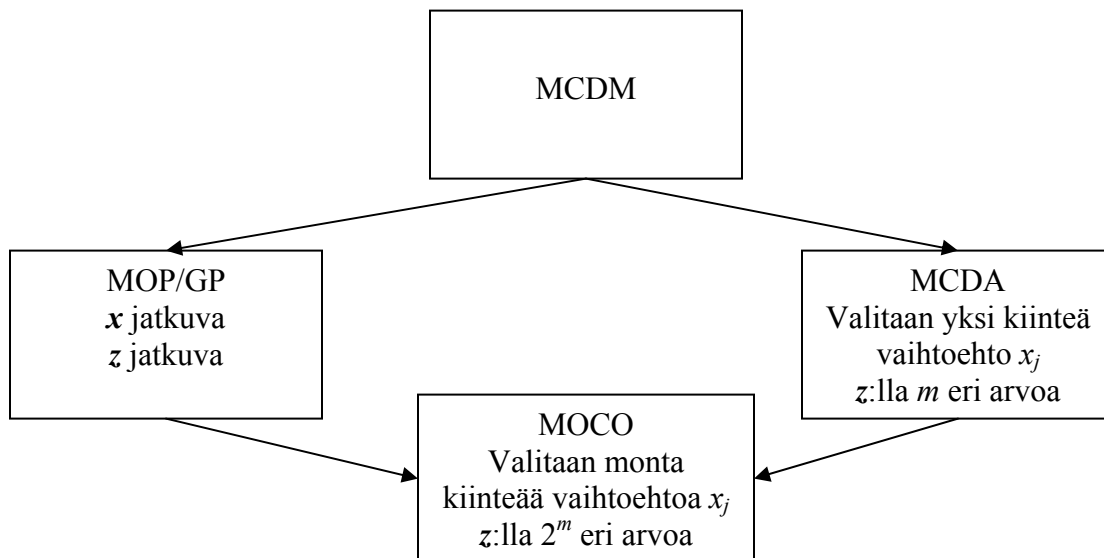
Monitavoiteoptimoinnissa pyritään nimensä mukaisesti etsimään ratkaisu, jolla optimoidaan useampia kohdefunktioita (tavoitteita) samanaikaisesti. Yleinen monitavoiteoptimointitehtävän formulaatio on:

$$\begin{aligned} & \max \{f_1(\mathbf{x}) = z_1\} \\ & \max \{f_2(\mathbf{x}) = z_2\} \\ & \vdots \\ & \max \{f_n(\mathbf{x}) = z_n\} \\ & \text{s.t } \mathbf{x} \in X \end{aligned} \tag{1}$$

Yllä olevassa formulaatiossa on maksimoitavana (tarvittaessa  $\min f \Leftrightarrow \max -f$ )  $n$  kpl kohdefunktioita (vrt. ”tavallinen” optimointitehtävä, jossa on yksi kohdefunktio), s.e. päätösvektori  $\mathbf{x}$  kuuluu käypään alueeseen  $X$ . Käypä päätösvektori  $\mathbf{x}$  tuottaa siis  $n$ -ulotteisen attribuuttivektorin  $\mathbf{z}$ , jonka komponentit ovat eri kohdefunktioiden saamat arvot päätösvektorilla  $\mathbf{x}$ . Ei-triviaalissa tapauksessa oletetaan, että tehtävälle ei löydy ratkaisua  $\mathbf{x}^*$ , joka samanaikaisesti maksimoisi kaikki  $i=1, \dots, n$  kohdefunktiota.

Monitavoitteisen päätöksenteon ”Multiple Criteria Decision Making, MCDM” eri haaroissa formulaatioon (1) liittyy erilaisia tulkintoja. Jatkuvassa (diskreetissä) monitavoiteoptimoinnissa (”MOP/GP”) päätösvektori  $\mathbf{x}$  kuuluu jatkuvaan (diskreettiin) epäyhtälörajoituksin kuvattuun käypään alueeseen. Täten myös attribuuttivektori on jatkuva (ei-numeroituvasti diskreetti). ”Multiattribute Decision Analysis, MCDA” -

haarassa (esim. Keeney ja Raiffa, 1976) valitaan joku monikriteerisistä kiinteistä vaihtoehtoista, eli päätösvektori  $x$  on periaatteessa binäärivektori, joka viittaa eri vaihtoehtoihin. Tyypillisessä MCDA-tehtävässä voidaan valita vaihtoehtoista ( $j = 1, \dots, m$ ) vain yksi, eli päätösvektorin komponenteista vain yksi voi kerrallaan saada arvon 1 muiden saadessa arvon 0. MCDA-tehtävässä attribuuttivektorilla on  $m$  mahdollista kiinteää arvoa. Päätösongelmana on valita mieluisin tarjolla oleva attribuuttivektori,  $z_j = (z_{j1}, z_{j2}, \dots, z_{jn})$ . Monitavoitteisessa kombinatorisessa optimoinnissa, ”Multiple Objective Combinatorial Optimization, MOCO” (esim. Ehrgott ja Gandibleux, 2000; Stummer ja Heidenberger, 2003), tehtävänä on valita portfolio kiinteitä attribuuttivektoreita rajallisilla resursseilla. Kukaan tarjolla oleva vaihtoehto kuluttaa vähintään yhtä resurssia, ja ei-triviaalissa tehtävässä oletetaan, että resurssit eivät riitä kaikkien tarjolla olevien vaihtoehtojen valitsemiseen. MOCO-tehtävässä päätösvektori  $x$  on eri vaihtoehtoihin osoittava binäärivektori, jossa siis useampi komponentti voi kerrallaan saada arvon 1.



**Kuva 1: Monitavoitteisen päätöksenteon luokittelua**

Koska ei-triviaaleissa tehtävissä oletetaan, ettei mikään käypä ratkaisu maksimoi samanaikaisesti kaikkia tavoitteita, monitavoiteongelmien keskeinen kysymys on erilaisten attribuuttivektorien vertailu, ”*tradeoff*” eri tavoitteiden välillä. Erilaisiin – keskenään ristiriitaisiin – attribuuttivektoreihin  $z$  liitetään kokonaisarvofunktio  $V$  (josta

kirjallisuudessa käytetään myös nimityksiä hyötyfunktio ja preferenssifunktio), joka yhdistää erilaisten attribuuttien tulemat yhdeksi mittariksi, jonka perusteella attribuuttivektorit, ja siten ne tuottaneet päätösvaihtoehdot, voidaan asettaa mielekkyyssjärjestykseen. Tärkeä kokonaisarvofunktion malli on *additiivinen malli*, jossa vaihtoehdon (attribuuttivektorin) kokonaisarvo esitetään painotettuna summana attribuuttikohtaisista arvoista. Toisaalta useissa, etenkin jatkuvan monitavoiteoptimoinnin ratkaisumenetelmissä oletetaan ainoastaan, että kokonaisarvofunktio on kaikkien kriteerien suhteen monotoninen (eli maksimointitehtävässä kasvava). Näissä interaktiivisissa menetelmissä (kts. esim. Wallenius, 1991) päätöksentekijälle annetaan vertailtavaksi erilaisia attribuuttivektoreita, joiden välillä tehtyjen valintojen perusteella menetelmä ehdottaa uusia ratkaisuja, kunnes tyydyttävä ratkaisu on saavutettu. Interaktiiviset menetelmät vaativat kuitenkin kustomoitua tietokonetukea sekä tiivistä osallistumista päätöksentekijältä. Lisäksi menetelmien konvergenssi etenkin suurten kombinatoristen tehtävien yhteydessä saattaa olla hidasta.

Additiivisessa mallissa oletetaan, että ratkaisun kokonaisarvo on ei-negatiivisilla painokertoimilla painotettu summa attribuuttikohtaisista arvoista, eli

$$V(\mathbf{x}) = \sum_{i=1}^n w_i f_i(\mathbf{x}) = \sum_{i=1}^n w_i z_i . \quad (2)$$

Tyypillisesti additiivisessa mallissa attribuuttikohtaiset arvot skaalataan välille  $[0,1]$  siten, että paras mahdollinen (rajoitusten puitteissa käypä) suoritustaso saa normeeratun arvon 1 ja huonoin mahdollinen taso saa arvon 0. Normeeratussa mallissa kriteerikohtainen painokerroin  $w_i$  kuvaa sitä kokonaisarvossa tapahtuvaa muutosta, joka liittyy tavoitteen  $i$  siirtymiseen huonoimmalta mahdolliselta tasoltaan parhaimmalle mahdolliselle tasolle.

Koska additiivinen arvofunktio on affineja positiivisia muunnoksia vaille yksikäsitteinen, painot voidaan normeerata halutulla tavalla. Tyypillisesti painot normeerataan siten, että painot kuuluvat alueeseen  $S_0 = \{ \mathbf{w} \in R^n \mid \sum_{i=1}^n w_i = 1, w_i \geq 0, i = 1, \dots, n \}$ .

Jos kyseessä on MOCO-tehtävä, jossa syntyvän portfolion attribuuttikohtaiseksi arvoksi oletetaan portfolioon kuuluvien projektien attribuuttikohtaisten arvojen summa, saadaan additiivisen kokonaisarvon maksimointitehtävä muotoon

$$\max_{\mathbf{x} \in X} V(\mathbf{x}) = \mathbf{x} \mathbf{Q} \mathbf{w}^T, \quad (3)$$

missä  $\mathbf{x}$  on  $1 \times m$  kokoinen binäärivektori,  $\mathbf{Q}$  on  $m \times n$  kokoinen matriisi, joka sisältää kaikkien  $m$  projektin arvot kaikkien  $n$  attribuutin suhteen, ja  $\mathbf{w}^T$  on  $n \times 1$  kokoinen painovektori. Tähän työhön rakennettavan mallin skaalaus, ja siten mallin lopullinen muoto esitellään kappaleessa 4.

Additiivisesta arvofunktiosta on huomioitava, että se olettaa eri kriteerit (tavoitteet) keskenään *preferenssiriippumattomiksi*, eli vaihtoehtojen keskinäinen mielekkyys tietyn kriteerin suhteen ei saa riippua muiden kriteerien arvoista (kts. esim. French, 1988).

Jos päätöksentekijä kykenee tarkasti määrittelemään eri tavoitteiden väliset keskinäiset tärkeydet, eli asettamaan tehtävään tarkat painokertoimet, supistuu monitavoitetehtävä tavalliseksi yhden kohdefunktion optimointitehtäväksi. Tyypillisesti monitavoitteisen päätöksenteon tukijärjestelmissä pyritään kuitenkin pääsemään ratkaisuun tukeutuen *epätäydelliseen preferenssi-informaatioon* (esimerkiksi kriteerien tärkeysjärjestys).

Kun tehtävän (3) painokertoimia ei tunneta tarkasti, vaan niiden tiedetään kuuluvan tiettyyn käypään alueeseen  $S$ , ei tehtävän ratkaisu enää ole yksikäsitteinen vaan muuttuu painovektorin  $\mathbf{w}$  funktiona. Tällöin tehtävästä voidaan erottaa *tehokkaita* (käytetään myös nimitystä Pareto-optimaalinen) ratkaisuja, joita voidaan tarjota päätöksentekijälle järkevinä päätösvaihtoehtoina. Tehokkaaseen ratkaisuun  $\mathbf{x}^*$  liittyvä attribuuttivektori  $\mathbf{z}^*$  on *ei-dominoitu*, kun ei ole olemassa toista sellaista käypää ratkaisua, jonka attribuuttivektori saavuttaa kaikkien attribuuttien suhteen vähintään saman arvon kuin  $\mathbf{z}^*$  ja ainakin yhden attribuutin suhteen korkeamman arvon. Päätöksentekijän ei missään tapauksessa kannata valita ratkaisua, joka ei ole tehokas, koska vertailukohtaksi on aina löydettävissä ratkaisu, joka saa korkeamman arvon jokaisen attribuutin suhteen ja on siis kaikilla monotonisilla arvofunktiolla parempi.

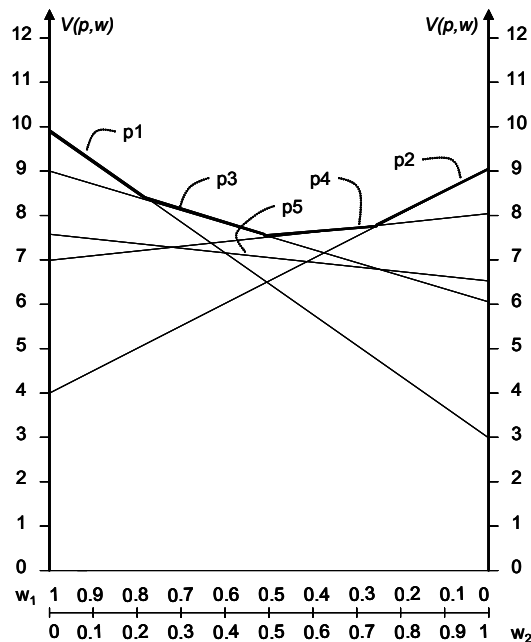
Additiivisille arvofunktiolle voidaan tehokkaiden ratkaisujen joukosta etsiä myös *potentiaalisesti optimaalisten* ratkaisujen joukko. Ratkaisu  $\mathbf{x}^*$  ja siihen liittyvä

kriteerivektori  $z^*$  on potentiaalisesti optimaalinen, jos on olemassa sellainen käypä painovektori  $w$ , jolla  $x^*$  on maksimointitehtävän (3) ratkaisu. Kääntäen, valittaessa mikä tahansa käypä painovektori  $w \in S$  saadaan kyseisellä painolla additiivisen arvofunktion maksimoivaksi ratkaisuksi joku potentiaalisesti optimaalisista ratkaisuista. Kaikki potentiaalisesti optimaaliset ratkaisut löydetään periaatteessa ratkaisemalla maksimointitehtävä (3) kaikilla käyvillä painovektoreilla. Lisäksi potentiaalisesti optimaalisten ratkaisujen joukko on tehokkaiden ratkaisujen osajoukko.

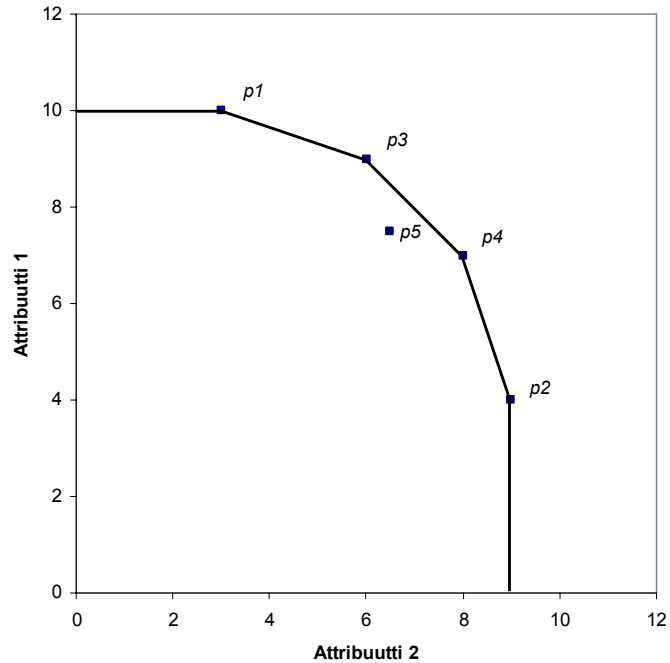
Ei-dominoituja ja potentiaalisesti optimaalisia attribuuttivektoreita voidaan kahden kriteerin tapauksessa havainnollistaa graafisesti. Taulukossa 1 esitetään viisi kiinteää päätösvaihtoehtoa ( $p^1, \dots, p^5$ ), joiden 2-ulotteiset attribuuttivektorit esitetään kuvassa 2 ja additiiviset kokonaisarvot painovektorin funktiona ( $V(p, w)$ ) kuvassa 3.

**Taulukko 1: Kuvien 2-5 vaihtoehdot**

	$p^1$	$p^2$	$p^3$	$p^4$	$p^5$
$z_1$	10	4	9	7	7.5
$z_2$	3	9	6	8	6.5



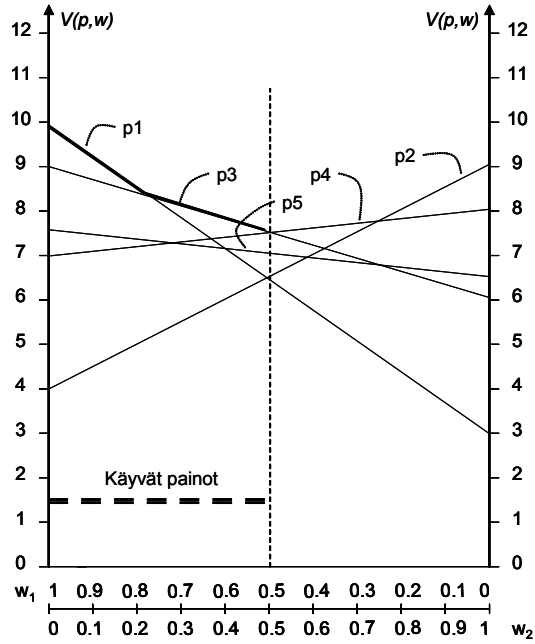
**Kuva 2: Taulukon 1 vaihtoehtojen additiivinen arvo**



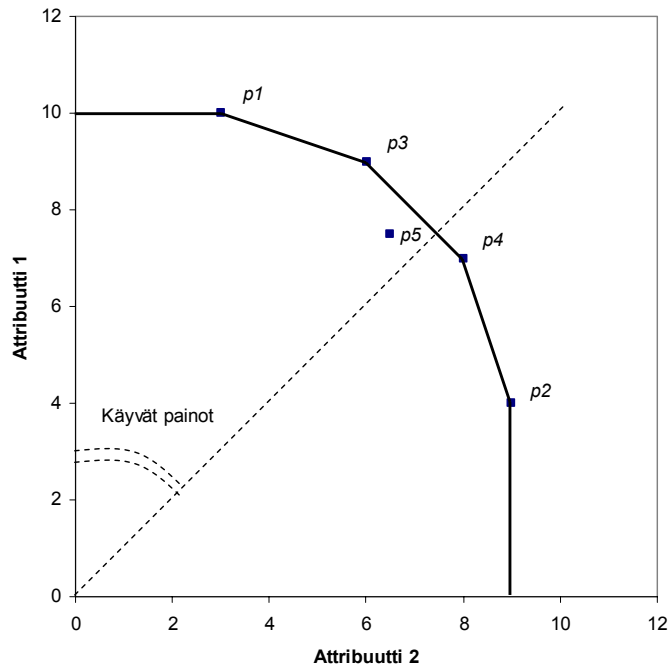
**Kuva 3: Taulukon 1 vaihtoehtojen attribuuttivektorien sijainti**

Kuvat 2 ja 3 esittävät siis samaa tilannetta erilaisista näkökulmista. Molemmista nähdään, että vaihtoehdot  $p^1$ , ...,  $p^4$  ovat potentiaalisesti optimaalisia, eli on olemassa käypä painovektori, jolla ko. vaihtoehdot saavat suurimman additiivisen kokonaisarvon. Myös vaihtoehto  $p^5$  on tehokas, mutta se ei maksimoi additiivista kokonaisarvofunktiota millään käyvällä painovektorilla.

Kuvissa 4 ja 5 esitetään tilanteen muuttuminen, kun tehtävän käyvät painokertoimet rajataan siten, että vaaditaan ehto  $w_1 > w_2$  (eli intuitiivisesti ilmaistaan kriteerin 1 olevan tärkeämpi kuin kriteeri 2). Tällä preferenssi-informaatiolla ainoastaan vaihtoehdot  $p^1$  ja  $p^3$  ovat potentiaalisesti optimaalisia. Kaikki sellaiset painovektorit, joilla vaihtoehdot  $p^1$  ja  $p^3$  olivat aiemmin optimaalisia, on nyt rajattu käyvän painoalueen ulkopuolelle.



**Kuva 4 : Taulukon 1 vaihtoehtojen additiivinen arvo preferenssi-informaatiolla  $w_1 > w_2$**



**Kuva 5 : Taulukon 1 vaihtoehtojen attribuuttivektorien sijainti preferenssi-informaatiolla  $w_1 > w_2$**



Koska MOCO-tehtävän kaikki ratkaisut ovat binäärivektoreita (portfolioita)  $\mathbf{x} = (x_1, \dots, x_m)$ ,  $x_j \in \{0,1\} \forall j$ , voidaan potentiaalisesti optimaalisten ratkaisujen joukosta johtaa myös yksittäisiä  $x_j$ :tä (hankkeita) koskevia analyyseja.

Systeemianalyysin laboratoriossa kehitetty *Ydinluku*-indikaattori kertoo, kuinka suuressa osassa tehtävän potentiaalisesti optimaalisista portfolioista tietty hanke on valittuna. Erityisesti, jos hanke saavuttaa ydinluvun 1, eli hanke kuuluu kaikkiin potentiaalisesti optimaalisiin portfolioihin, tiedetään kyseisen hankkeen kuuluvan varmasti tehtävän optimaaliseen portfolioon, riippumatta siitä mitä käyvistä painovektoreista käytetään. Vastaavasti, jos ydinluku on 0, voidaan päätellä, ettei kyseinen hanke missään tapauksessa kuulu optimaaliseen portfolioon. Hankkeet, joiden ydinluku on välillä (0,1), kuuluvat optimaaliseen ratkaisuun vain tietyssä osajoukossa käypää painoaluetta.

Ydinluku-analyysin perusteella voidaan lopulliseen portfolioon valita turvallisesti ne hankkeet, joiden ydinluku on 1. Vastaavasti 0-arvoiset hankkeet voidaan perustellusti hylätä. Lopuille hankkeille voidaan tehdä tarkempaa jatkoanalyysia ja/tai pyrkiä pienentämään vaihtoehtoisten ratkaisujen joukkoa esimerkiksi painoinformaatiota tarkentamalla tai attribuuttikohtaisia vaatimustasoja kiristämällä.

## 4 Mallin rakentaminen

Mallinamme päätöstilannetta, jossa on tarkoitus muodostaa optimaalinen päällystysohjelma yhdelle vuodelle. Päällystysohjelma valitaan hanke-ehdokasjoukosta rajallisella budjetilla. Mallin käyttämistä useamman vuoden päällystysohjelman valintaan pohditaan raportin lopussa.

Olkoon hanke-ehdokkaita  $m$  kappaletta, kriteerejä  $n$  kappaletta ja kokonaisbudjetti  $b$ . Hanke-ehdokkaan  $j$  suoritustaso kriteerin  $i$  suhteen on  $S_{ij}$  ja kustannus  $c_j$ . Olkoon  $x_j$  binääriarvoinen päätösmuuttuja, joka saa arvon 1 hanke-ehdokkaan  $j$ :n kuuluessa päällystysohjelmaan ja arvon 0, kun hanke-ehdokkaan  $j$  ei kuulu päällystysohjelmaan. Tällöin käyvien päällystysohjelmien joukon muodostavat ne päällystysohjelmat, joihin kuuluvien hanke-ehdokkaiden kustannusten summa ei ylitä budjettia. Tämä voidaan lausua epäyhtälörajoituksena

$$\sum_{j=1}^m x_j c_j \leq b.$$

Käytämme tässä additiivista arvofunktiota yhdistämään hanke-ehdokkaan kriteerikohtaiset suoritusarvot kokonaisarvoksi. Tällöin hanke-ehdokkaan  $j$  kokonaisarvo on

$$\sum_{i=1}^n w_i S_{ij},$$

missä kriteerin  $i$  paino  $w_i$  kuvaa kriteerin  $i$  suhteellista tärkeyttä. Eri kriteerit ovat yleisesti yhteismitattomia. Näin ollen  $w_i$ :n absoluuttinen arvo ei kerro kriteerin  $i$  suhteellisesta tärkeydestä, vaan se kuvaa, kuinka paljon yksi yksikkö kriteeriä  $i$  lisää kokonaisarvoa. Jotta painojen absoluuttisia arvoja voitaisiin keskenään vertailla, skaalataan jokainen kriteeri siten, että suurin mahdollinen suoritusarvo kyseisen kriteerin suhteen on 1 ja huonoin 0. Kriteerin  $i$  suhteen suurin mahdollinen suoritusarvo ilman skaalausta  $S_i^{\max}$  saadaan ratkaisuna lineaariselle kokonaislukutehtävälle

$$S_i^{\max} = \max_x x_j S_{ij}$$

$$\sum_{j=1}^m x_j c_j \leq b,$$

$$x_j = \{0,1\},$$

joka suoritetaan kaikille kriteereille  $i=1\dots n$ . Nyt hanke-ehdokkaan  $j$  skaalattu suoritustaso kriteerin  $i$  suhteen on

$$s_{ij} = \frac{S_{ij}}{S_i^{\max}}.$$

Tällä skaalauksella painot voidaan valita siten, että ne ovat positiivisia ja summautuvat ykköseksi (eli  $w = [w_1 \dots w_n]^T \in \mathcal{S}_0 = \{w \mid \sum_{i=1}^n w_i = 1, w_i \geq 0\}$ ). Nyt kriteerin  $i$  paino kuvaa lisäarvoa, joka syntyy, kun päällystysohjelman suoritustaso kyseisen kriteerin suhteen siirtyy nollasta parhaimpaan ( $0 \rightarrow S_i^{\max}$ ).

Lopullinen malli on muotoa

$$\max_x \sum_{j=1}^m (x_j \sum_{i=1}^n w_i s_{ij})$$

$$\sum_{j=1}^m x_j c_j \leq b$$

$$x_j = \{0,1\}.$$

Tarkoilla painokertoimilla tehtävä vastaa yhden kriteerin maksimointia, eli kyseessä on lineaarinen kokonaislukutehtävä. Tällainen tehtävä on ratkaistavissa useimmilla LP-paketeilla. Tehtävä on luonteeltaan NP-täydellinen, eli sen ratkaiseminen parhaallakin algoritmilla vie pahimmassa tapauksessa ajan, joka on verrannollinen lukuun  $e^n$ . Aika siis kasvaa eksponentiaalisesti ongelman koon kasvaessa.

Voidaan kuitenkin olettaa, ettei päätöksentekijä kykene antamaan tarkkoja painoja tai ainakin haluaa herkkyysoanalyysimielessä tietää, miten optimaalinen ratkaisu muuttuu

painojen muuttuessa. Tällöin halutaan ratkaista tehtävä kaikilla painoilla, jotka kuuluvat päätöksentekijän antamasta preferenssi-informaatiosta johdettuun käypään painoalueeseen  $S$ . Näin ollen ratkaisuksi saadaan joukko tehokkaita ratkaisuja. Systeemianalyysin laboratoriossa on viimeaikoina kehitetty tarkkoja algoritmeja tällaisen ongelman ratkaisemiseen (Mild, 2004; Liesiö, 2004; Salminen, 2004). Näillä algoritmeilla laskenta-ajat pysyvät siedettävänä vain, jos hanke-ehdokkaiden määrä  $m$  on alle 50. Mainittakoon, että tapauksessa, jossa koko painoalue on käypä (eli  $S = S_0$ ), tehtävä vastaa monitavoitteista Knapsack-ongelmaa, johon on kehitetty tarkkoja algoritmeja (esim. Erlebach ja Keller, 2002). Myöskään nämä algoritmit eivät sovellu projektin kannalta kiinnostavan kokoiisiin tehtäviin.

Koska tarkkoja ratkaisuja ei voida laskea riittävän nopeasti, on tyydyttävä approksimatiivisiin ratkaisuihin. Monitavoitteiselle Knapsack-ongelmalle on kehitetty geneettisiin algoritmeihin perustuvia heuristiikkoja (esim. Zitzler ja Thiele, 1999). Koska kyseessä on kuitenkin puhtaat heuristiikat, ei ratkaisujen tarkkuudelle pystytä johtamaan analyttisiä rajoja. Pahimmassa tapauksessa yksikään heuristiikan ratkaisusta ei siis ole tehokas, vaan eksakteilla menetelmillä olisi löydettävissä kaikkien kriteerien suhteen parempi ratkaisu.

Erlebach ja Keller (2002) todistavat, että monitavoitteiselle Knapsack-ongelmalle on olemassa algoritmi, joka antaa approksimaation ratkaisuille polynomisessa ajassa. Erona heuristiikkoihin on, että tämän approksimaation jokainen ratkaisu on korkeintaan tietyllä etäisyydellä  $\varepsilon$  todellisesta ratkaisusta. Parametria  $\varepsilon$  säätämällä voidaan valita sopiva kompromissi laskenta-ajan ja approksimaation tarkkuuden välillä. Erlebach ja Keller eivät kuitenkaan esitä tarkkaa algoritmia tai empiirisiä kokeita, joiden perusteella voitaisiin päätellä, minkä kokoiisiin ongelmiin polynomiainen approksimaatio soveltuu.

Edellisten havaintojen perusteella päädyimme käyttämään Systeemianalyysin laboratoriossa kehitettyä hilamenetelmää, jossa käypä painoavaruus  $S$  kuvataan systemaattisella tasavälisellä hilapisteistöllä. Hilapisteistö muodostetaan diskreetin koordinaatiston tapaan siten, että jokainen kriteeriakseli jaetaan yhtä moneen osaväliin ja poimitaan täten muodostuvasta hilasta käypään painoalueeseen  $S$  kuuluvat pisteet

mukaan ko. aluetta edustavaan hilaan. Systeemianalyysin laboratoriossa on kehitetty algoritmi, joka tuottaa hilapisteistön suoraan käyväälle kriteeripainoalueelle (huomioiden ei-negatiivisuus- ja summautumisvaatimukset). Liitteessä 1 esitetään esimerkkinä hilapisteistö, jossa kriteerien lukumäärä on 3, ja kukin akseli on jaettu 5 osaan. Jokaisen 21 pisteen kaikki kolme komponenttia ovat siis  $1/5 = 0,2:n$  monikertoja ja jokaisen pisteen komponenttien summa on 1.

Jokaisessa hilapisteessä ratkaistaan edellä esitetty lineaarinen kokonaislukutehtävä ja saadaan yksi tehokas ratkaisu eli päällystysohjelma. Samaan ratkaisuun voidaan luonnollisesti päätyä useammassa painopisteessä, mutta saatujen (toisistaan eroavien) ratkaisujen ylärajana on hilapisteiden lukumäärä. Approksimaatio ei ole erityisen tehokas laskenta-ajan kannalta, mutta sillä on muita hyviä ominaisuuksia: jokainen saatu ratkaisu tiedetään potentiaalisesti optimaaliseksi. Löytämättä jääneet päällystysohjelmat sijaitsevat kriteeriavaruuden osissa, joita yksikään potentiaalisesti optimaalinen ohjelma ei dominoi. Lisäksi kun tiedetään aika, joka kuluu lineaarisen kokonaislukutehtävän ratkaisemiseen ensimmäisessä painopisteessä, voidaan esittää erittäin tarkka ennuste kokonaisratkaisuaikalle kertomalla kyseinen aika painopisteiden kokonaismäärällä. Taulukossa 2 on esitetty hilapisteiden lukumäärä eri kriteerien määrillä, kun jokaisen kriteeripainon vaihteluväli on jaettu  $k$  :hon osaan.

Taulukossa 3 on puolestaan arvio yhden lineaarisen kokonaislukutehtävän ratkaisemiseen kuluvasta ajasta hanke-ehdokkaiden määrän  $m$  funktiona. Arviot perustuvat Xpress-MP-ohjelmistolla (prosessori Pentium III 800Mhz) tehtyihin koeajoihin, jossa datana on käytetty projektin ensimmäistä testiaineistoa. Todellinen aika on erittäin herkkä ratkaistavan tehtävän luonteelle: laskenta-aikaa pidentävät esim. vahvasti hanke-ehdokkaan kustannuksen kanssa korreloivat kriteerikohtaiset suoritustasot sekä suuri budjetti suhteessa kaikkien hanke-ehdokkaiden kokonaiskustannukseen.

**Taulukko 2 Hilapisteiden lukumäärä**

k   n	2	3	4	5	6	7	8
5	6	21	56	126	252	462	792
10	11	66	286	1001	3003	8008	19448
20	21	231	1771	10626	53130	230230	888030
40	41	861	12341	135751	1221759	9366819	62891499

**Taulukko 3 Yhden lineaarisen kokonaislukutehtävän ratkaisemiseen kuluva aika**

m	250	650	1000	1600
Aika	0.4s	4s	10s	20s

Hilapisteiden tiheyden valinnassa tehdään siis kompromissi tarkkuuden ja laskenta-ajan välillä. Sopivaa tarkkuutta arvioitaessa on kaksi lähestymistapaa. Jos tarkoitus on löytää mahdollisimman suuri osa tehokkaista päällystysohjelmista, on hilapisteiden tiheyttä nostettava, kunnes pisteistöstä löydettyjen erilaisten päällystysohjelmien lukumäärä on selvästi hilapisteiden määrää pienempi, eli useammassa pisteessä on löydetty sama päällystysohjelma. Mahdollisimman monen tehokkaan päällystysohjelman löytäminen ei kuitenkaan ole välttämättä tarkoituksenmukaista, vaan riittää, että päätöksentekijällä on riittävän tiheässä tehokkaita päällystysohjelmia joista valita. Tällöin on siis syytä kiinnittää huomiota siihen, mikä on kahden tehokkaan päällystysohjelman etäisyys tietyn kriteerin suhteen. Jos esimerkiksi yhtenä kriteerinä on päällystysohjelmaan sisältyvien hankkeiden yhteispituus, halutaan kahden tämän kriteerin suhteen vierekkäisen päällystysohjelman etäisyyden olevan korkeintaan 10 km.

## 5 Testidatan mallintaminen

### 5.1 Koesuunnitelma

Toimeksiantajalta saatiin kaksi data-aineistoa, joihin sovellamme edellisessä kappaleessa esitettyä mallia. Ensimmäisessä testidatassa on 626 hanke-ehdokasta ja budjetti 20 miljoonaa euroa. Toisessa testidatassa on puolestaan 222 hanke-ehdokasta ja budjetti 16,3 miljoonaa euroa. Toisen testidatan pohjalta on myös tehty todellinen päätös (valittu päällystysohjelma).

Testiajot on tehty kolmella kriteerillä, jotka toimeksiantaja totesi kaikkein olennaisimmiksi. Rajoitimme tarkastelun vain kolmeen kriteeriin, koska tällöin laskentaajat pysyvät hallinnassa ja toisaalta tulokset ovat helpommin visualisoitavissa. Projektin puitteissa ei ollut mahdollista implementoida työkalua, jonka avulla mallin ratkaisu ja toisaalta tulosten visualisointi voitaisiin automatisoida. Näin ollen, malli on ratkaistu Xpress-MP-ohjelmistolla ja visualisoinneissa on käytetty Matlab-ohjelmistoa, mikä vaati huomattavasti 'käsityötä'.

Koesuunnitelma on esitelty taulukossa 4. Tehokkaat päällystysohjelmat ratkaistiin molemmille testidatoille käyttäen kahta erilaista preferenssi-informaatiota: i) kriteerien suhteellisesta tärkeydestä ei ole informaatiota ja ii) kriteerit asetetaan tärkeysjärjestykseen, mutta ei tiedetä, kuinka paljon yksi kriteeri on toista tärkeämpi. Kohdassa ii) kriteerien tärkeysjärjestys asetettiin asiakkaan ilmoittamaan tärkeysjärjestykseen:

- 1. Päällystysohjelmaan kuuluvien hankkeiden yhteenlaskettu viallisen tien pituus "peitto%\*pituus" (yksikkönä metriä)*
- 2. Päällystysohjelmaan kuuluvien hankkeiden yhteensä ensimmäisenä vuonna synnyttämä ajokustannussäästö "ajokust\*pituus" (ei mittayksikköä)*
- 3. Päällystysohjelmaan kuuluvien hankkeiden yhteenlaskettu pituudella painotettu kestoikä "kestoikä\*pituus" (yksikkönä vuosimetriä)*

Tärkeydellä tarkoitetaan tässä esimerkiksi, että kriteerin "peitto%\*pituus" nousu nollassa tasolta maksimiarvoon (37000 km) on arvokkaampaa kuin kriteerin "ajokust\*pituus"

nousu nollassa maksimiin ( $9 \cdot 10^8$ ). Mallissa tämä preferenssi-informaatio johtaa käyvän painoalueen rajaamiseen, jolloin käyväksi painoalueeksi jää

$$S = \{w \mid \sum_{i=1}^3 w_i = 1, w_i \geq 0, w_1 \geq w_2, w_2 \geq w_3\}.$$

Sekä tapauksessa i) että ii) preferenssi-informaatio on epätäydellistä (ei ole vain yhtä käypää painovektoria), jolloin tehokkaita ratkaisuja voi olla useita.

**Taulukko 4 Koesuunnitelma**

	<i>Testidata 1</i>	<i>Testidata 2</i>
i) Ei preferenssi informaatiota	Ajo 1	Ajo 3
ii) Kriteerien tärkeysjärjestys tiedetään	Ajo 2	Ajo 4

Toisen testidatan tapauksessa luonnollinen referenssiratkaisu on datan mukana saatu todellinen päätös. Voimme siis verrata todellista tehtyä päätöstä mallilla ratkaistuihin tehokkaisiin päällystysohjelmiin ja tehdä mahdollisista eroista johtopäätöksiä. Ensimmäiseen testidataan ei kuulunut todellista päätöstä, joten referenssipäätöksiksi valitsimme kahdella erilaisella heuristiikalla saadut ratkaisut. Nämä heuristiikat jäljittelevät todellista, tällä hetkellä käytössä olevaa ratkaisumenetelmää.

*Heuristiikka 1:* Järjestetään hanke-ehdokkaat tunnusluvun  $peitto\% \cdot pituus$  mukaiseen alenevaan järjestykseen ja valitaan hankkeita päällystysohjelmaan tässä järjestyksessä, kunnes budjettiin ei enää mahdu yhtään hanketta.

*Heuristiikka 2:* Järjestetään hanke-ehdokkaat tunnusluvun  $peitto\%$  mukaiseen alenevaan järjestykseen ja valitaan hankkeita päällystysohjelmaan tässä järjestyksessä, kunnes budjettiin ei enää mahdu yhtään hanketta.

Heuristiikat pyrkivät siis yhden kriteerin suhteen maksimaaliseen päällystysohjelmaan. Heuristiikat eivät välttämättä tuota tämänkään kriteerin suhteen optimaalista tulosta, mutta ovat toisaalta helposti toteutettavissa tavallisella taulukkolaskentaohjelmistolla.

Referenssiratkaisujen avulla ratkaistiin myös minimikustannusohjelma eli päällystysohjelma, joka on vähintään yhtä hyvä jokaisen kriteerin suhteen kuin



referenssinä toimiva päällystysohjelma, mutta on kustannuksiltaan minimaalinen. Minimikustannusohjelma saadaan ratkaisemalla seuraava lineaarinen kokonaislukuoptimointitehtävä:

$$\begin{aligned} \min_x \quad & \sum_{j=1}^m c_j x_j \\ & \sum_{j=1}^m x_j s_{1j} \geq S_1^{ref} \\ & \sum_{j=1}^m x_j s_{2j} \geq S_2^{ref} \\ & \sum_{j=1}^m x_j s_{3j} \geq S_3^{ref} \\ & x_j = \{0,1\}. \end{aligned}$$

Tässä  $S_i^{ref}$  on referenssinä toimivan päällystysohjelman (heuristiikan tuottama tai todellinen päätös) suoritustaso kriteerin  $i$  suhteen.

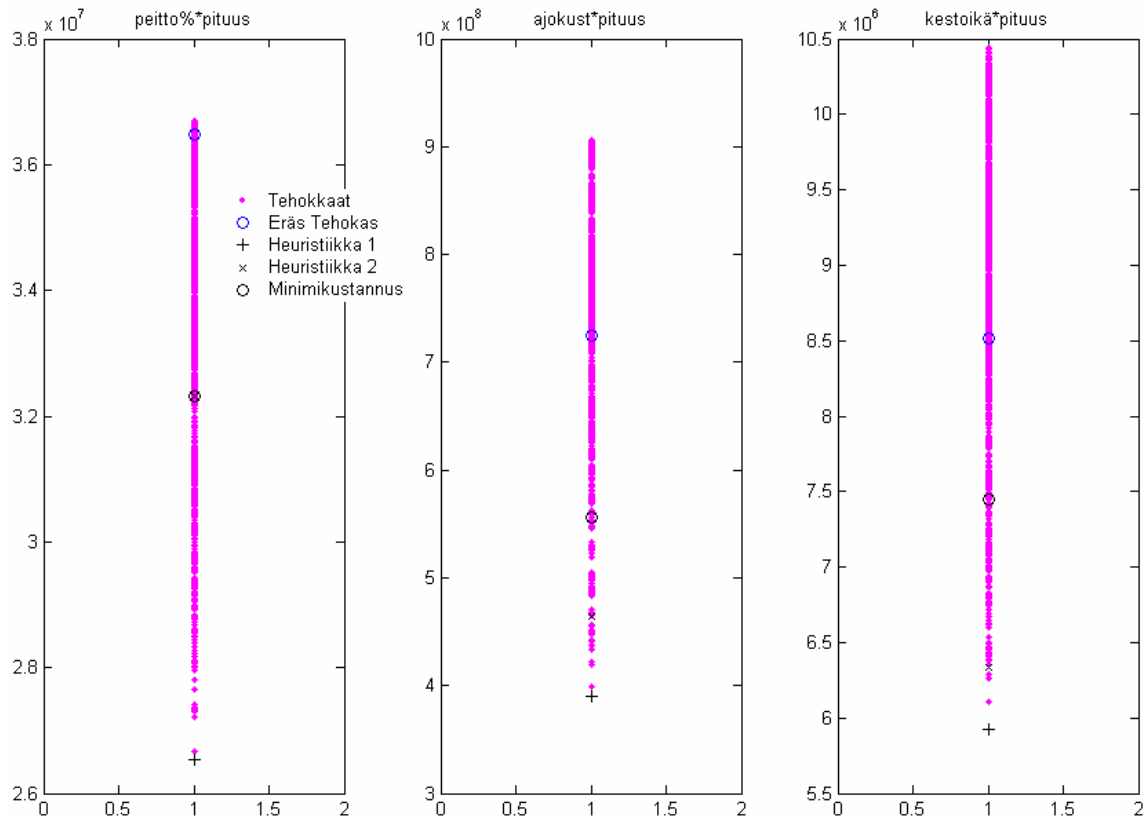
## 5.2 Tulokset

### 5.2.1 Ajo 1

Ensimmäisessä testidatassa oli siis 626 hanke-ehdokasta. Tehokkaat päällystysohjelmat laskettiin käyttäen hilaa, jossa jokaisen kriteeripainon vaihteluväli oli jaettu 50:een osaan. Yhteensä hilapisteitä oli tässä kolmen kriteerin tapauksessa 1326 kappaletta. Keskenään erilaisia tehokkaita päällystysohjelmia löytyi yhteensä 1233 kappaletta. Keskimäärin tehokkaaseen päällystysohjelmaan kuului 340 hanketta. Halvin tehokas päällystysohjelma jätti käyttämättä budjetista vain 2000 euroa, joten käytännössä kaikki tehokkaat ohjelmat käyttivät budjetin kokonaisuudessaan (kokonaisbudjetti 20 miljoonaa euroa).

Tehokkaiden päällystysohjelmien kriteerikohtaiset suoritustasot esitetään kuvassa 6. Jokainen tehokas päällystysohjelma tuottaa yhden pisteen jokaiseen kolmeen kriteeri-

sarakkeeseen, eli yksittäistä ohjelmaa edustaa yksi ”pistekolmikko”. Pisteiden sijainti tietyssä sarakkeessa kuvaa kyseisen päällystysohjelman suoritustasoa kyseisen kriteerin suhteen. Esimerkiksi ympyröity päällystysohjelma sisältää yhteensä noin 36500 km päällystettävää tietä (peitto%\*pituus). Koska kyseessä ovat tehokkaat päällystysohjelmat, tästä joukosta ei löydy kahta päällystysohjelmaa, joista toinen saavuttaisi kaikkien kriteerien suhteen paremman tason kuin toinen. Tämä siis tarkoittaa, ettei ole mahdollista valita päällystysohjelmaa, joka olisi jokaisessa kriteeri-sarakkeessa ylimpänä.



**Kuva 6 Ajo 1, Päällystysohjelmien kriteerikohtaiset suoritustasot: kukin tehokaspäällystysohjelma tuottaa yhden pisteen jokaiseen sarakkeeseen. Pisteiden sijainti tietyssä sarakkeessa kuvaa päällystysohjelman sisältämien hankkeiden yhteenlaskettua arvoa kyseisen kriteerin suhteen. Esimerkiksi ympyröity tehokas päällystysohjelma on kaikkien kriteerien suhteen heuristiikkojen ratkaisuja parempi.**

Lisäksi kuvaan 6 on piirretty heuristiikoilla 1 ja 2 ratkaisuna saadut päällystysohjelmat. Selvästi näistä kumpikaan ei ole tehokas, sillä esimerkiksi ympyröity tehokas päällystysohjelma on kaikkien kriteerien suhteen parempi kuin kumpikaan heuristiikoilla

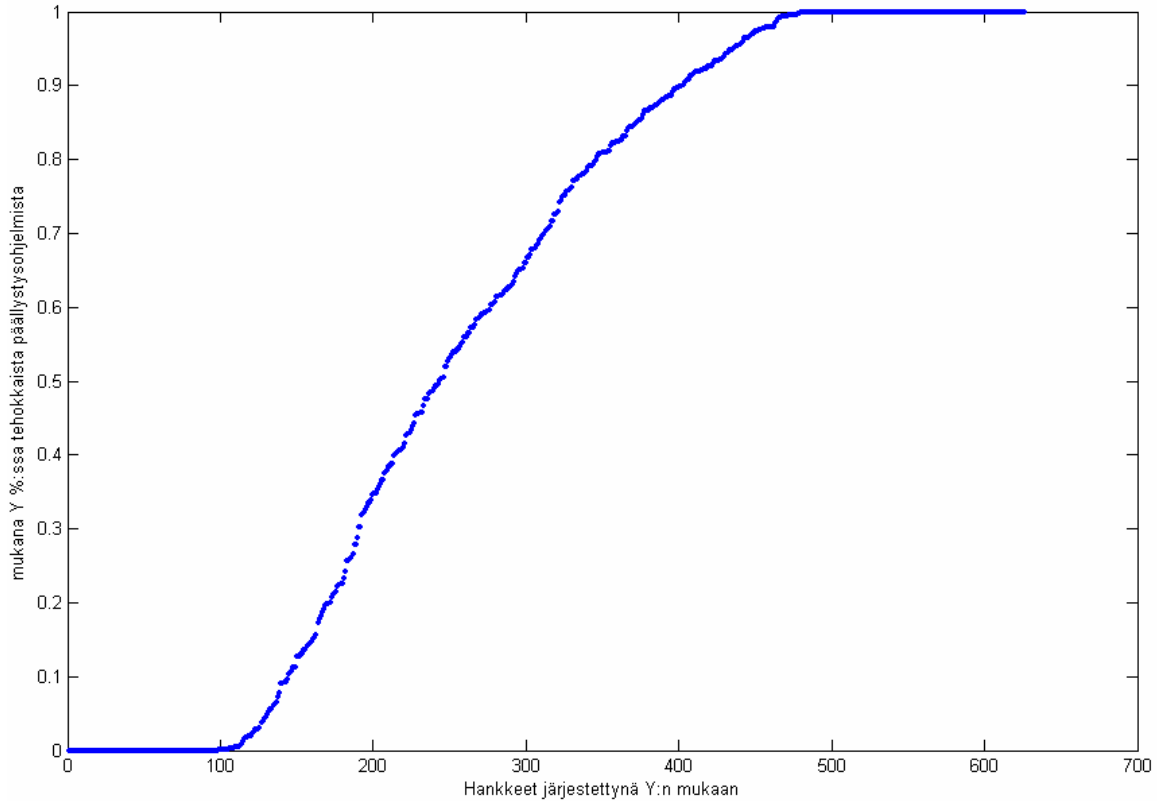
saaduista päällystysohjelmista. Lisäksi heuristiikan 2 antama päällystysohjelma on kaikkien kolmen kriteerin suhteen heuristiikan 1 päällystysohjelmaa parempi. Syy tähän on, että pyrittäessä maksimoimaan päällystysohjelmaan kriteeriä ”peitto%\*pituus”, heuristiikka 2 muistuttaa yleisesti käytettyä hyöty-kustannussuhteeseen perustuvaa optimointiheuristiikkaa, jossa projekteja valitaan portfolioon alenevassa hyöty-kustannussuhteen mukaisessa järjestyksessä, kunnes budjetti on käytetty loppuun. Yhteys johtuu datassa ilmenevästä voimakkaasta korrelaatiosta hankkeen kustannuksen ja sen pituuden välillä. Formaalisti tämä voidaan ilmaista seuraavasti:

$$\frac{\text{hyöty}}{\text{kustannus}} = \frac{\text{peitto\%} \cdot \text{pituus}}{\text{kustannus}} \approx \frac{\text{peitto\%} \cdot \text{pituus}}{\text{pituus}} = \text{peitto\%} .$$

Minimikustannusohjelman kriteerikohtaiset suoritustasot on myös merkitty kuvaan 6. Tämä päällystysohjelma on sellainen, joka minimoii kustannukset ehdolla, että jokaisen kriteerin suhteen suoritustasot ovat vastaavia heuristiikan 2 antaman päällystysohjelman suoritustasoja. Kuvasta nähdään, että rajoitus kriteerin ”peitto%\*pituus” suhteen tulee ensimmäisenä vastaan. Minimikustannus-päällystysohjelman hinta on noin 16,1 miljoonaa euroa, vaikka se on kaikkien kriteerien suhteen vähintään yhtä hyvä kuin 20 miljoonaa maksava heuristiikan 2 tuottama päällystysohjelma. Optimoinnilla saavutetaan siis merkittäviä lisähyötyjä suhteessa heuristiikkoihin.

Kuvassa 7 on kuvattu, mitkä yksittäiset hanke-ehdokkaat kuuluvat tehokkaiisiin päällystysohjelmiin: vaaka-akselilla on hankkeet järjestettynä ydinluvun mukaan ja pysty-akselilla vastaava ydinluku. Esimerkiksi ensimmäiset sata pistettä ovat nollassa, mikä tarkoittaa, etteivät kyseiset hankkeet kuulu yhteenkään (ydinluku = 0) 1233 tehokkaasta päällystysohjelmasta. Toisaalta noin 150 hanketta on saanut ydinluvun arvoksi yhden, eli ne kuuluvat kaikkiin 1233 tehokkaaseen päällystysohjelmaan.

Tehokkaiden päällystysohjelmien joukko kuvaa siis optimaalisia ratkaisuja erilaisilla preferensseillä, eli erilaisilla näkemyksillä siitä, kuinka arvokkaita kriteerit ovat toisiinsa nähden. Koska tässä ajossa laskettiin tehokkaat päällystysohjelmat yli kaikkien preferenssien, voidaan tulosten nojalla todeta, että kyseiset 100 ydinluvun nolla saanutta hanketta voidaan jättää valitsematta.



**Kuva 7 Ajo 1, ydinhankeet: Jokaiselle 626 hankkeelle on laskettu ydinluku, joka kertoo kuinka monessa prosentissa tehokkaista päällystysohjelmista kyseinen hanke on mukana.**

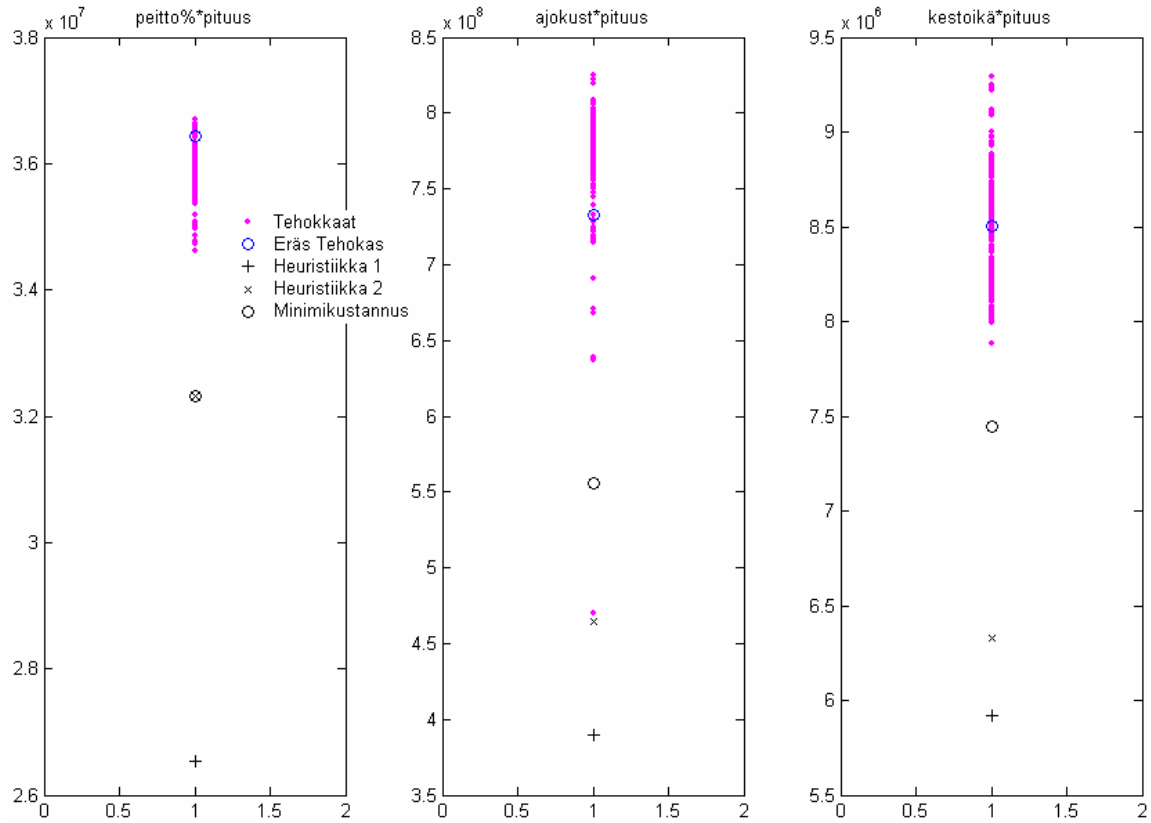
**Hankkeet on järjestetty kuvassa ydinluvun mukaan, joten helposti nähdään että 100 ensimmäistä hanketta eivät kuulu tehokkaaseen päällystysohjemaan missään tapauksessa ja toisaalta noin 150 viimeistä hanketta ovat varmoja valintoja.**

### 5.2.2 Ajo 2

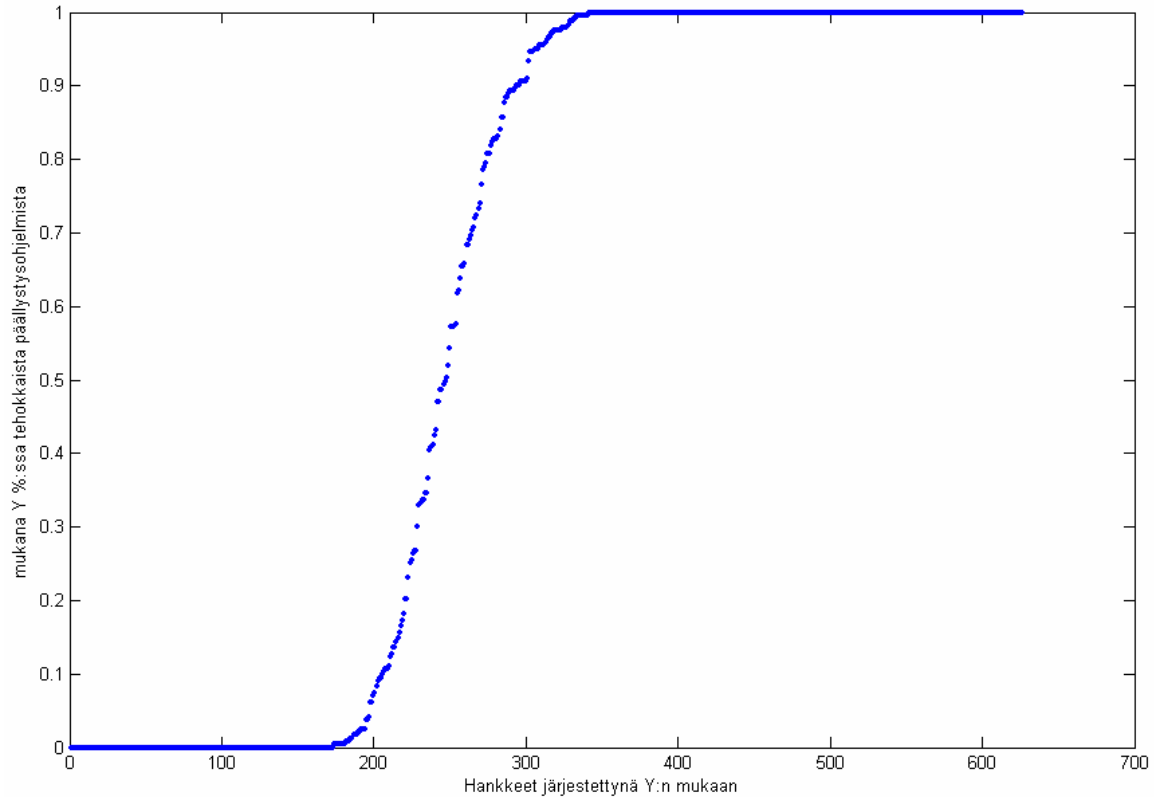
Tässä testiajossa data on sama kuin edellisessä ajossa, mutta kriteereille on annettu tärkeysjärjestys. Tehokkaiden päällystysohjelmien joukko on siis osajoukko edellisen ajon tehokkaista päällystysohjelmista. Hilapisteiden lukumäärä oli 243. Yhteensä löytyi 212 tehokasta päällystysohjelmaa, eli useista pisteistä löytyi sama ratkaisu.

Tulokset on visualisoitu kuvissa 8 ja 9 kuten edellisessä ajossa. Huomataan, että ensimmäisen kriteerin suhteen (joka on siis suhteellisesti tärkein annetun preferenssi-informaation perusteella) tehokkaat päällystysohjelmat sijoittuvat huomattavasti lähemmäksi maksimia kuin edellisessä ajossa. Kuvasta 9 nähdään, että varmoja valintoja on jo lähes 300 kappaletta ja varmasti hylättyjä lähes 200 kappaletta. Annetun

preferenssi-informaation ansiosta tehokkaat päällystysohjelmat ovat nyt lähempänä toisiaan kuin edellisessä ajossa.



**Kuva 8 Ajo 2, Päällystysohjelmien kriteerikohtaiset suoritustasot: Preferenssi-informaation antaminen pienentää tehokkaiden päällystysohjelmien joukkoa. Heuristiikkojen ratkaisut eivät yllä edes samoille vaihteluväleille tehokkaiden päällystysohjelmien kanssa.**



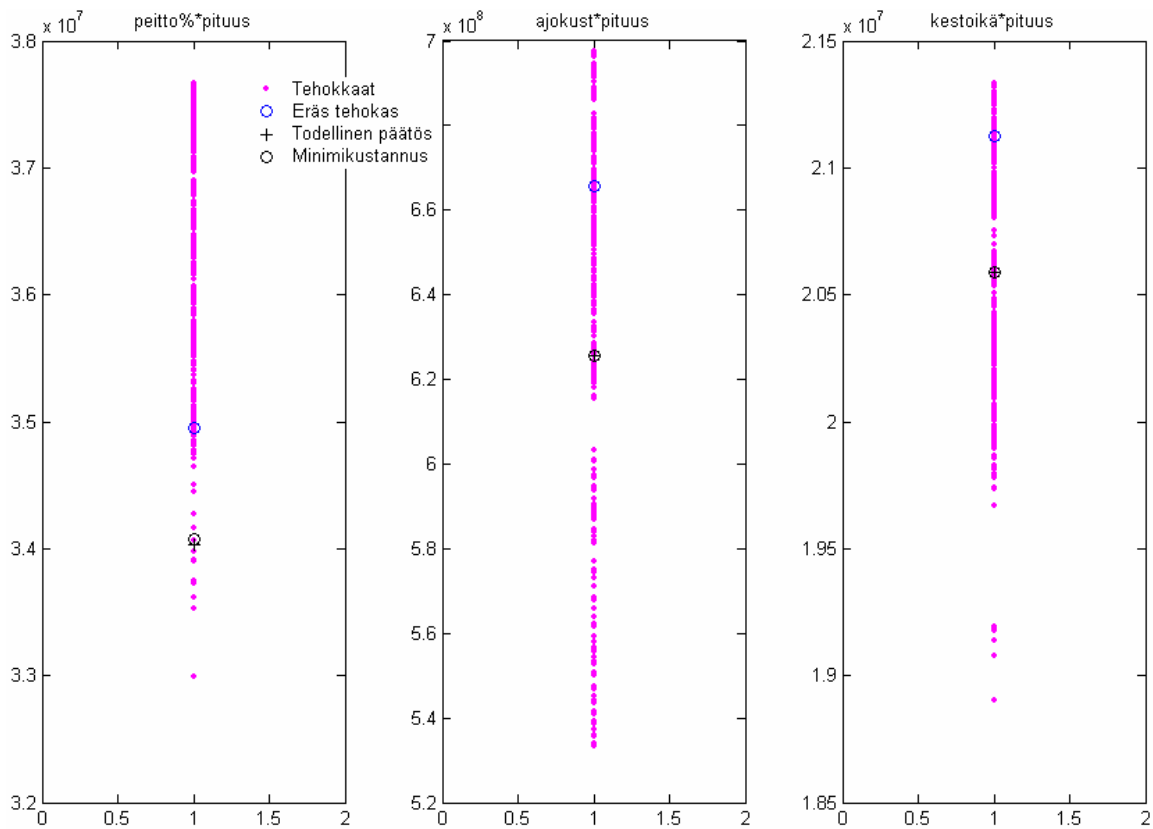
**Kuva 9 Ajo 2, ydinhankkeet: Preferenssi-informaation avulla voidaan jo lähes 200 hanketta jättää varmasti pois päällystysohjelmasta ja toisaalta noin 300 hanketta todeta varmoiksi valinnoiksi.**

### 5.2.3 Ajo 3

Hilapisteiden tiheys oli sama kuin käytettäessä ensimmäistä testidataa ilman preferenssi-informaatiota (yhteensä 1326 hilapistettä). Tehokkaita päällystysohjelmia löytyi yhteensä 397 kappaletta, eli monista painopisteistä löytyi sama ratkaisu, mikä osoittaa hilapisteiden tiheyden olevan riittävä. Tehokkaisiin päällystysohjelmiin kuului keskimäärin 165 hanketta. Tehokkaiden päällystysohjelmien joukossa budjetista jäi suurimmillaan käyttämättä 4000 euroa (kokonaisbudjetti 16,3 miljoonaa euroa).

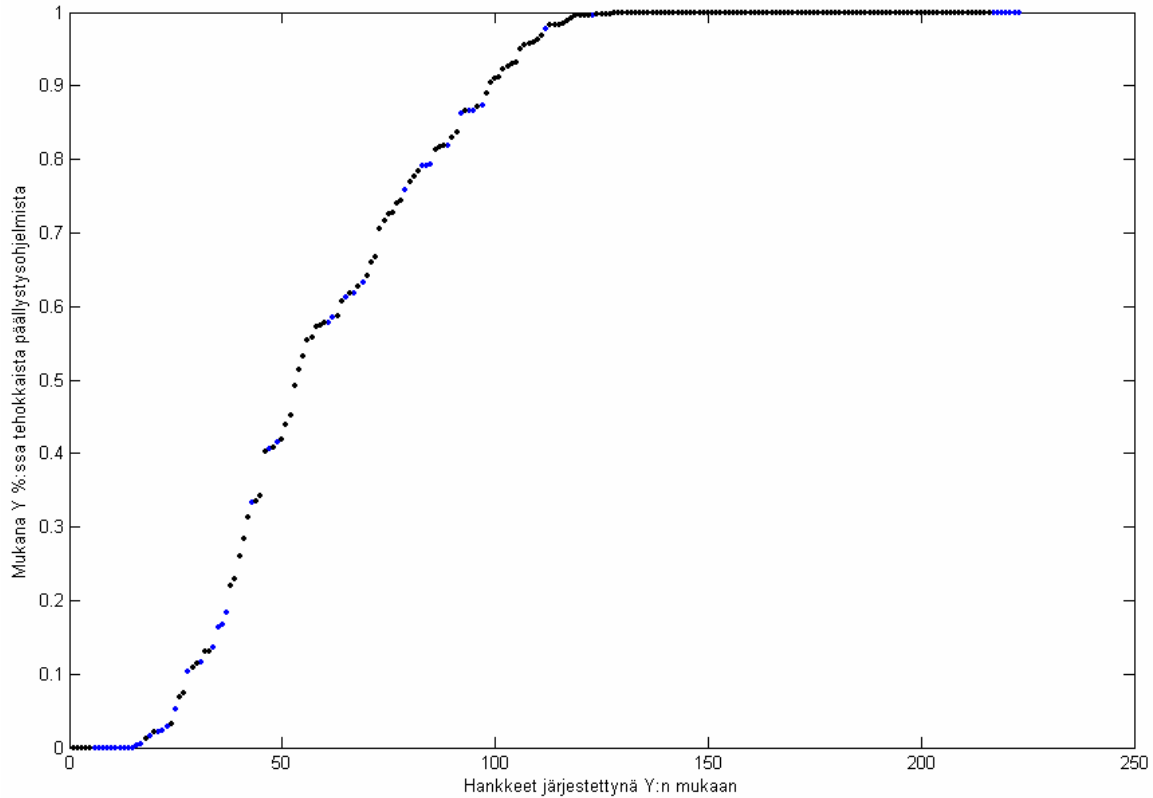
Kuvassa 10 on tehokkaiden päällystysohjelmien kriteerikohtaiset suoritustasot. Selvästi nähdään, että todellinen tehty päätös ei ole tehokas käytettäessä näitä kolmea kriteeriä ilman preferenssi-informaatiota. Esimerkiksi ympyröity tehokas päällystysohjelma on kaikkien kriteerien suhteen parempi. Minimikustannusohjelma on kustannukset minimoiva päällystysohjelma, ehdolla että jokainen suoritustaso on vähintään sama kuin

todellisessa päätöksessä. Minimikustannusohjelman hinta on noin 15 miljoonaa eli yli miljoona euroa halvempi kuin todellinen päätös.



**Kuva 10 Ajo 3, Päälystysohjelmien kriteerikohtaiset suoritustasot: Todellinen valittu päälystysohjelma ei ole tehokas, koska esimerkiksi ympäröity tehokas päälystysohjelma on jokaisen kriteerin suhteen parempi vaikka hinta on sama.**

Kuvassa 11 on esitetty ydinhankeet. Koska hankkeita oli tässä testidatassa vähemmän kun ensimmäisessä, saadaan ilman preferenssi-informaatiotakin huomattava määrä varmoja valintoja (noin 100 kappaletta). Tummat pisteet ovat hankkeita, jotka kuuluivat todelliseen päätökseen. Nähdään, että osa hankkeista, jotka kuuluvat kaikkiin tehokkaisiin päälystysohjelmiin, eivät kuulu todelliseen päätökseen. Toisaalta todellinen päätös sisältää hankkeita, jotka eivät kuulu yhteenkään tehokkaaseen päälystysohjelmaan.

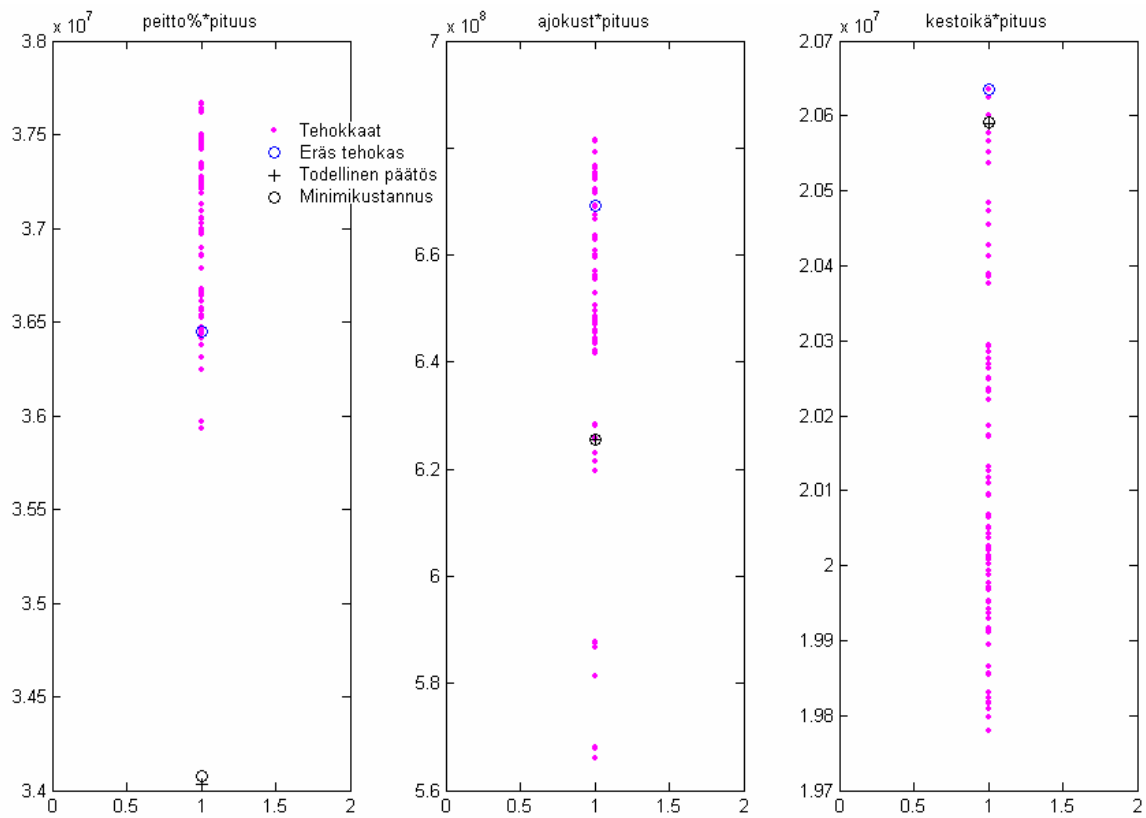


**Kuva 11 Ajo 3, ydinhankkeet: Todelliseen valittuun päällystysohjelmaan kuuluvat hankkeet on tummennettu. Todelliseen päällystysohjelmaan kuuluu hankkeita jotka eivät kuulu mihinkään tehokkaaseen päällystysohjelmaan ja toisaalta osa varmoista valinnoista ei kuulu siihen.**

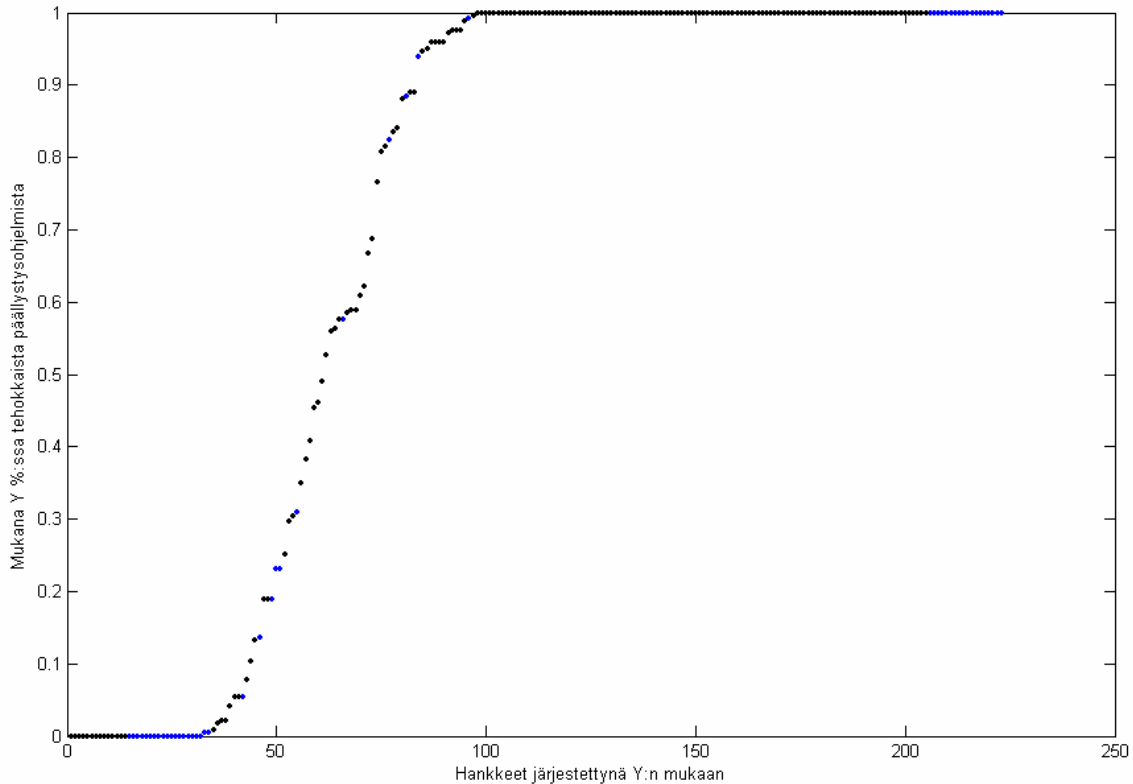
#### 5.2.4 Ajo 4

Tehokkaita päällystysohjelmia löytyi 83 kappaletta (hilapisteitä 243). Tulokset on edellisten ajojen tapaan visualisoitu kuvissa 12 ja 13.





**Kuva 12 Ajo 4, Päälystysohjelmien kriteerikohtaiset suoritustasot: Annetun preferenssi-informaation mukaan 'kestoikä\*pituus'-kriteeri on vähinten tärkein, mikä aiheuttaa siinä hajontaa. Ensisijaisesti optimoidaan kahta ensimmäistä kriteeriä, jolloin kolmannelle ei aseteta niin paljon painoa.**



**Kuva 13 Ajo 4, ydinhankkeet: Preferenssi-informaation avulla voidaan jo 40 hanketta hylätä ja toisaalta 125 valita varmasti.**

### 5.2.5 Yhteenveto tuloksista

Kaikkien ajojen tulokset on esitetty taulukossa 5. Hankkeiden määrä kertoo hanke-ehdokkaiden määrän ohjelmaa valitessa. Tehokkaiden päälystysohjelmien määrä kuvaa sitä, kuinka monta tehokasta, budjettiin sopivaa osajoukkoa hanke-ehdokkaista löydettiin kyseisessä ajossa. Hilapisteiden määrä kuvaa, kuinka monessa eri painopisteessä optimointitehtävä ratkaistiin. Varmasti valittavien hankkeiden määrä kertoo, kuinka moni hankkeista oli mukana kaikissa tehokkaissa päälystysohjelmissa (ydinluku=1). Varmasti pois jätettävien hankkeiden (ydinluku=0) määrä taas kertoo, kuinka moni hanke-ehdokkaista ei ollut mukana yhdessäkään tehokkaassa päälystysohjelmassa. Lisäksi ensimmäisestä aineistosta on laskettu säästö, joka on mahdollista saavuttaa valitsemalla minimikustannusohjelma, eli halvin päälystysohjelma, joka antaa jokaisen kriteerin suhteen yhtä hyvän tai paremman tuloksen kuin peittoprosentin avulla (heuristiikka 2) valittu ohjelma. Toisen aineiston kohdalla taas on laskettu säästö, joka on mahdollista

saavuttaa valitsemalla halvin päällystysohjelma, joka antaa jokaisen kriteerin suhteen yhtä hyvän tai paremman tuloksen kuin todellinen, tiepiirissä käytetty päällystysohjelma.

**Taulukko 5 Yhteenveto eri ajojen tuloksista**

	Aineisto 1		Aineisto 2	
	Ajo1	Ajo2	Ajo3	Ajo4
<b>Hanke-ehdokkaiden määrä</b>	626	626	222	222
<b>Budjetti (M€)</b>	20	20	16,3	16,3
<b>Tehokkaita päällystysohjelmia</b>	1233	212	397	83
<b>Hilapisteitä</b>	1326	243	1326	243
<b>Kriteerien tärkeysjärjestys annettu</b>	Ei	kyllä	Ei	kyllä
<b>Varmasti valittavia hankkeita</b>	n.150	n.300	n.100	n.125
<b>Varmasti poisjätettäviä hankkeita</b>	n.100	n.200	n.20	n.40
<b>Säästö (M€) (heuristiikka II vs. minimikust.)</b>	3,9	3,9	-	-
<b>Säästö (M€) (todellinen ohjelma vs. Minimikust.)</b>	-	-	1,3	1,3

Tehokkaita päällystysohjelmia oli molemmilla aineistoilla löydettävissä suuri määrä. Antamalla kriteereille tärkeysjärjestys tehokkaiden päällystysohjelmien määrä pieneni huomattavasti. Samalla hanke-ehdokkaat jakaantuivat entistä selkeämmin tehokkaissa päällystysohjelmissa mukana oleviin ja niiden ulkopuolelle jääviin hankkeisiin. Lisäksi kaikilla ajokerroilla löydettiin ratkaisuja, jotka antoivat kaikkien kriteerien suhteen parempia tuloksia kuin vertailussa käytetyt heuristiikat tai todellinen ohjelma.

## 6 Johtopäätökset

Tämän projektin tulosten perusteella monikriteerisen päätöksenteon menetelmillä voitaisiin saavuttaa merkittäviä lisähyötyjä päällystysohjelmaa valittaessa. Yhden kriteerin suhteen maksimoivilla heuristiikoilla ei päästä parhaaseen mahdolliseen ratkaisuun. Jo yhden kriteerin suhteen suoritettava tarkka optimointi antaa esiteltyihin heuristiikkoihin verrattuna huomattavasti paremman arvon valitun kriteerin suhteen. Tällöin löydetty ratkaisu – päällystysohjelma – maksimoi valitun kriteerin arvon. Monitavoitteisella lähestymisellä saavutettiin puolestaan kaikkien kriteerien suhteen parempia ratkaisuja kuin esiteltyjä heuristiikkoja käyttämällä. Käytettyjen kolmen tärkeimmän kriteerin suhteen menetelmällä saavutettiin myös todellista, tiepiiriin valitsemaa päällystysohjelmaa parempi ratkaisu.

Esittämämme malli sopisi hyvin päätöstukiohjelmistossa käytettäväksi. Ohjelmisto tulisi jakaa kahteen osaan: varsinaiseen ratkaisimeen ja interaktiiviseen käyttöliittymään. Ratkaiseminen voitaisiin aluksi suorittaa ilman preferenssi-informaatiota (koko painoalueessa) erillisellä ohjelmistolla, joka laskennan päätyttyä tallentaisi tehokkaat päällystysohjelmat jatkoanalyysiä varten. Tällöin laskennan kesto ei olisi rajoittava tekijä, koska se voitaisiin toteuttaa erillään interaktiivisesta päätöksentuesta. Esittämämme ratkaisumenetelmä voitaisiin helposti implementoida rinnakkaislaskentaympäristöön, jossa kullekin laskentayksikölle jaettaisiin joukko painopisteitä ratkaistavaksi.

Varsinaiseen päätöstukiohjelmistoon kuuluisi graafinen käyttöliittymä, jolla päätöksentekijä voisi analysoida interaktiivisesti ratkaisimen tallentamien tehokkaiden päällystysohjelmien joukkoa. Jos päätöksentekijä haluaisi antaa preferenssi-informaatiota, ei tehtävää tarvitsisi ratkaista kokonaan uudelleen, vaan ainoastaan etsiä koko painoalueessa lasketuista päällystysohjelmien joukosta ne päällystysohjelmat, jotka ovat tehokkaita uuden preferenssi-informaation valossa. Vaikka tehokkaita päällystysohjelmia olisi satoja, ei tämä operaatio veisi aikaa kuin pari sekuntia.

Graafisen käyttöliittymän tulisi sisältää ainakin tässä raportissa esitellyt tulosten visualisoinnit. Erityisesti kuvat, joissa on esitetty tehokkaiden päällystysohjelmien kriteerikohtaiset suoritustasot (ks. esim. kuva 10), olisivat erityisen hyödyllisiä, jos

käyttäjä voisi hiiren avulla korostaa valitsemansa päällystysohjelman suoritustasot. Tällöin päätöksentekijä voisi helposti tutkia, millaisia kombinaatioita eri kriteereistä on mahdollista saavuttaa. Toisaalta samaan kuvaan voisi liittää mahdollisuuden asettaa kriteerikohtaisia vaatimustasoja: kun päätöksentekijä näkee eri kriteerien vaihteluvälit, hän voi asettaa minimivaatimuksia kriteerikohtaisille suoritustasoille, mikä karsii tehokkaiden päällystysohjelmien joukkoa.

Päätöstukijärjestelmä tulisi implementoida ennen menetelmän esittelemistä loppukäyttäjille (tiehallintopiirien johtajat). Loppukäyttäjien kanssa tulisi tehdä yhteistyötä graafisen käyttöliittymän kehittämisessä ja ennen kaikkea relevanttien kriteerien valinnassa. Myös menetelmän kouluttamiseen loppukäyttäjille tulisi panostaa, jotta malli ymmärrettäisiin ainakin yleisellä tasolla. Koska malli on rakennettu hyvin intuitiiviseksi, uskomme, että loppukäyttäjät omaksuvat suhteellisen helposti mallin sisältämän optimointitehtävän yleisluonteen sekä tehokkaan päällystysohjelman käsitteen.

Malli voitaisiin laajentaa kattamaan päällystysohjelman valinnan lisäksi myös siihen kuuluvien hankkeiden skeduloinnin. Päätöksentekijälle ratkaistaisiin siis päällystysohjelman lisäksi optimaalinen aikataulu hankkeiden toteuttamisesta. Tämä vaatisi kuitenkin päätöstukiohjelmiston integroimista osaksi teiden kuntotiedon hallintajärjestelmää, jotta voitaisiin ennustaa hankkeiden kuntotiedossa tapahtuvat muutokset seuraavien vuosien kuluessa. Tällöin optimointimallin avulla voitaisiin päätellä, kannattaako yksittäinen hanke toteuttaa jo tänä vuonna vai siirtää tulevaisuuteen.

Teiden kuntotiedon hallinnassa voitaisiin ottaa entistä paremmin huomioon päätöksenteon tukeminen. Erityisen hyödyllistä olisi, jos hankkeiden välisistä synergioista olisi saatavissa dataa. Esimerkiksi maantieteellisesti toisiaan lähellä olevat hankkeet saattaa olla kannattavaa toteuttaa yhdessä, koska tällöin kaluston ja henkilökunnan kuljetuksista aiheutuvat kustannukset minimoituvat.

## 7 Viitteet

Arbel, A., (1989). Approximate Articulation of Preference and Priority Derivation, *European Journal of Operational Research*, Vol. 43, pp. 317-326.

Belt J., Lämsä V.P., Liimatta L., Ehrola E. (2000). Kevytpäällysteteiden vauriomallien ja mitoitusmenetelmien kehittämisen perusteet. Tielaitoksen selvityksiä 18/2000, TIEL 3200607, Edita Oyj, Helsinki

Belt J, Lämsä V.P., Savolainen M., Ehrola E, (2002) Tierakenteen vaurioituminen ja tiestön kunto. Tiehallinnon selvityksiä 15/2002. TIEH 3200747, Edita Oyj, Helsinki

Charnes, A., Cooper, W.W., Ferguson, R.O., (1955). Optimal Estimation of Executive Compensation by Linear Programming, *Management Science*, Vol. 1, No. 8, pp. 138-151.

Clemen, (1996). *Making Hard Decisions – An Introduction to Decision Analysis, 2nd Edition*, Duxbury Press, Cambridge.

Ehrgott, M., Gandibleux, X., (2000). A Survey and Annotated Bibliography of Multiobjective Combinatorial Optimization. *OR Spektrum*, Vol. 22, pp. 425-460.

Erlebach T., Keller H., Prefschy, U., (2002). Approximating Multiobjective Knapsack Problems, *Management Science*, Vol. 48, No. 12.

French, S., (1988). *Decision Theory – an Introduction to the Mathematics of Rationality*, Ellis Horwood Limited.

Hammond, J.S., Keeney, R.L., Raiffa, H., (1999). *Smart Choices: a Practical Guide to Making Better Decisions*, Harvard Business School Press, 1999.

Keeney, R.L., Raiffa, H., (1976). *Decisions with Multiple Objectives: Preferences and Value Trade-Offs*. John Wiley, New York.

Korhonen, P. and Wallenius, J. (1988): "A Pareto Race", *Naval Research Logistics*, Vol. 35, N:o 6, pp. 615-623.

Lahdelma, R., Hokkanen, J., Salminen, P., (1998). SMAA – Stochastic Multiobjective Acceptability Analysis. *European Journal of Operational Research*, Vol. 106, pp. 137-143.

Liesiö, J. (2004). Multi-Criteria Capital Budgeting with incomplete Information, Master's Thesis, Systems Analysis Laboratory, Helsinki University of Technology. (to appear at [http://www.sal.hut.fi/Thesis/pdf-files/.](http://www.sal.hut.fi/Thesis/pdf-files/))

Lämsä, V.P. (2001). Päälystetyn tiestön mitattu kunto. Tiehallinnon selvityksiä 82/2001. TIEL 3200728, Edita Oyj, Helsinki

Mild, P. (2004). Multicriteria Portfolio Analysis for Strategic Resource Allocation, Master's Thesis, Systems Analysis Laboratory, Helsinki University of Technology. (to appear at [http://www.sal.hut.fi/Thesis/pdf-files/.](http://www.sal.hut.fi/Thesis/pdf-files/))

Petäjä S., Spoo H. (2001) Päälysrakenteen elinkaarikustannusanalyysi. VTT Rakennus- ja yhdyskuntatekniikka. TPPT20, Espoo

Ruotoistenmäki A. (2003). Tieverkon kuntotiedon kehittäminen – Nykytilaselvitys. HKKK. Helsinki.

Saaty, T.L., (1980). *The Analytic Hierarchy Process*. McGraw – Hill, New York.

Salminen, L. (2004). Efficiency Analysis of Multi-Criteria Capital Budgeting Algorithms, Independent Research Project in Applied Mathematics, Systems Analysis Laboratory, Helsinki University of Technology. (to appear at [http://www.sal.hut.fi/Publications/index\\_fi.html](http://www.sal.hut.fi/Publications/index_fi.html))

Salo, A., Hämäinen, R. P., (1992). Preference Assessment by Imprecise Ratio Statements, *Operations Research*, Vol. 40, pp. 1053-1061.

Salo, A., Hämäläinen, R.P., (2003). Preference Programming. *Manuscript Submitted for Publication*, Systems Analysis Laboratory, HUT.

Salo, A., Hämäläinen, R. P., (1995). Preference Programming through Approximate Ratio Comparisons, *European Journal of Operational Research*, Vol. 82, pp. 458-475.

Salo, A., Hämäinen, R. P., (2001). Preference Ratios in Multiattribute Evaluation (PRIME) - Elicitation and Decision Procedures under Incomplete Information. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 31, pp. 533-545.

Steuer, R. E., (1986). *Multiple Criteria Optimization: Theory, Computation and Application*, John Wiley & Sons.

Stummer, C., Heidenberger, K., (2003). Interactive R&D Portfolio Analysis with Project Interdependencies and Time Profiles of Multiple Objectives, *IEEE Transactions on Engineering Management*, Vo. 50, No. 2, pp. 175-183.

Tiehallinto (2001). Tieliikenteen ajokustannukset 2000, Tiehallinnon ohjeita. TIEH 2123614-01, Edita Oyj, Helsinki

Tielaitos (1999). Päälystettyjen teiden ylläpidon toimintalinjat ja ohjaus, Tiehallinto, tie- ja liikenneolojen suunnittelu. TIEL 1000019, Edita Oyj, Helsinki

Wallenius, H. (1991). *Implementing Interactive Multiple Criteria Decision Methods in Public policy*, Jyväskylä Studies in Computer Science, Economics and Statistics, No. 18.

Weber, M., (1987). Decision Making with Incomplete Information. *European Journal of Operational Research*, Vol. 28, pp. 44-57.

Zionts, S., Wallenius, J., (1976). An Interactive Programming Method for Solving the Multiple Criteria Problem, *Management Science*, Vol. 22, No. 6, pp. 652-663.

Zitzler, E., Thiele, L., (1999). Multiobjective Evolutionary Algorithms: A Comparative Case Study and Strength Pareto Approach, *IEEE Transactions on evolutionary computation*, Vol. 3, No. 4.

Matlab - Language of Technical Computing. Ohjelmisto, Mathworks.

Xpress-MP. Ohjelmisto, Dash Optimization.



## Liite 1: Hilapisteistö $n = 3, k = 5$

niste	w 1	w 2	w 3
1	1	0	0
2	0.8	0.2	0
3	0.8	0	0.2
4	0.6	0.4	0
5	0.6	0.2	0.2
6	0.6	0	0.4
7	0.4	0.6	0
8	0.4	0.4	0.2
9	0.4	0.2	0.4
10	0.4	0	0.6
11	0.2	0.8	0
12	0.2	0.6	0.2
13	0.2	0.4	0.4
14	0.2	0.2	0.6
15	0.2	0	0.8
16	0	1	0
17	0	0.8	0.2
18	0	0.6	0.4
19	0	0.4	0.6
20	0	0.2	0.8
21	0	0	1