

LAPPEENRANNAN TEKNILLINEN YLIOPISTO
Tietotekniikan osasto

POPULAATION MONIMUOTOISUUDEN MITTAAMINEN LIUKULUKUKOODATUISSA EVOLUUTIOALGORITMEISSA

Diplomityön aihe on hyväksytty Tietotekniikan osaston osastoneuvostossa 12.2.2003.

Työn tarkastajana ja ohjaajana toimi Professori, KTT Jouni Lampinen ja toisena tarkastajana Lehtori, KTL Timo Mantere.

Lappeenrannassa 25.3.2003

Esa Ruuth
Ruotsalaisenraitti 3 B 21
53850 Lappeenranta
041 436 2156

TIIVISTELMÄ

Lappeenrannan teknillinen yliopisto
Tietotekniikan osasto

Esa Ruuth

Populaation monimuotoisuuden mittaaminen liukulukukoodatuissa evoluutioalgoritmeissa

Diplomityö

2003

53 sivua, 11 kuvaa, 1 taulukko ja 6 liitettä

Tarkastajat: Professori, KTT Jouni Lampinen ja Lehtori, KTL Timo Mantere

Hakusanat: Populaation monimuotoisuus, differentiaalievoluutio, evoluutioalgoritmit
Keywords: Population diversity, differential evolution, evolutionary algorithms

Diplomityössä esitetään menetelmä populaation monimuotoisuuden mittaamiseen liukulukukoodatuissa evoluutioalgoritmeissa, ja tarkastellaan kokeellisesti sen toimintaa. Evoluutioalgoritmit ovat populaatiopohjaisia menetelmiä, joilla pyritään ratkaisemaan optimointiongelmia. Evoluutioalgoritmeissa populaation monimuotoisuuden hallinta on välttämätöntä, jotta suoritettu haku olisi riittävän luotettavaa ja toisaalta riittävän nopeaa. Monimuotoisuuden mittaaminen on erityisen tarpeellista tutkittaessa evoluutioalgoritmien dynaamista käyttäytymistä.

Työssä tarkastellaan haku- ja tavoitefunktioavaruuden monimuotoisuuden mittaamista. Toistaiseksi ei ole ollut olemassa täysin tyydyttäviä monimuotoisuuden mittareita, ja työn tavoitteena on kehittää yleiskäyttöinen menetelmä liukulukukoodattujen evoluutioalgoritmien suhteellisen ja absoluuttisen monimuotoisuuden mittaamiseen hakuavaruudessa. Kehitettyjen mittareiden toimintaa ja käyttökelpoisuutta tarkastellaan kokeellisesti ratkaisemalla optimointiongelmia differentiaalievoluutioalgoritmillä.

Toteutettujen mittareiden toiminta perustuu keskihajontojen laskemiseen populaatiosta. Keskihajonnoille suoritetaan skaalaus, joko alkupopulaation tai nykyisen populaation suhteen, riippuen lasketaanko absoluuttista vai suhteellista monimuotoisuutta. Kokeellisessa tarkastelussa havaittiin kehitetyt mittarit toimiviksi ja käyttökelpoisiksi. Tavoitefunktion venyttäminen koordinaattiakseleiden suunnassa ei vaikuta mittarin toimintaan. Myöskään tavoitefunktion kiertäminen koordinaatistossa ei vaikuta mittareiden tuloksiin. Esitetyn menetelmän aikakompleksisuus riippuu lineaarisesti populaation koosta, ja mittarin toiminta on siten nopeaa suuriakin populaatioita käytettäessä. Suhteellinen monimuotoisuus antaa vertailukelpoisia tuloksia riippumatta parametrien lukumäärästä tai populaation koosta.

ABSTRACT

Lappeenranta University of Technology
Department of Information Technology

Esa Ruuth

Population Diversity Measurement in Floating Point Encoded Evolutionary Algorithms

Master's thesis

2003

53 pages, 11 figures, 1 table and 6 appendices

Supervisors: Professor, D.Sc. Jouni Lampinen and Lecturer, Lic.Sc. Timo Mantere

Keywords: Population diversity, differential evolution, evolutionary algorithms

In this thesis a new method for population diversity measurement for floating point encoded evolutionary algorithms is developed and tested experimentally. Evolutionary algorithms are population-based methods, which are used for solving optimization problems. In evolutionary algorithms it is necessary to control population diversity, to assure that search is reliable and on the other hand fast enough. Population diversity measurement is especially necessary when the dynamical behavior of an evolutionary algorithm is studied.

Population diversity measurement was considered in search and objective function spaces. So far there has not been fully satisfactory indicator for population diversity. The goal in this thesis is to develop a general method for population diversity measurement in floating point encoded evolutionary algorithms. Functionality and usability of developed indicators are considered experimentally by solving optimization problems with a differential evolution algorithm.

The developed indicators operate by calculating the standard deviations of the population. The standard deviations are then scaled regarding initial or current population, depending whether absolute or relative diversity is calculated. By experimental consideration the developed indicators were found functional and useful. Stretching of objective function in the direction of coordinate axes does not affect the operation of the indicator. Rotating of the objective function does not affect the results of the indicator. Time complexity of the developed method depends linearly on the population size, thus the operation of the indicator is fast even with large populations. Relative diversity gives comparable results regardless of number of dimensions or population size.

SISÄLLYSLUETTELO

1	JOHDANTO	5
2	POPULAATION MONIMUOTOISUUDEN MITTAAMINEN	7
2.1	Monimuotoisuuden merkitys evoluutioalgoritmeissa	7
2.2	Monitavoiteoptimointi	9
2.3	Populaation monimuotoisuus	12
2.3.1	Smuc	12
2.3.2	Lopez Cruz <i>et al.</i>	13
2.3.3	Luokka ja Saastamoinen	14
2.4	Pareto-optimaalisen rintaman monimuotoisuus	16
2.4.1	Spacing	16
2.4.2	Spread	17
2.4.3	Maximum spread	17
3	DIFFERENTIAALIEVOLUUTIOALGORITMI	19
3.1	Populaation alustus	20
3.2	Mutaatio ja risteytys	20
3.3	Valinta	21
3.4	Rajoitteet	22
4	MONIMUOTOISUUTTA MITTAAVIEN MITTAREIDEN KEHITTÄMINEN	25
4.1	Alkupopulaation monimuotoisuus	25
4.2	Suhteellinen ja absoluuttinen monimuotoisuus	28
5	MITTAREIDEN TOIMINNAN TARKASTELU	32
5.1	Testiongelmat	32
5.1.1	Testifunktio Sphere	33
5.1.2	Testifunktio Ellipsoid 1	34
5.1.3	Testifunktio Ellipsoid 2	35
5.1.4	Testifunktio Rosenbrock	36
5.1.5	Testifunktio Schwefel	37
5.1.6	Testifunktio Griewank	38
5.2	Kokeellinen tarkastelu	39
6	TULOKSET	41
6.1	Testitapaus 1	41
6.2	Testitapaus 2	43
6.3	Testitapaus 3	44
6.4	Testitapaus 4	44
6.5	Testitapaus 5	45
6.6	Tulosten tulkintaa	46
7	JOHTOPÄÄTÖKSET	48
	LÄHTEET	
	LIITTEET	

SYMBOLI- JA LYHENNELUETTELO

Δ	Populaation monimuotoisuus, Population diversity
Δ_A	Populaation absoluuttinen monimuotoisuus, Absolute diversity
Δ_R	Populaation suhteellinen monimuotoisuus, Relative diversity
ε	Toleranssiraja kromosomin monimuotoisuudelle, Tolerance for the chromosome diversity
η	Monimuotoisuusindeksi, Diversity index
$\mathcal{O}(\cdot)$	Kompleksisuusluokan yläraja, Upper limit for complexity
ρ	Populaation monimuotoisuusaste, Degree of population diversity
a	Kerroin, joka ilmaisee valintaoperaation vaikutuksen populaation monimuotoisuuteen, Effect of selection operation on population diversity
c	Kerroin, joka ilmaisee variaatio-operaatioiden vaikutuksen populaation monimuotoisuuteen, Effect of variation operations on population diversity
CR	Risteytysvakio, Crossover constant
D	Yksilövektorin parametrien (kromosomien) lukumäärä, Number of dimensions
$d(\mathbf{u}, \mathbf{v})$	Kahden yksilön \mathbf{u} ja \mathbf{v} välinen etäisyys, Distance between two individuals \mathbf{u} and \mathbf{v}
F	Mutaatiovakio, Mutation constant
$f(\cdot)$	Tavoitefunktio, Objective function
$f(\mathbf{X}^*)$	Funktion optimiarvo, Optimum value
G	Sukupolvi, Generation
G_{max}	Sukupolvien maksimimäärä, Maximum number of generations
k	Indeksi joukon \mathcal{Q} yksilöön, Index to an individual in set \mathcal{Q}
l	Tavoitefunktioiden lukumäärä, Number of objective functions
NP	Populaation koko, Population size
P_G	Sukupolven G populaatio, Population of generation G
P_{G+1}^*	Seuraavan sukupolven yritepopulaatio, Trial population for the next generation

$rand[0,1]$	Tasaisesti jakautunut satunnaismuuttuja väliltä [0,1], Evenly distributed random variable between [0,1]
$U_{i,G+1}$	Yritevektori, joka on muodostettu käyttäen kohdevektorina populaation i :ttä vektoria, Trial vector
$u_{j,i,G+1}$	Yritevektorin j :s parametri (kromosomi), Chromosome of the trial vector
$Var(\cdot)$	Varianssi, Variance
$V_{i,G+1}$	Kohinavektori, joka on muodostettu käyttäen kohdevektorina populaation i :ttä vektoria, Noisy vector
$v_{j,i,G+1}$	Kohinavektorin j :s parametri (kromosomi), Chromosome of the noisy vector
$X_{i,G}$	Sukupolven G populaation i :s yksilö, Individual
$x_{j,i,G}$	Yksilön j :s parametri (kromosomi), Chromosome
$X^{(L)}$	Yksilövektorin alaraja, Lower limit for the individual
$X^{(U)}$	Yksilövektorin yläraja, Upper limit for the individual
$x_j^{(L)}$	Yksilön j :nnen kromosomin arvon alaraja, Lower limit for the j :th chromosome
$x_j^{(U)}$	Yksilön j :nnen kromosomin arvon yläraja, Upper limit for the j :th chromosome
Q	Ei-dominoitujen ratkaisuiden joukko, Set of non-dominated solutions
$ Q $	Ei-dominoitujen ratkaisuiden lukumäärä, Number of non-dominated solutions
DE	Differentiaalievoluutio, Differential Evolution
EA	Evoluutioalgoritmi, Evolutionary Algorithm

ALKUSANAT

Tämä työ on tehty Lappeenrannan teknillisen yliopiston Tietotekniikan osastolle. Kiitän Tietojenkäsittelytekniikan laitosta, joka mahdollisti tämän työn tekemisen. Haluan kiittää myös Tietojenkäsittelytekniikan laitoksen henkilökuntaa mukavasta työympäristöstä.

Kiitän työn ohjaajaa ja tarkastajaa, professori Jouni Lampista hyvästä ohjauksesta ja mielenkiinnosta työtäni kohtaan. Kiitokset myös työn toiselle tarkastajalle, lehtori Timo Mantereelle. Kiitän myös tekn. yo. Jani Rönkköstä saamastani avusta työn aikana ja DI Jarkko Vartiaista työni oikolukemisesta.

1 JOHDANTO

Evoluutioalgoritmit ovat menetelmiä, joilla pyritään ratkaisemaan optimointiongelmia luonnon evoluutiota jäljittelevällä tavalla. Evoluutioalgoritmit ovat populaatiopohjaisia menetelmiä, joissa alkupopulaatio muodostetaan yleensä satunnaisesti. Populaation jokainen jäsen, yksilö, on ratkaisuvaihtoehto ratkaistavaan ongelmaan. Alkupopulaation luomisen jälkeen sen jäsenille suoritetaan mutaatio- ja risteytysoperaatioita. Yksilöille lasketaan tavoitefunktion arvot käyttämällä optimoitavaa funktiota. Parhaat yksilöt valitaan seuraavaan sukupolveen, ja huonoimmat yksilöt karsiutuvat. Näin hyvien yksilöiden ominaisuudet periytyvät seuraaviin sukupolviin, ja huonot ominaisuudet katoavat. Evoluutiota jatketaan kunnes optimaalinen tai riittävän hyvä yksilö löytyy, ennalta asetettu sukupolvien maksimimäärä tai muu lopetusehto täyttyy.

Evoluutioalgoritmien käyttö on lisääntynyt viime vuosina, sillä niiden toteuttaminen ja käyttäminen on melko helppoa. Useiden epälineaaristen optimointiongelmiin ratkaiseminen perinteisillä algoritmeilla on hidasta, ellei jopa mahdotonta, sillä mahdollisten ratkaisuvaihtoehtojen lukumäärä kasvaa räjähdysmäisesti ongelmakoon kasvaessa. Evoluutioalgoritmissa uudet yrittäjät muodostetaan kunkin sukupolven parhaista ratkaisuksista, jolloin pyritään välttämään kaikkien mahdollisten ratkaisuvaihtoehtojen läpikäyminen.

Perinteisesti suurin osa optimoinnin tutkimuksesta ja sovelluksista on keskittynyt yksitavoiteongelmien ratkaisemiseen. Reaalimaailman ongelmista suurin osa sisältää useamman kuin yhden tavoitteen. Tehokkaiden monitavoiteoptimointimenetelmien puuttuessa on monitavoiteongelmia tavallisimmin ratkaistu muuntamalla ne yksitavoiteongelmiksi. Tästä seuraa kuitenkin se, että optimoinnin tuloksena saadaan vain yksi ratkaisu, vaikka monitavoiteongelman ratkaisuna tulisi olla useita vaihtoehtoisia ratkaisuja, eli niin sanottu Pareto-optimaalisten ratkaisuiden joukko. Evoluutioalgoritmit ovat osoittautuneet käyttökelpoisiksi myös monitavoiteoptimointiongelmiin ratkaisemisessa, sillä populaatiopohjaisina menetelminä ne tuottavat useita ratkaisuja samalla kertaa.

Monitavoiteoptimoinnin tulosten paremmuuden vertailu on ollut viime vuosina tutkijoiden kiinnostuksen aiheena. Tarkoitukseen on kehitetty useita erilaisia mittareita, jotka mittaavat Pareto-optimaalisen ratkaisun hyvyttä. Useista eri lähestymistavoista huolimatta ei hyvää ja yleiskäyttöistä mittaria ole onnistuttu kehittämään. Työssä kehitettävän mittarin tulisi olla myöhemmin yleistettävissä mittaamaan Pareto-optimaalisten pisteiden jakautumista tavoitefunktioavaruudessa. Useiden aiempien mittausten ongelmien on ollut laskennallinen raskaus ja kompleksisuus, joka etenkin ongelmakoon kasvaessa aiheuttaa ongelmia. Lisäksi niiden antamat tulokset eivät ole olleet informatiivisia, eivätkä keskenään vertailukelpoisia.

Tämän työn tavoitteena on kehittää liukulukukoodatulle evoluutioalgoritmille sopiva menetelmä monimuotoisuuden mittaamiseksi hakuavaruuden pisteille. Yksitavoiteongelmien tapauksessa populaation monimuotoisuuden tulisi hävitä, kun taas monitavoiteongelmilla monimuotoisuuden tulisi säilyä riittävän suurena, sallien suppenemisen Pareto-rintamalle. Monitavoiteoptimoinnissa monimuotoisuuden tulisi säilyä korkeana tavoitefunktioavaruudessa. Kehitettyjen menetelmien toimivuutta tarkastellaan kokeellisesti ratkaisemalla optimointiongelmiä differentiaalievoluutioalgoritmilla.

2 POPULAATION MONIMUOTOISUUDEN MITTAAMINEN

Evoluutioalgoritmien toiminnan ja dynaamisen käyttäytymisen mittaamiseksi on kehitetty useita erilaisia mittareita. Tässä työssä keskitytään näistä vain populaation monimuotoisuuden mittaamiseen. Populaation monimuotoisuus kuvaa populaation jäsenten sijaintia toisiinsa nähden, ja sen mittaamiseen käytetään yksilöiden välisiä eroavaisuuksia. Yksilöiden keskittyessä pienelle alueelle on niiden monimuotoisuus pieni, ja päinvastoin. Monimuotoisuuden mittaamiseksi on kehitetty useita hyvinkin erilaisia mittareita, mutta siitä huolimatta luotettavaa, nopeaa ja monikäyttöistä mittaria ei ole onnistuttu luomaan. Tässä kappaleessa esitellään muutamia kirjallisuudessa esitettyjä monimuotoisuuden mittareita liukulukukoodatuille evoluutioalgoritmeille.

Populaation monimuotoisuutta voidaan mitata sekä haku- että tavoitefunktioavaruudessa. Yksitavoiteongelmissa populaation hakuavaruuden monimuotoisuuden tulisi vähentyä optimoinnin edetessä ja olla optimoinnin lopussa mahdollisimman pieni. Monitavoiteongelmissa pyritään tavoitefunktioavaruuden monimuotoisuus säilyttämään korkeana koko optimoinnin ajan, sallien kuitenkin suppeneminen Pareto-rintamalle.

2.1 Monimuotoisuuden merkitys evoluutioalgoritmeissa

Tyypillisesti evoluutioalgoritmin valintaoperaatio vähentää populaation monimuotoisuutta siten, että populaatio suppenee (konvergoituu). Populaation varianssia, $Var(\mathbf{P})$, käyttäen tämä voidaan ilmaista seuraavasti:

$$Var(\mathbf{P}_{G+1}) = a \cdot Var(\mathbf{P}'_{G+1}) \quad , \quad (1)$$

jossa

\mathbf{P}_{G+1} on seuraavan sukupolven populaatio,

\mathbf{P}'_{G+1} on seuraavan sukupolven yritteiden populaatio,

$a \leq 1$ on kerroin, joka ilmaisee valintaoperaation vaikutuksen populaation

monimuotoisuuteen. Oletetaan, että mielekkäästi toteutettu valintaoperaatio ei voi lisätä monimuotoisuutta, koska yleensä valitaan vain parhaita yksilöitä. Tyypillisesti $a < 1$, jolloin valinta aiheuttaa populaation suppenemisen.

Tavallisesti variaatio-operaatiot, mutaatio ja risteytys, suunnitellaan lisäämään populaation monimuotoisuutta:

$$\text{Var}(\mathbf{P}'_{G+1}) = c \cdot \text{Var}(\mathbf{P}_G) \quad , \quad (2)$$

jossa $c \geq 1$ on kerroin, joka ilmaisee variaatio-operaatioiden vaikutuksen populaation monimuotoisuuteen. Tyypillisesti $c > 1$, jolloin populaation monimuotoisuutta pyritään lisäämään. Yhdistämällä (1) ja (2) saadaan:

$$\text{Var}(\mathbf{P}_{G+1}) = a \cdot c \cdot \text{Var}(\mathbf{P}_G) \quad . \quad (3)$$

Populaation toivotaan suppenevan globaaliin optimiin. Suppenemisehdoksi voidaan edellisen perusteella päätellä:

$$a \cdot c < 1 \quad . \quad (4)$$

Koska parhaiden ratkaisuiden valinta yhdessä tavoitefunktion kanssa määrää, että

$$a \leq 1 \quad , \quad (5)$$

niin evoluutioalgoritmin variaatio-operaatiot tulisi suunnitella siten, että

$$c \geq 1 \quad . \quad (6)$$

Yhdistettäessä a ja c , vaikuttavat ne populaatioon seuraavasti:

jos $a \cdot c > 1$	populaatio hajaantuu,
jos $a \cdot c \ll 1$	populaatio suppenee liian nopeasti, eikä hakuvaruutta käydä riittävän kattavasti läpi,

jos $0 \ll a \cdot c < 1$ populaatio suppenee sopivalla nopeudella, haku on riittävän kattava ja riittävän nopea.

Täten c :n optimiarvo, joka riippuu tavoitefunktion ominaisuuksista, ja siten a :sta, on myös riippuva ratkaistavasta ongelmasta. [Zah02]

Jotta evoluutioalgoritmin dynaamista käyttäytymistä, ja sille tärkeää populaation monimuotoisuutta voitaisiin kokeellisesti tutkia, tarvitaan mittari populaation monimuotoisuudelle. Täysin tyydyttäviä mittareita ei ole tähän mennessä kyetty kehittämään liukulukukoodatuille evoluutioalgoritmeille.

Tarvittavia mittareita ovat:

- absoluuttisen monimuotoisuuden mittari, joka kuvaa kuinka populaation jäsenet ovat jakautuneet hakuavaruudessa. Suuri absoluuttisen monimuotoisuuden arvo tarkoittaa hakuavaruudessa hajallaan olevaa populaatiota, ja pieni arvo pienelle alueelle keskittynyttä populaatiota.
- suhteellisen monimuotoisuuden mittari, joka kuvaa kuinka populaation jäsenet ovat jakautuneet hakuavaruudessa suhteessa toisiinsa.
- näiden variaatiot monitavoiteongelmien ratkaisujoukon monimuotoisuuden mittaamiseen tavoitefunktioavaruudessa.

2.2 Monitavoiteoptimointi

Merkittävä osa optimoinnin tutkimuksesta ja sovelluksista keskittyy pelkästään yksitavoiteongelmien ratkaisemiseen, vaikka reaali maailman ongelmista suurin osa sisältää useamman kuin yhden tavoitefunktion. Koska monitavoiteoptimoinnissa tavoitefunktioita on useita, saadaan optimoinnin tuloksena myös useita vaihtoehtoisia ratkaisuja. Perinteiset optimointimenetelmät voivat parhaimmillaan löytää yhden ratkaisun yhdellä simulointiajolla, jonka vuoksi ne eivät sovellu monitavoiteongelmien ratkaisemiseen. Evoluutioalgoritmit ovat populaatiopohjaisia menetelmiä ja voivat siten löytää useita

optimaalisia ratkaisuja yhdellä simulointiajolla. Tämän vuoksi evoluutioalgoritmit sopivat hyvin monitavoiteongelmien ratkaisemiseen. [Deb02]

Yleispätevien ja tehokkaiden monitavoiteoptimointimenetelmien puuttuessa on monitavoiteongelmia tavallisimmin ratkaistu muuttamalla ne yksitavoiteongelmiksi. Tämä on tehty antamalla tavoitefunktioille ennalta määrättyt painot, ja yhdistämällä ne yhdeksi tavoitefunktioiksi. Tämän jälkeen optimointiongelma on voitu ratkaista käyttämällä yksitavoiteongelmien ratkaisemiseen kehitettyjä algoritmeja. Ongelmia tässä menettelyssä aiheuttaa eri tavoitteiden keskinäisen painoituksen, tavoitteiden suhteellisen tärkeyden ilmaiseminen *a priori*, sillä yhdellä painoyhdistelmällä saadaan vain yksi ratkaisu. [Lam00]

Toinen tapa ratkaista monitavoiteoptimointiongelmiä on käyttää Pareto-optimointia. Pareto-optimoinnissa ei tarvita etukäteen määriteltyjä painoituksia, ja sen tuloksena saadaan kerralla kaikki eri ratkaisuvaihtoehdot. Ratkaisun sanotaan olevan Pareto-optimaalinen mikäli yksikään hyväksyttävä ratkaisu ei dominoi sitä. Ei-dominoitu (Non-dominated) ratkaisu tarkoittaa, ettei ole olemassa ratkaisua, jonka tavoitefunktion arvoista vähintään yksi on saatua ratkaisua parempi, ja loput tavoitefunktion arvot yhtä hyviä tai parempia. Pareto-optimoinnin antamat keskenään yhtä hyvät ratkaisut muodostavat niin sanotun Pareto-optimaalisen rintaman (Pareto-front). Koska täydellisen Pareto-optimaalisen rintaman löytäminen tuntuu olevan mahdotonta millään olemassa olevalla menetelmällä, on yleensä tyydyttävä aproksimoimaan Pareto-optimaalista rintamaa rajatulla lukumäärällä Pareto-optimaalisia pisteitä. [Lam00]

Pareto-dominanssi (Pareto dominance):

Vektori U sisältää l tavoitefunktion arvoa hakuavaruuden S pisteelle X , siten U on tavoitefunktioavaruuden piste $U = \{f_1(X), \dots, f_l(X)\}$.

Vektorin $U^* = \{f_1(X^*), \dots, f_l(X^*)\}$ sanotaan dominoivan vektoria $U = \{f_1(X), \dots, f_l(X)\}$ jos ja vain jos U^* on osittain pienempi kuin U , eli $\forall i \in \{1, \dots, l\}, f_i(X^*) \leq f_i(X) \wedge \exists i \in \{1, \dots, l\}: f_i(X^*) < f_i(X)$.

Vahvasti tehollinen ratkaisu (Strongly efficient solution):

Ratkaisu $\mathbf{X}^* \in S$ on vahvasti tehollinen ratkaisu, jos ei ole olemassa toista ratkaisua $\mathbf{X} \in S$ siten, että $f(\mathbf{X}) \leq f(\mathbf{X}^*)$ kaikilla $i = 1, \dots, l$ ja $f(\mathbf{X}) < f(\mathbf{X}^*)$ vähintään yhdellä i :llä.

Heikosti tehollinen ratkaisu (Weakly efficient solution):

Ratkaisu $\mathbf{X}^* \in S$ on heikosti tehollinen ratkaisu, jos ei ole olemassa toista ratkaisua $\mathbf{X} \in S$ siten, että $f(\mathbf{X}) < f(\mathbf{X}^*)$ kaikilla $i = 1, \dots, l$. Vahvasti teholliset ratkaisut ovat heikosti tehollisten ratkaisuiden alijoukko.

Ei-dominoitu ratkaisu (Non-dominated solution):

Monitavoiteongelmia ratkaistaessa algoritmit aproksimoivat Pareto-rintamaa rajallisella määrällä erillisiä ratkaisuja. Ratkaisu on ei-dominoitu, mikäli mikään muu kyseisen ratkaisujoukon jäsen ei dominoi kyseistä ratkaisua. On huomattava, että saattaa olla olemassa sallittuja ratkaisuja, jotka voisivat dominoida kyseisen ratkaisujoukon pisteitä, mutta niitä ei vain ole löydetty.

Pareto-optimaalinen ratkaisu (Pareto-optimal solution):

Pareto-optimaalisella ratkaisulla tarkoitetaan usein ei-dominoitua ratkaisua. On kuitenkin syytä tarkistaa millaista määritelmää Pareto-optimaaliselle ratkaisulle on käytetty, sillä terminologia vaihtelee suuresti kirjallisuudessa.

Pareto-rintama (Pareto-front):

Pareto-rintama on Pareto-optimaalisten ratkaisuiden muodostama pinta tavoitefunktioavaruudessa. Nimitystä käytetään joskus myös diskreetistä joukosta Pareto-optimaalisia ratkaisuja.

Ideaaliset pisteet (Ideal points):

Ideaalisen vektorin $\{f_1^*, \dots, f_l^*\}$ komponentit saadaan minimoimalla jokainen tavoitefunktio yksitellen sallitussa hakuavaruudessa S . [Lam00]

2.3 Populaation monimuotoisuus

Barker ja Martin [Bar00] määrittivät ominaisuudet, jotka minkä tahansa kahden yksilön \mathbf{u} ja \mathbf{v} välisellä etäisyysfunktiolla $d(\mathbf{u}, \mathbf{v})$ tulee olla:

- symmetrisyys: $d(\mathbf{u}, \mathbf{v}) = d(\mathbf{v}, \mathbf{u})$,
- ei-negatiivisuus: $d(\mathbf{u}, \mathbf{v}) \geq 0$,
- nolla ainoastaan identtisille yksilöille: $d(\mathbf{u}, \mathbf{v}) = 0$ vain ja jos vain $\mathbf{u} = \mathbf{v}$,
- additiivinen (additive) siten, että $d(\mathbf{u}, \mathbf{v}) = \sum_{i=1}^{NP} d_i(\mathbf{u}_i, \mathbf{v}_i)$, jossa d_i :t voivat olla mielivaltaisia funktioita, jotka täyttävät edellä mainitut ehdot.

He määrittivät populaation monimuotoisuuden mittarin $\Delta_{BM}(G)$ kaikkien populaation yksilöiden välisten pareittaisten etäisyyksien summana ajanhetkellä G :

$$\Delta_{BM}(G) \equiv \frac{1}{2} \sum_{i=1}^{NP} \sum_{j=1}^{NP} d(\mathbf{X}_{i,G}, \mathbf{X}_{j,G}) = \frac{NP(NP-1)}{2} \bar{d}_G, \quad (7)$$

jossa NP on populaation koko, $\mathbf{X}_{i,G}$ ja $\mathbf{X}_{j,G}$ ovat populaation jäseniä, ja \bar{d}_G on kaikkien etäisyyksien keskiarvo hetkellä G . Etäisyysfunktiona voidaan käyttää mitä tahansa edellä mainitut ehdot täyttävää funktiota, esimerkiksi L1-normia.

2.3.1 Šmuc

Šmuc [Šmu02] esitteli differentiaalievoluutioalgoritmiin populaation päivittämiseksi menetelmän, jonka tarkoituksena oli estää juuttumista (Stagnation) [Lam00a] ja parantaa algoritmin suppenemisnopeutta. Hän seurasi populaation tilaa mittaamalla sen monimuotoisuutta. Työssä käytetty mittari mittaa nykyisen populaation keskimääräistä monimuotoisuutta:

$$\Delta_{Smuc}(G) = \frac{\sum_{i=1}^{NP} \sum_{j=i+1}^{NP} \left(\|\mathbf{X}_{i,G} - \mathbf{X}_{j,G}\| / (\mathbf{X}^{(U)} - \mathbf{X}^{(L)}) \right)}{2 \cdot D \cdot (NP-1) \cdot NP}, \quad (8)$$

jossa $X_{i,G}$ ja $X_{j,G}$ ovat i :s ja j :s D -dimensioinen yksilö nykyisessä sukupolvessa G . Muuttujat $X^{(U)}$ ja $X^{(L)}$ ovat yksilöiden ylä- ja alarajat. Tuloksena saadaan nykyisen sukupolven jäsenten väliset keskimääräiset normalisoidut etäisyydet, jotka kuvaavat ratkaisun suppenemista. Šmuc käytti monimuotoisuuden laskemista differentiaalievoluutioalgoritmin populaation monimuotoisuuden säilyttämiseen päivittämällä populaatiota monimuotoisuuden laskiessa määritellyn toleranssirajan ali. Populaation päivittäminen tapahtui korvaamalla populaation satunnaisia kromosomeja satunnaisesti generoiduillaromosomeilla, kuitenkin parhaan yksilönromosomeja muuttamatta. Monimuotoisuutta laskettaessa joudutaan laskemaan kaikkien populaation jäsenten väliset etäisyydet, joka on etenkin suuria populaatioita käytettäessä raskas ja hidas operaatio. Haittana voidaan pitää myös menetelmän tarvitsemia ylimääräisiä parametreja, kuten esimerkiksi monimuotoisuuden toleranssirajaa. Esitettyjen tulosten perusteella menetelmä näyttää toimivan ja estävän algoritmin tilojen juuttumista pientä populaation kokoa käytettäessä.

Yhtälössä (8) esitetyn algoritmin aikakompleksisuus on luokkaa:

$$O(NP^2 \cdot D \cdot G).$$

Populaation monimuotoisuuden laskemiseen tarvittava aika kasvaa eksponentiaalisesti populaation koon kasvaessa. Siten se ei sovellu kaikkein vaikeimmille ongelmille, joiden ratkaiseminen vaatii hyvin suurta populaatiota.

2.3.2 Lopez Cruz *et al.*

Lopez Cruz *et al.* [Lop01] käytti populaation monimuotoisuutta hyväksi differentiaalievoluutioalgoritmin mutaatio- ja risteytysparametrien säädössä. Parametrien säätö tapahtuu ajonaikaisesti, ja ratkaistavan ongelman tilaa seurataan tarkkailemalla populaation monimuotoisuutta parhaan yksilön ympärillä. Tätä varten määritellään monimuotoisuusindeksi (Diversity index):

$$\eta_{ji} = \begin{cases} 1, & \text{jos } \left| \frac{x_{j,i,G} - x_{j,best,G}}{x_{j,best,G}} \right| > \varepsilon \\ 0, & \text{muutoin} \end{cases}, \quad (9)$$

jossa $x_{j,best,G}$ on sukupolven G populaation parhaan yksilön j :s kromosomi, ja ε on ennalta määrätty toleranssi kromosomin monimuotoisuudelle. Arvo $\eta = 1$ tarkoittaa siis monimuotoista kromosomia. Populaation monimuotoisuus ($0 \leq \rho \leq 1$) lasketaan seuraavasti:

$$\rho = \sum_{i=1}^{NP} \sum_{\substack{j=1 \\ i \neq best}}^D \eta_{ji} / (D \cdot (NP - 1)) . \quad (10)$$

Monimuotoisuusmittarin huonona puolena on se, että populaatio joudutaan käymään läpi kahdesti, ensin monimuotoisuusindeksiä luotaessa, ja toisen kerran varsinaista monimuotoisuutta laskettaessa. Lisäksi pitää määrittää toleranssiraja ε , joka määrää sen onko kromosomi monimuotoinen vai ei. Mittari ei siten ota huomioon monimuotoisuuden määrää. Esitetty monimuotoisuusmittari on siten toleranssirajan ylittävien kromosomien lukumäärä jaettuna kaikkien kromosomien lukumäärällä

Yhtälön (10) monimuotoisuuden laskennan aikakompleksisuus on luokkaa:

$$O(NP \cdot D \cdot G).$$

Monimuotoisuuden laskemiseen tarvittava aika kasvaa lineaarisesti populaation koon kasvaessa.

2.3.3 Luokka ja Saastamoinen

Luokka ja Saastamoinen [Luu02] esittelivät uuden lähestymistavan populaation monimuotoisuuden säilyttämiseksi. Menetelmän tavoitteena on mahdollistaa nopea suppeneminen ja välttää ennaikasta suppenemistä. Menetelmässä käytetään sumeaa samankaltaisuutta (Fuzzy similarity) säätämään mutaatioastetta (Mutation rate), joka

riippuu risteytettävistä yksilöistä. Kahden yksilön välisen samankaltaisuuden perusteella lasketaan mutaatioaste niiden jälkeläisille. Kahden geneettisesti lähekkäisen yksilön jälkeläiselle tulee korkea mutaatioaste, ja geneettisesti toisistaan paljon poikkeavien yksilöiden jälkeläiselle mutaatioaste on matala. Perinteisissä menetelmissä huomattavasti muita paremman yksilön ilmaantuessa populaatioon valitaan se useita kertoja seuraaviin populaatioihin, jolloin sen vaikutus koko populaatioon on lopulta huomattava. Tuloksena on lähes yhdenmukainen populaatio, josta on arvattavasti usein seurauksena ennenaikainen suppeneminen. Samanlaisuuteen perustuvassa mutaatioasteen säädössä huomattavan hyvän yksilön vaikutuksesta syntyneet lähes identtiset yksilöt aiheuttavat keskenään risteytyessään jälkeläisilleen erittäin korkean mutaatioasteen. Korkean mutaatioasteen haittana voidaan pitää hyvien ratkaisuiden mahdollista häviämistä populaatiosta. Toisaalta korkea mutaatioaste varmistaa korkean monimuotoisuuden populaatiossa, jolloin hyvien ratkaisuiden mahdollinen häviäminen populaatiosta ei vaikuta ratkaisevasti saatavaan ratkaisuun.

Kahden yksilön X ja Y samankaltaisuus lasketaan seuraavasti [Luu02]:

$$SIM(X, Y) = 1 - \left(\frac{\sum_{i=1}^D W_i |x_i - y_i|^r}{\sum_{i=1}^D W_i (x_i^{(U)} - x_i^{(L)})} \right)^{\frac{1}{r}}, \quad (11)$$

jossa x_i ja y_i ovat X :n ja Y :n i :nnet muuttujat, $x_i^{(U)}$ ja $x_i^{(L)}$ ovat i :nnet muuttujan ylä- ja alarajat ja W_i on ennalta määrätty paino. Liian voimakkaan mutaation välttämiseksi on määritetty korkein sallittu mutaatioaste M_m . Siten tehollinen mutaatioaste (Effective mutation rate)

$$P_r(X, Y) = SIM(X, Y)M_m . \quad (12)$$

M_m ja r ovat kiinteitä käyttäjän määräämiä arvoja, joille julkaisussa [Luu02] on käytetty arvoja $M_m = 0,9$ ja $r = 0,2$. Kirjoittajien tekemien testien mukaan menetelmä toimi muita

testattuja menetelmiä paremmin kolmessa tapauksessa neljästä. Menetelmän tehokkuudesta on olemassa vain vähän kokeellista tietoa.

2.4 Pareto-optimaalisen rintaman monimuotoisuus

Monitavoiteoptimoinnin tuloksena saadut ei-dominoidut ratkaisut aproksimoivat todellista Pareto-optimaalista rintamaa tavoitefunktioavaruudessa. Jotta saatu ratkaisu olisi hyvä, on ei-dominoitujen ratkaisuiden noudatettava mahdollisimman hyvin todellista Pareto-optimaalista rintamaa. Ratkaisun hyvyyteen vaikuttaa siten ei-dominoitujen ratkaisuiden etäisyys Pareto-optimaaliseen rintamaan ja se, kuinka hyvin saavutettu ratkaisu kattaa koko Pareto-optimaalisen rintaman alueen. Ei-dominoitujen ratkaisuiden tulisi olla myös mahdollisimman tasaisesti jakautuneita Pareto-optimaaliseen rintamaan nähden.

Molempien ominaisuuksien mittaamiseen on kehitetty useita menetelmiä, sekä niin sanottuja yhdistelmämittareita, jotka mittaavat molempia ominaisuuksia. Tässä työssä käsitellään kuitenkin vain ei-dominoitujen ratkaisuiden kattavuutta todelliseen Pareto-optimaaliseen rintamaan nähden monimuotoisuutta mittaamalla. Seuraavaksi on esitelty muutamia yleisimmin käytettyjä Pareto-optimaalisen rintaman kattavuutta mittaavia menetelmiä.

2.4.1 Spacing

Schott [Sch95] kehitti mittarin, jossa lasketaan peräkkäisten ei-dominoitujen ratkaisuiden väliset suhteelliset etäisyydet:

$$spacing = \sqrt{\frac{1}{|\mathcal{Q}|} \sum_{i=1}^{|\mathcal{Q}|} (d_i - \bar{d})^2} , \quad (13)$$

jossa \mathcal{Q} on ei-dominoitujen ratkaisuiden joukko, $d_i = \min_{k \in \mathcal{Q} \wedge k \neq i} \sum_{m=1}^M |f_m^i - f_m^k|$ ja \bar{d} on etäisyyden keskiarvo $\bar{d} = \sum_{i=1}^{|\mathcal{Q}|} d_i / |\mathcal{Q}|$. Etäisyys d_i on siten pienin i :nnen ja minkä tahansa

muun ei-dominoidun ratkaisun välisen tavoitefunktioiden arvojen erotuksen summa. Mittari mittaa eri d_i :n arvojen keskihajontoja, ja mitä pienempi sen arvo on, sitä tasaisemmin ratkaisut ovat jakautuneet. Siten pienempi arvo tarkoittaa parempaa ratkaisua.

Mittari tuottaa hyödyllistä tietoa saavutettujen ei-dominoitujen ratkaisuiden jakautumisesta. Spacing-mittarin aikakompleksisuus on luokkaa $\mathcal{O}(|Q|^2)$. [Deb01]

2.4.2 Spread

Deb *et al.* [Deb02] kehittivät monimuotoisuuden mittaamiseksi menetelmän [Deb01]:

$$spread = \frac{\sum_{m=1}^M d_m^e + \sum_{i=1}^{|Q|} |d_i - \bar{d}|}{\sum_{m=1}^M d_m^e + |Q|\bar{d}}, \quad (14)$$

jossa d_m^e on m :nnen tavoitefunktion Q :n etäisimmän pisteen Euklidinen etäisyys todellisen Pareto-optimaalisen rintaman etäisimpään pisteeseen, M on tavoitefunktioiden lukumäärä, d_i on Q :n peräkkäisten yksilöiden välinen Euklidinen etäisyys, ja \bar{d} on kaikkien d_i :den keskiarvo. Mittarin pieni arvo tarkoittaa suurta monimuotoisuutta.

Spread-mittarilla voidaan mitata vain korkeintaan kaksitavoiteongelmien monimuotoisuutta, joka rajoittaa mittarin käyttöä. Lisäksi sen laskemiseksi tulee tietää todellisen Pareto-optimaalisen rintaman äärimmäisimmät pisteet. [Deb01]

2.4.3 Maximum spread

Zitzler esitteli väitöskirjassaan [Zit99] ei-dominoidun joukon ääripisteiden lävistäjän pituuteen perustuvan tavan mitata monimuotoisuutta [Deb01]:

$$maximum\ spread = \sqrt{\sum_{m=1}^M \left(\max_{i=1}^{|Q|} f_m^i - \min_{i=1}^{|Q|} f_m^i \right)^2}. \quad (15)$$

Kahden tavoitefunktion tapauksessa mittari vastaa ei-dominoidun joukon kahden toisistaan etäisimmän pisteen välistä Euklidista etäisyyttä. Deb esitti kirjassaan [Deb01] myös normalisoidun version kyseisestä mittarista:

$$\overline{\text{maximum spread}} = \sqrt{\frac{1}{M} \sum_{m=1}^M \left(\frac{\max_{i=1}^{|\mathcal{Q}|} f_m^i - \min_{i=1}^{|\mathcal{Q}|} f_m^i}{F_m^{\max} - F_m^{\min}} \right)^2}, \quad (16)$$

jossa F_m^{\max} ja F_m^{\min} ovat todellisen Pareto-optimaalisen rintaman m :nnen tavoitteen maksimi- ja minimiarvot. Kumpikaan mittareista ei ota kantaa ääripisteiden välissä olevien pisteiden jakautumiseen. [Deb01]

3 DIFFERENTIAALIEVOLUUTIOALGORITMI

Rainer Storn:in ja Kenneth Price:n kehittämä differentiaalievoluutioalgoritmi [Sto95] on eräs uusimmista evoluutiopohjaisista menetelmistä. Kyseisen algoritmin toteutus ja käyttö on yksinkertaista, ja se tarvitsee vain vähän parametreja toimiakseen:

- D dimensioiden lukumäärä eli kromosomien lukumäärä yksilössä, määräytyy käytettävän ongelman mukaan,
- NP populaation koko,
- CR risteytysvakio, todennäköisyys jolla kromosomi otetaan yritevektoriin kohinavektorista,
- F mutaatiovakio, käytetään erotusvektorin painottamiseen.

Edellisten lisäksi algoritmin sovelluskohtainen lopetusehto vaatii useimmiten ainakin yhden käyttäjän asettaman parametrin.

Differentiaalievoluutiossa populaatio (Population) \mathbf{P}_G muodostuu yksilöistä (Individual) $\mathbf{X}_{i,G}$, jotka ovat vaihtoehtoisia ratkaisuja ongelmaan. Populaation yksilöt ovat reaaliarvovektoreita, joiden kukin parametri $x_{j,i,G}$ vastaa yksilön yhtä kromosomia. Differentiaalievoluutioalgoritmissa arvoja käsitellään koko suorituksen ajan liukulukuina. Populaation koko pysyy vakiona, ja se koostuu NP :stä reaaliarvoisesta vektorista $\mathbf{X}_{i,G}$, jossa i tarkoittaa populaation jäsentä ja G sukupolvea:

$$\mathbf{P}_G = \mathbf{X}_{i,G} \quad i = 1, \dots, NP, \quad G = 1, \dots, G_{\max} \quad . \quad (17)$$

Jokainen vektori sisältää D reaaliarvoista parametria, kromosomia:

$$\mathbf{X}_{i,G} = x_{j,i,G} \quad i = 1, \dots, NP, \quad j = 1, \dots, D \quad . \quad (18)$$

Kullekin yksilölle lasketaan tavoitefunktion $f(\mathbf{X})$ avulla tavoitefunktion arvo, joka kertoo yksilön hyvyyden. Yksilön kromosomeja optimoimalla saadaan minimoitua

tavoitefunktion arvoa. Evoluutiota jatketaan kunnes saavutetaan riittävän hyvä ratkaisu, ennalta asetettu sukupolvien maksimimäärä tai muu lopetusehto täyttyy. [Lam01]

Differentiaalievoluutioalgoritmin toiminta selviää kuvasta 2. Käytetty differentiaalievoluutioalgoritmin versio on DE/rand/1/bin,

jossa

rand tarkoittaa, että mutaation kohde valitaan satunnaisesti,

1 on erotusvektoreiden lukumäärä,

bin viittaa käytettyyn risteytystapaan. [Sto97]

3.1 Populaation alustus

Ennen varsinaisen optimoinnin aloittamista tulee populaatio alustaa. Alkupuolaatio $P_{G=0}$ muodostetaan yleensä satunnaisesti, sillä tavallisesti ei ole muuta tietoa optimin sijainnista kuin ongelman muuttujien rajat.

$$P_0 = x_{j,i,0} = rand_j[0,1] \cdot (x_j^{(U)} - x_j^{(L)}) + x_j^{(L)} \quad i = 1, \dots, NP, \quad j = 1, \dots, D \quad , \quad (19)$$

jossa $x_j^{(U)}$ ja $x_j^{(L)}$ ovat ylä- ja alaraja muuttujalle x_j ja $rand_j[0,1]$ tarkoittaa tasaisesti jakautunutta satunnaislukua välillä $[0,0,1,0]$ jokaiselle j :lle. [Lam01]

3.2 Mutaatio ja risteytys

Ensimmäisestä sukupolvesta lähtien nykyisen populaation P_G vektoreita valitaan ja yhdistellään satunnaisesti, ja näin saadaan yritektoreita seuraavaa sukupolvea P_{G+1} varten. Aluksi valitaan kohdevektori $X_{i,G}$, joka on ensimmäisellä kierroksella populaation ensimmäinen jäsen. Kohdevektorin lisäksi valitaan populaatiosta satunnaisesti kaksi vektoria $X_{r1,G}$ ja $X_{r2,G}$, joista muodostetaan erotusvektori. Mutaatiota varten valitaan vielä kolmas satunnainen vektori $X_{r3,G}$. Erotusvektori kerrotaan mutaatiovakiolla F , ja näin

painotettu erotusvektori summataan mutaatiota varten valitun vektorin $\mathbf{X}_{r_3,G}$ kanssa. Saatua vektoria kutsutaan kohinavektoriksi, ja risteytys tapahtuu valitsemalla kukin kromosomi todennäköisyydellä CR kohinavektorista yritevektoriin $\mathbf{U}_{i,G+1}$. Yritevektori muodostuu siis kohde- ja kohinavektoreiden kromosomeista. Yritepopulaatio $\mathbf{P}'_{G+1} = \mathbf{U}_{i,G+1} = \mathbf{u}_{j,i,G+1}$ muodostetaan seuraavasti:

$$\mathbf{u}_{j,i,G+1} = \begin{cases} v_{j,i,G+1} = x_{j,r_3,G} + F \cdot (x_{j,r_1,G} - x_{j,r_2,G}) & \text{jos } \text{rand}_j[0,1] \leq CR \vee j = k \\ x_{j,i,G} & \text{muutoin} \end{cases}, \quad (20)$$

jossa

$$i = 1, \dots, NP, \quad j = 1, \dots, D,$$

$k \in \{1, \dots, D\}$, satunnainen parametrin indeksi, valitaan erikseen jokaiselle i :lle

$r_1, r_2, r_3 \in \{1, \dots, NP\}$, valitaan satunnaisesti siten, että: $r_1 \neq r_2 \neq r_3 \neq i$,

$CR \in [0,1]$, $F \in (0,1+]$, vakioita kontrolliparametreja.

Populaatio käydään järjestyksessä läpi kunnes kaikki populaation yksilöt ovat olleet vuorollaan kohdevektorina. [Lam01]

3.3 Valinta

Yritevektorille lasketaan tavoitefunktion arvo, ja sitä verrataan kohdevektorin tavoitefunktion arvoon. Seuraavan sukupolven populaatio \mathbf{P}_{G+1} valitaan nykyisestä populaatiosta \mathbf{P}_G ja yritepopulaatiosta \mathbf{P}'_{G+1} seuraavan valintasäännön mukaan:

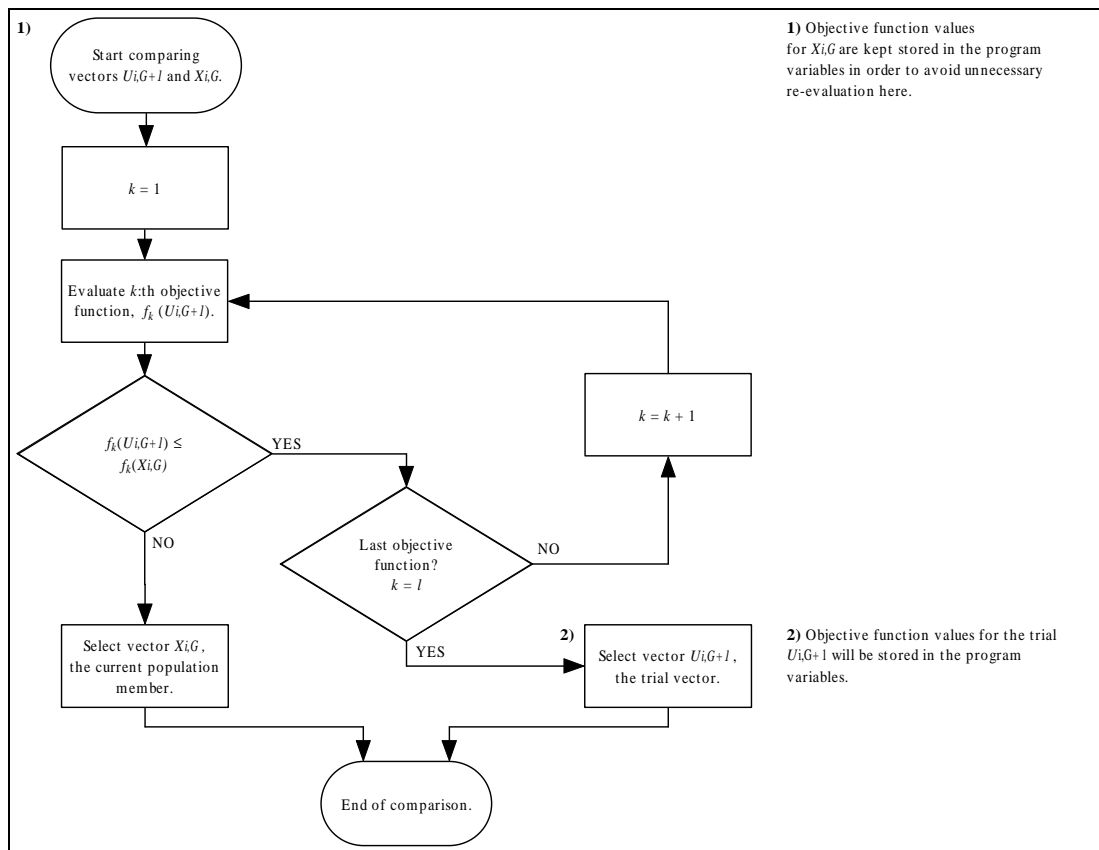
$$\mathbf{X}_{i,G+1} = \begin{cases} \mathbf{U}_{i,G+1} = \mathbf{u}_{j,i,G+1} & \text{jos } f(\mathbf{U}_{i,G+1}) \leq f(\mathbf{X}_{i,G}) \\ \mathbf{X}_{i,G} & \text{muutoin} \end{cases}. \quad (21)$$

Jokaista yritepopulaation yksilöä verrataan sitä vastaavan nykyisen populaation yksilön kanssa. Yritevektoria verrataan vain yhteen populaation jäseneseen. Kaikki seuraavan sukupolven yksilöt ovat yhtä hyviä tai parempia kuin vastaavat yksilöt nykyisessä sukupolvessa. [Lam01]

Monitavoiteoptimoinnin tapauksessa valintamenettely eroaa edellä esitetystä, ja se perustuu kappaleessa 2.2 määriteltyyn Pareto-dominanssin käsitteeseen:

$$\mathbf{X}_{i,G+1} = \begin{cases} \mathbf{U}_{i,G+1} & \text{jos } \forall k \in \{1, \dots, l\}: f_k(\mathbf{U}_{i,G+1}) \leq f_k(\mathbf{X}_{i,G}) \\ \mathbf{X}_{i,G} & \text{muutoin} \end{cases}, \quad (22)$$

jossa l on tavoitefunktioiden lukumäärä ja k tavoitefunktion indeksi. Yritektori valitaan siis jatkoon, jos sen kaikkien tavoitefunktioiden arvot ovat paremmat tai yhtä hyvät kuin vastaavan nykyisen populaation jäsenen arvot. [Lam01a]



Kuva 1. Pareto-optimointiin perustuva valintamenettely. [Lam01a]

3.4 Rajoitteet

Mutaation ja risteytyksen jälkeen on oleellista tarkistaa, että tuotettujen kromosomien arvot ovat sallitulla alueella. Parametreille on yleensä määrätty ylä- ja alarajat $x^{(U)}$ ja $x^{(L)}$, joiden

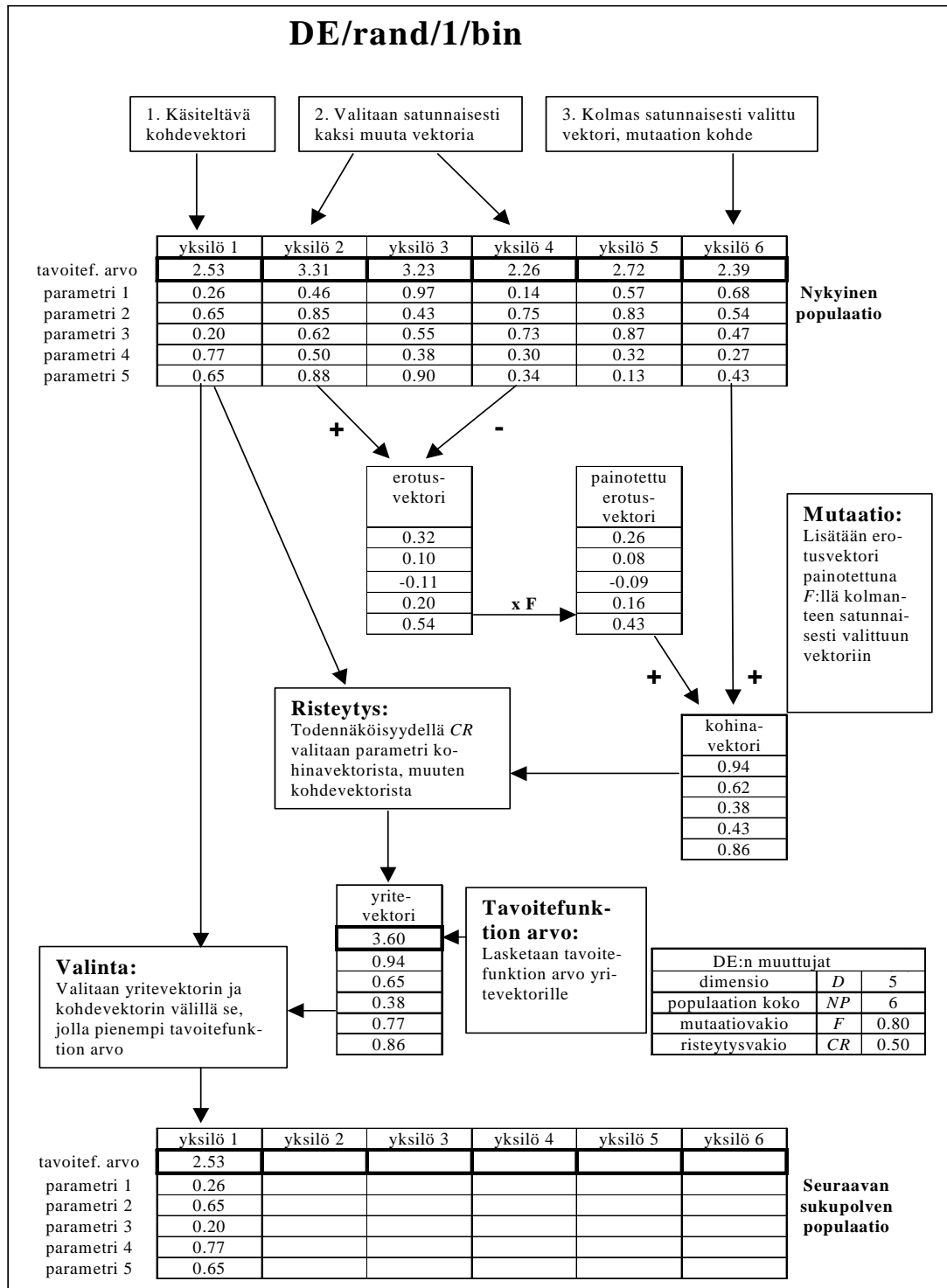
välissä kromosomin arvon tulee olla. Näitä rajoja kutsutaan usein laatikkorajoitteiksi. Rajoitteita rikkovat kromosomit voidaan esimerkiksi korvata satunnaisella rajoitteiden mukaisella arvolla:

$$u_{j,i,G+1} = \begin{cases} \text{rand}_j[0,1] \cdot (x_j^{(U)} - x_j^{(L)}) + x_j^{(L)} & \text{jos } u_{j,i,G+1} < x_j^{(L)} \vee u_{j,i,G+1} > x_j^{(U)} \\ u_{j,i,G+1} & \text{muutoin} \end{cases}, \quad (23)$$

jossa

$$i = 1, \dots, NP, \quad j = 1, \dots, D.$$

Laatikkorajoitteiden lisäksi ongelmassa voi olla myös rajoitefunktioita, joiden käsittelyyn on olemassa useita menetelmiä [Lam01b]. [Lam01]



Kuva 2. Differentiaalievoluutioalgoritmin toiminta. Tavoitefunktiona on esimerkissä käytetty $f(X) = x_1 + x_2 + x_3 + x_4 + x_5$. [Lam01]

4 MONIMUOTOISUUTTA MITTAAVIEN MITTAREIDEN KEHITTÄMINEN

Populaation monimuotoisuuden mittaamiseen kehitetyt menetelmät ovat usein laskennallisesti liian raskaita käytännön sovellutuksissa käytettäväksi. Tämän työn tarkoituksena on kehittää käyttökelpoinen menetelmä hakuavaruuden populaation absoluuttisen ja suhteellisen monimuotoisuuden mittaamiseksi liukulukukoodatuissa evoluutioalgoritmeissa. Tavoitteena on tehdä mittarista laskennallisesti kevyt, jotta se olisi käyttökelpoinen käytettäessä myös suuria populaatioita. Lisäksi mittarin tulisi olla yleiskäyttöinen, jotta sitä voitaisiin hyödyntää laajasti. Myös mittareiden antamien tulosten tulisi olla vertailukelpoisia eri ongelmilla.

4.1 Alkupuolaation monimuotoisuus

Evoluutioalgoritmeissa alkupuolaatio alustetaan tavallisesti tasaisen jakauman satunnaisluvuilla. Tällöin voidaan ajatella populaation olevan tasaisesti jakautunut hakuavaruudessa, ja populaation yhden parametrin j keskihajonta A_j voidaan estimoida yhtälöllä:

$$A_j = \frac{x_j^{(U)} - x_j^{(L)}}{\sqrt{12}} , \quad (24)$$

jossa $x_j^{(U)}$ ja $x_j^{(L)}$ ovat ylä- ja alaraja, joiden välistä kyseinen parametri voi saada arvoja populaatiota alustettaessa. Yhtälöstä (24) nähdään, ettei yksittäisen parametrin keskihajonta riipu pisteiden lukumäärästä, eli populaation koosta NP .

Eri parametrien saamat arvot voivat siis vaihdella välillä $x_j^{(U)}$ ja $x_j^{(L)}$, joten yksittäisen parametrin j keskihajonta tulee skaalata keskihajontojen vaikutuksen tasaamiseksi. Skaalaus voidaan tehdä esimerkiksi jakamalla keskihajonta A_j kyseisen parametrin suurimman ja pienimmän arvon erotuksella. Tasaisesti jakautuneen alkupuolaation

tapauksessa voidaan ajatella parametrin suurimman ja pienimmän arvon olevan samat kuin parametrin rajat $x_j^{(U)}$ ja $x_j^{(L)}$. Täten skaalaukselle saadaan:

$$A'_j = \frac{A_j}{x_j^{(U)} - x_j^{(L)}} \quad (25)$$

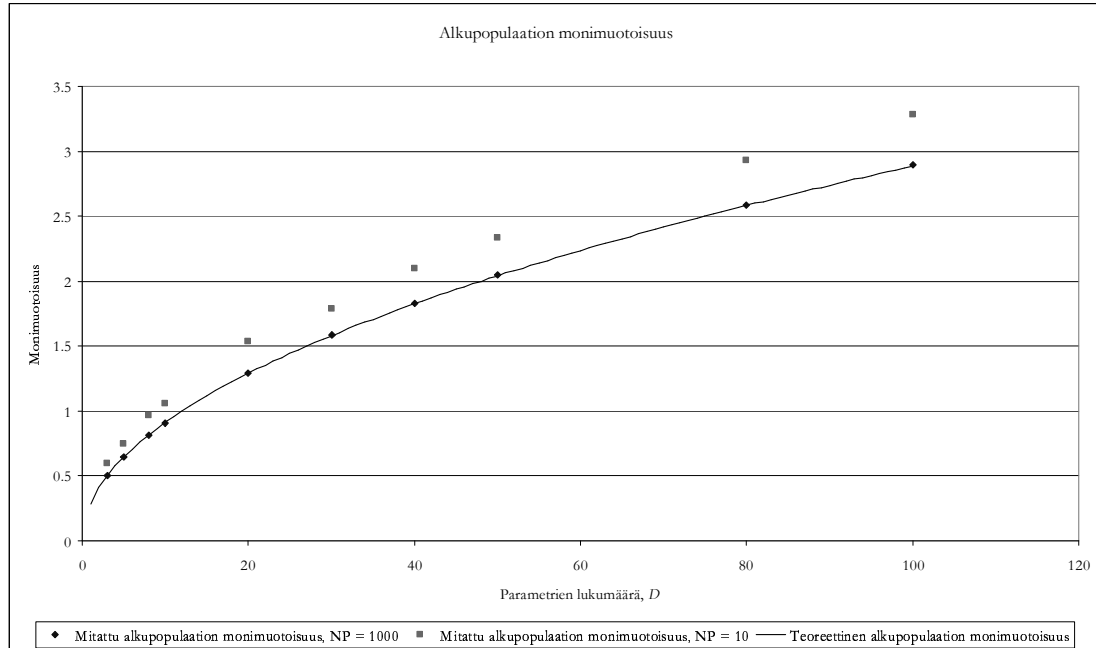
Sijoittamalla yhtälö (24) yhtälöön (25) saadaan alkupopulaation yhden parametrin skaalatuksi keskihajonnaksi

$$A'_j = \frac{1}{\sqrt{12}} \quad (26)$$

Yhdistämällä kaikkien parametrien skaalatut keskihajonnat neliöön korottamalla, summaamalla, ja ottamalla neliöjuuri saadusta arvosta, saadaan alkupopulaation monimuotoisuus

$$\Delta_{\text{alkupopulaatio}} = \sum_{j=1}^D \sqrt{A_j'^2} = \sqrt{D \cdot \left(\frac{1}{\sqrt{12}}\right)^2} = \frac{\sqrt{D}}{\sqrt{12}} \quad (27)$$

Yhtälöstä (27) nähdään, että näin määritelty alkupopulaation monimuotoisuus riippuu vain parametrien lukumäärästä D . Käytetyllä populaation koolla NP ei siten pitäisi olla merkitystä alkupopulaation monimuotoisuuteen. Mittaamalla alkupopulaation monimuotoisuutta käyttämällä useita parametrien arvoja ja populaation kokoja, ja vertaamalla saatuja arvoja teoreettiseen alkupopulaation monimuotoisuuden arvoon, havaittiin populaation koon kuitenkin vaikuttavan alkupopulaation monimuotoisuuteen, kuten kuvasta 3 nähdään. Vaikutus on suurin pieniä populaation kokoja käytettäessä, ja suurilla populaatioilla käyttäytyminen noudattaa teoreettista mallia. Pienillä populaatioilla, kun $NP < 50$, mitatut alkupopulaation monimuotoisuudet olivat korkeampia kuin teoreettiset monimuotoisuudet. Luultavimpana syynä tähän on se, että pieniä populaatioita käytettäessä yksilöt eivät populaatiota alustettaessa jakaudu tasaisesti hakuavaruuteen, vaan hakualueen reunoille jää tyhjiä alueita. Suurella populaatiolla on taas todennäköistä, että yksilöitä osuu alustettaessa myös hakuavaruuden reunoille.

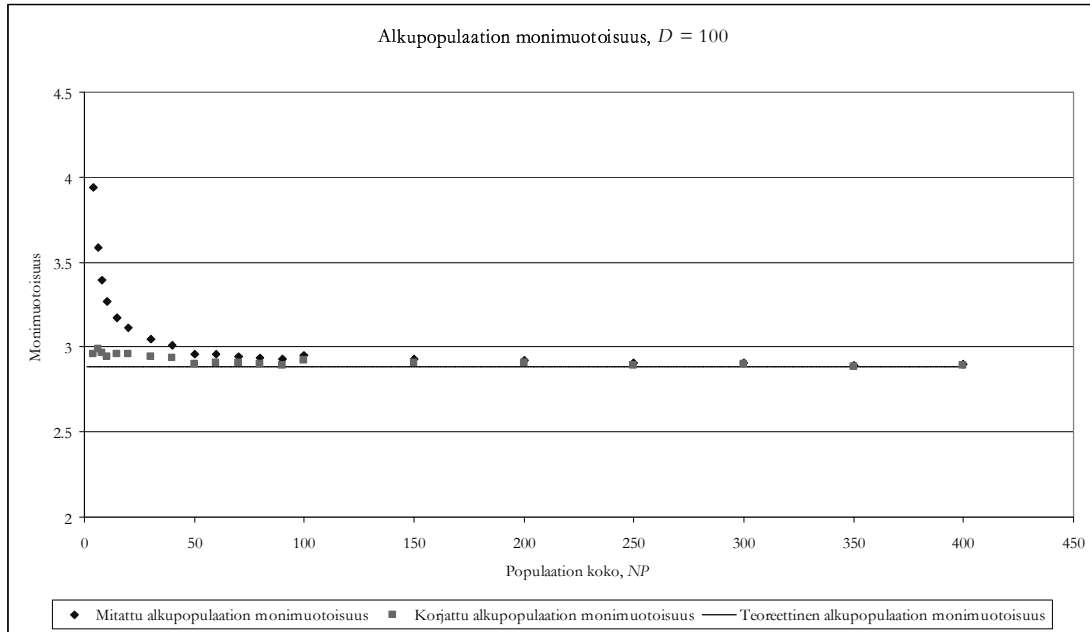


Kuva 3. Alkupopulaation monimuotoisuus parametrien lukumäärän funktiona.

Jotta myös pieniä populaation kokoja käytettäessä alkupopulaation monimuotoisuus noudattaisi teoreettista mallia, voidaan sille tehdä korjaus kertomalla se sopivalla kertoimella:

$$\Delta'_{alkupopulaatio} = \frac{NP - 1}{NP} \cdot \Delta_{alkupopulaatio} \quad (28)$$

Koska korjauskertoimen vaikutus on suuri pienillä NP :n arvoilla ja pieni suurilla populaatioita käytettäessä, voidaan sitä käyttää kaikilla NP :n arvoilla. Kuvassa 4 on esimerkki korjauskertoimen vaikutuksesta alkupopulaation monimuotoisuuden arvoon eri kokoisilla populaatioilla, kun parametrien lukumääränä on 100.



Kuva 4. Alkupopulaation monimuotoisuus populaation koon funktiona, parametrien lukumäärän ollessa 100.

4.2 Suhteellinen ja absoluuttinen monimuotoisuus

Suhteellisen ja absoluuttisen monimuotoisuuden mittarit perustuvat keskihajontojen laskemiseen parametreittain populaatiosta. Populaatio koostuu NP yksilöstä, joista jokaisessa on D parametria. Populaatiota voidaan siten pitää $NP \cdot D$ matriisina, jossa NP on sarakkeiden lukumäärä ja D rivien lukumäärä.

Alkupopulaation oletettiin edellä olevan tasaisesti jakautunut, ja sen keskihajontaa estimoitiin yhtälöllä (24). Optimoinnin edetessä populaatio suppenee kohti optimin tuottavia arvoja. Tällöin keskihajontojen laskemiseen ei voida enää käyttää tasajakauman keskihajonnan odotusarvoa, vaan on käytettävä todellista populaatiosta laskettua normaalijakauman keskihajontaa. Keskihajonnat lasketaan populaation jokaiselle riville

$$A_j = \sqrt{\frac{\sum_{i=1}^{NP} (x_{j,i} - \bar{x}_j)^2}{NP}} \quad j = 1, \dots, D \quad , \quad (29)$$

jossa \bar{x}_j on populaation j :n parametririvin arvojen keskiarvo. Kun keskihajonnat on laskettu kaikille parametririveille, saadaan niistä D -paikkainen vektori A , joka pitää sisällään kaikki keskihajonnat.

Saadut keskihajonnat on skaalattava, sillä niiden suuruusluokka voi vaihdella suuresti, ja siten niiden vaikutus lopputulokseen olisi eri suuruinen. Valittava skaalaustapa vaikuttaa mittarin antamaan arvoon, ja juuri skaalaus on erona suhteellisen ja absoluuttisen monimuotoisuuden laskemisessa. Suhteellista monimuotoisuutta laskettaessa kunkin parametririvin keskihajonta jaetaan saman populaation kyseisen parametririvin suurimman ja pienimmän arvon erotuksella. Jokainen keskihajonta on siis skaalattu sitä vastaavan rivin suurimmalla mahdollisella hajonnalla. Skaalauksessa on lisäksi otettava huomioon, että A_j :n laskemisessa käytetty jakaja saattaa joissakin tapauksissa saada arvon 0. Koska 0:lla jakamista ei ole määritelty, asetetaan jakajan tullessa 0:ksi kyseinen skaalattu keskihajonta A_j 0:ksi:

$$\begin{cases} A_j' = \frac{A_j}{\max(x_j) - \min(x_j)} & j = 1, \dots, D, \text{ jos } \max(x_j) \neq \min(x_j) \\ A_j' = 0 & j = 1, \dots, D, \text{ jos } \max(x_j) = \min(x_j) \end{cases} \quad (30)$$

Vektori A' sisältää siten skaalatut keskihajonnat. Vektorin A' elementit voidaan yhdistää yhdeksi luvuksi ottamalla siitä L2-normi. Tämä tapahtuu korottamalla vektorin elementit A_j' neliöön, summaamalla ne, ja ottamalla neliöjuuri:

$$A_R' = \|A_j'\| = \sqrt{A_1'^2 + A_2'^2 + \dots + A_D'^2} = \sqrt{\sum_{j=1}^D A_j'^2} \quad (31)$$

Alkupopulaation monimuotoisuutta tutkittaessa havaittiin parametrien D lukumäärän vaikuttavan sen monimuotoisuuteen. Jotta suhteellisen monimuotoisuuden arvot olisivat vertailukelpoisia käytettäessä eri D :n arvoja, suoritetaan sille yhtälön (27) mukaisesti korjaus:

$$\Delta_R = \frac{\sqrt{12}}{\sqrt{D}} \cdot \Delta_R' \quad . \quad (32)$$

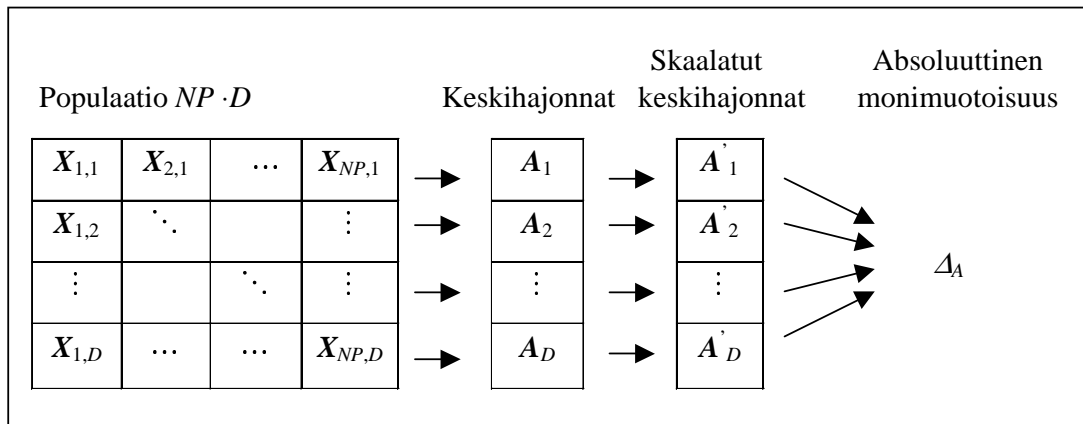
Saatu suhteellisen monimuotoisuuden mittari antaa siten tasaisesti jakautuneelle alkupopulaatiolle arvon 1 kaikilla D :n arvoilla. Koska kyse on estimaatista, on alkupopulaation suhteellinen monimuotoisuus käytännössä hieman yli tai alle arvon 1.

Absoluuttisen monimuotoisuuden laskeminen poikkeaa suhteellisen monimuotoisuuden laskemisesta lähinnä skaalaustavan perusteella. Absoluuttisessa monimuotoisuudessa yhtälöllä (29) lasketut keskihajonnat jaetaan alkupopulaation kunkin parametrin suurimman ja pienimmän arvon erotuksella. Siten jakaja pysyy samana kaikissa sukupolvissa parametreittain:

$$A_j'' = \frac{A_j}{\max(x_{j,G=1}) - \min(x_{j,G=1})} \quad j = 1, \dots, D \quad . \quad (33)$$

Saadut skaalatut keskihajonnat yhdistetään ottamalla vektorista A'' L2-normi, kuten suhteellistakin monimuotoisuutta laskettaessa:

$$\Delta_A = \|A''\| = \sqrt{A_1''^2 + A_2''^2 + \dots + A_D''^2} = \sqrt{\sum_{j=1}^D A_j''^2} \quad . \quad (34)$$



Kuva 5. Absoluuttisen monimuotoisuuden laskeminen.

Suhteellinen ja absoluuttinen monimuotoisuus lasketaan jokaiselle sukupolvelle G . Molempien arvojen laskemisessa käytetään samoja keskihajontoja, ja vain mittareiden skaalaustapa eroaa toisistaan. Siten molemmat monimuotoisuudet jokaiselle sukupolvelle laskevan algoritmin aikakompleksisuus on luokkaa

$$O(NP \cdot D \cdot G) .$$

Suhteellisen ja absoluuttisen monimuotoisuuden laskemiseen tarvittava aika kasvaa lineaarisesti populaation koon kasvaessa. Se on siten huomattavasti parempi kuin Šmuc:in esittämän monimuotoisuusmittarin eksponentiaalinen kompleksisuus ja samaa luokkaa Lopez Cruz *et al.* :n menetelmän kanssa. Monimuotoisuuksien laskeminen on nopeaa, ja mittari sopii käytettäväksi myös suuria populaatioita vaativien ongelmien kanssa.

5 MITTAREIDEN TOIMINNAN TARKASTELO

Mittareiden toimintaa tarkasteltiin ratkaisemalla tunnettuja testiongelmia differentiaalievoluutioalgoritmeilla, ja mittaamalla suhteellista ja absoluuttista monimuotoisuutta ongelmanratkaisun aikana. Mittarit laskevat suhteellisen ja absoluuttisen monimuotoisuuden jokaisen sukupolven populaatiolle, joten kyseisten arvojen kehitystä on mahdollista seurata ja arvioida optimoinnin etenemisen suhteen.

5.1 Testiongelmät

Menetelmän toiminnan testaamista varten valittiin hyvin tunnettuja ja paljon käytettyjä testiongelmia. Tunnetuin geneettisten algoritmien testaamiseen käytetty testisarja on De Jongin testisarja [DeJ75], joka koostuu viidestä ongelmasta. De Jongin sarjasta valittiin käytettäväksi kahta sen ensimmäistä ongelmaa, Sphere:ä ja Rosenbrock:ia. Lisäksi valittiin useita muita tarkoitukseen sopivia testiongelmia, jotka esitellään myöhemmin tässä kappaleessa.

Testifunktioiden ominaisuudet vaikuttavat suuresti optimointialgoritmien kykyyn löytää globaali optimi. Tiedetyt ominaisuudet voivat aiheuttaa joillekin algoritmeille suuria vaikeuksia, vaikka toisille algoritmeille niistä ei ole haittaa. Funktioiden tärkeimmät ominaisuudet ovat:

Jatkuvuus ja epäjatkuvuus:

Funktio on jatkuva, mikäli sen kuvaaja on yhtenäinen käyrä tarkasteltavana olevalla alueella. Funktio on siis oltava jatkuva jokaisessa tarkasteltavan välin pisteessä.

Kuperuus ja koveruus:

Käyrä on kupera, mikäli sen jokaiseen pisteeseen piirretty tangentti on kokonaan käyrän alapuolella. Vastaavasti käyrä on kovera, jos sen jokaiseen pisteeseen piirretty tangentti on kokonaan käyrän yläpuolella.

Yksihuippisuus (unimodaalisuus) ja monihuippisuus (multimodaalisuus):

Funktio on yksihuippuinen, mikäli sillä ei ole globaalin optimin lisäksi lokaaleja optimeita.

Monihuippuisella funktiolla on globaalin optimin lisäksi vähintään yksi lokaali optimi.

Skaalautuvuus:

Funktion sanotaan olevan skaalautuva, mikäli funktio soveltuu käytettäväksi millä tahansa määrällä muuttujia, eli parametrien määrällä D .

Separoituvuus:

Funktio on separoituva, jos se voidaan esittää yhden muuttujan funktioiden summana.

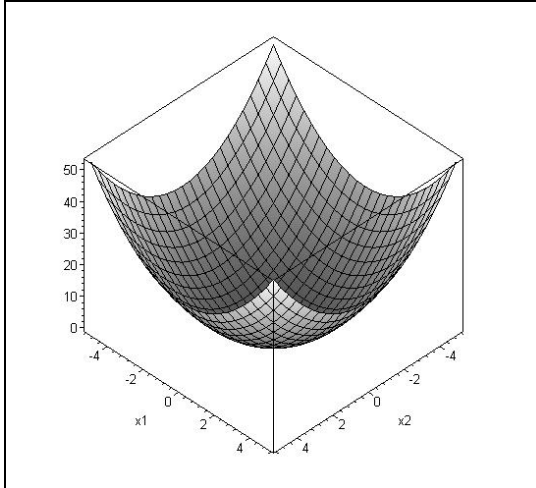
5.1.1 Testifunktio Sphere

Sphere on De Jong'in testisarjan ensimmäinen ja helpoin funktio. Funktio on symmetrinen, skaalautuva, jatkuva, konvekssi ja yksihuippuinen. Yksinkertaisuutensa vuoksi funktio soveltuu myös käytettävän algoritmin yleisen tehokkuuden arviointiin. [Yur94]

$$f(x_i |_{i=1,N}) = \sum_{i=1}^N x_i^2 \quad (35)$$

$$x_i \in [-5.12, 5.12]$$

Funktion globaali minimiarvo on 0, ja se sijaitsee origossa.



Kuva 6. Testifunktio Sphere.

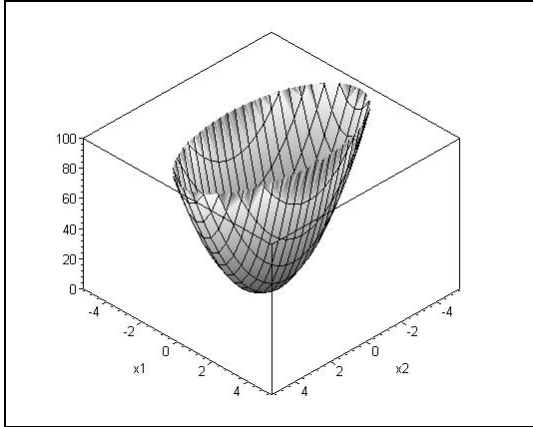
5.1.2 Testifunktio Ellipsoid 1

Testifunktio Ellipsoid 1 on edellisen testifunktion variaatio. Se on koordinaatiston akseleiden suuntainen ellipsoidi, jossa parametrikohtainen mutaatioaskeleen säätäminen on tarpeen. Funktion muuttujat ovat riippumattomia toisistaan.

$$f(x_i |_{i=1,2}) = (x_1 + 3 \cdot x_2)^2 + (x_2 - 3 \cdot x_1)^2 \quad (36)$$

$$x_i \in [-5.12, 5.12]$$

Ellipsoid 1:n globaali minimiarvo on 0, ja se sijaitsee origossa.



Kuva 7. Testifunktio Ellipsoid 1.

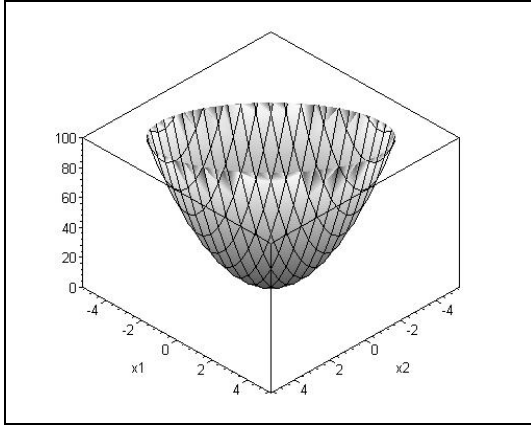
5.1.3 Testifunktio Ellipsoid 2

Funktio Ellipsoid 2 on kuten Ellipsoid 1, mutta ellipsoidia on kierretty koordinaattiakselien suhteen. Tämä aiheuttaa sen, että funktion muuttujat ovat riippuvaisia toisistaan. Muuttujien riippuvuuden vuoksi ei riitä, että hakua suoritetaan vain koordinaattiakselien suuntaisesti, vaan haun on kyettävä etenemään kaikkiin suuntiin. Tällaista kaikkiin suuntiin etenevää hakua kutsutaan rotaatioinvariantiksi.

$$f(x_i |_{i=1,2}) = (x_1 + 3 \cdot x_2)^2 + (x_2 + 3 \cdot x_1)^2 \quad (37)$$

$$x_i \in [-5.12, 5.12]$$

Globaali minimiarvo funktiolle Ellipsoid 2 on 0, ja se sijaitsee origossa.



Kuva 8. Testifunktio Ellipsoid 2.

5.1.4 Testifunktio Rosenbrock

Funktio Rosenbrock on klassinen optimointiongelma. Globaali optimi sijaitsee pitkässä, kapeassa parabolin muotoisessa laaksossa. Laakson löytäminen on helppoa, mutta suppeneminen globaaliin optimiin on vaikeaa. Jotta globaali optimi löydetään, on algoritmin kyettävä hakemaan kaikkiin mahdollisiin suuntiin, ei vain pelkästään koordinaattiakselien suuntaisesti. Kaikkiin suuntiin yhtä tehokkaasti hakua suorittavaa menetelmää kutsutaan rotaatioinvariantiksi, koska ongelman kiertäminen koordinaattiakselien suhteen ei vaikeuta kyseisellä menetelmällä globaalin optimin löytämistä. Rosenbrock:ista on olemassa kahden muuttujan versio (Rosenbrock) ja yleistetty skaalautuva versio (Generalized Rosenbrock).

Rosenbrock:

$$f(x_i |_{i=1,2}) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2 \quad (38)$$

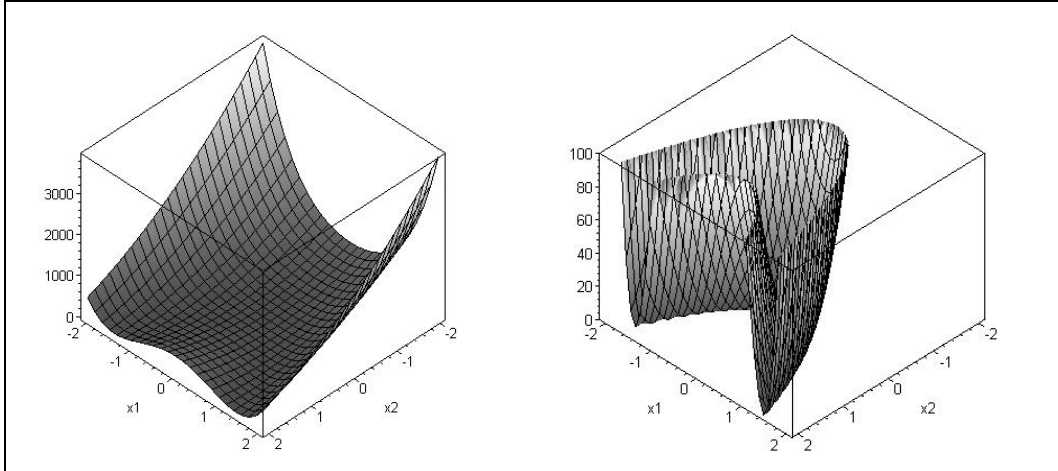
$$x_i \in [-2.048, 2.047]$$

Generalized Rosenbrock:

$$f(x_i |_{i=1,N}) = \sum_{i=1}^{N-1} (100 \cdot (x_{i+1} - x_i^2)^2 + (x_i - 1)^2) \quad (39)$$

$$x_i \in [-5.12, 5.12]$$

Kummankin funktion globaali minimiarvo on 0, ja se saavutetaan parametrien arvoilla $x_i = 1$. [Whi96], [Yur94]



Kuva 9. Testifunktio Rosenbrock.

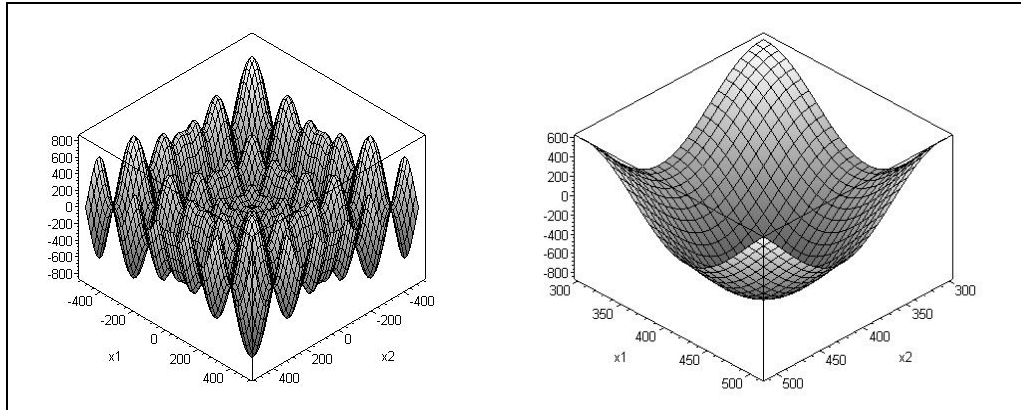
5.1.5 Testifunktio Schwefel

Schwefel'in funktio on petollinen, sillä globaali minimi sijaitsee hakuavaruuden suhteen etäällä seuraavaksi parhaasta lokaalista minimistä. Niinpä vaarana saattaa olla suppeneminen väärään suuntaan. Funktio on skaalautuva. [Poh94]

$$f(x_i |_{i=1,N}) = \sum -x_i \sin(\sqrt{|x_i|}) \quad (40)$$

$$x_i \in [-512, 511]$$

Globaali minimi sijaitsee pisteessä $x_i = 420,9687$, ja funktion globaali minimiarvo riippuu käytetystä parametrien määrästä ollen $-418,9829 \cdot N$. [Whi96]



Kuva 10. Testifunktio Schwefel.

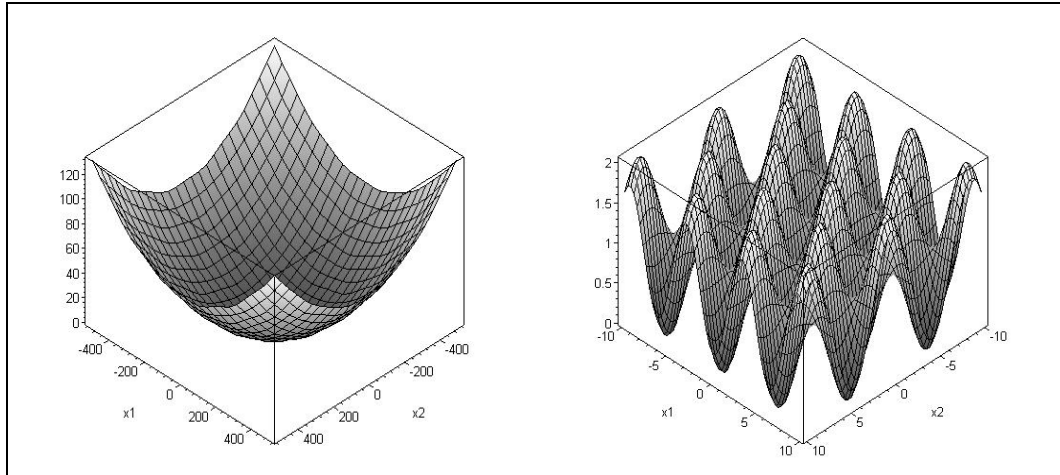
5.1.6 Testifunktio Griewank

Griewank:in funktio on skaalautuva, ja sillä on useita laajalle levinneitä lokaaleja minimipisteitä.

$$f(x_i |_{i=1,N}) = \sum_{i=1}^N \frac{x_i^2}{4000} - \prod_{i=1}^N \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (41)$$

$$x_i \in [-512, 511]$$

Griewank:in funktion globaali minimiarvo on 0, ja se sijaitsee origossa. Funktion lokaalien minimien lukumäärä kasvaa eksponentiaalisesti dimensioiden lisääntyessä. Tiettyjä algoritmeja käytettäessä globaalin minimin löytäminen tulee helpommaksi dimensioiden lisääntyessä. [Whi96], [Loc03]



Kuva 11. Testifunktio Griewank.

5.2 Kokeellinen tarkastelu

Kehitettyjen mittareiden toimintaa tarkasteltiin kokeellisesti ratkaisemalla testiongelmia differentiaalievoluutioalgoritmeilla ja mittaamalla monimuotoisuutta. Suoritettujen kokeiden tavoitteena oli saada alustava käsitys mittareiden toiminnasta ja niissä mahdollisesti esiintyvistä ongelmista. Monimuotoisuuden mittarit toteutettiin C-kielisenä funktiona, jota kutsuttiin olemassa olevasta C-kielisestä differentiaalievoluutioalgoritmin ohjelmasta. Valmiina olleeseen differentiaalievoluutio-ohjelmaan jouduttiin tekemään pieniä muutoksia, kuten alkupopulaation parametrien pienimmän ja suurimman arvon etsiminen, sekä suhteellisen ja absoluuttisen monimuotoisuuden tulostaminen tiedostoon ja näytölle. Populaation suhteellista ja absoluuttista monimuotoisuutta mittaavan mittarin C-lähdekielinen ohjelma löytyy liitteestä 1.

Kunkin testiajon tuloksista piirrettiin kolme kuvaajaa, joista ensimmäisessä on suhteellinen ja absoluuttinen monimuotoisuus sukupolvien funktiona, toisessa sama puolilogaritmista asteikkoa käyttäen ja kolmannessa paras tavoitefunktion arvo $f(\mathbf{X})$ sukupolvien funktiona puolilogaritmisella asteikolla. Niiden funktioiden kohdalla, joilla globaalit optimin arvo ei ole 0, käytettiin kolmannessa kuvaajassa parhaan tavoitefunktion arvon ja funktion optimiarvon erotusta $f(\mathbf{X}) - f(\mathbf{X}^*)$ sukupolvien funktiona puolilogaritmisella asteikolla.

Differentiaalievoluution kontrolliparametrit pyrittiin valitsemaan siten, että saadaan helposti ratkeavia tapauksia, sekä vaikeita tapauksia, joissa on ongelmia löytää optimiratkaisu tai optimointi epäonnistuu. Risteytysvakioksi CR valittiin arvo 1,0, ja mutaatiovakiolle F valittiin arvo 0,5. Risteytysvakion CR arvoon 1,0 päädyttiin, koska tällöin differentiaalievoluution haku on rotaatioinvariantti, toisin sanoen haku pääsee tapahtumaan kaikkiin mahdollisiin suuntiin yhtä tehokkaasti. Tästä on hyötyä etenkin ongelmissa, joissa muuttujat ovat riippuvaisia toisistaan. Testauksessa käytetyt parametrien lukumäärän D ja populaation koon NP yhdistelmät käytetyillä testifunktioilla on esitetty taulukossa 1.

Taulukko 1. Testitapaukset testifunktioittain, kaikissa $CR = 1,0$ ja $F = 0,5$.

Testifunktio	Testitapaus 1: $D = 2$ $NP = 40$	Testitapaus 2: $D = 10$ $NP = 100$	Testitapaus 3: $D = 20$ $NP = 50$	Testitapaus 4: $D = 20$ $NP = 100$	Testitapaus 5: $D = 20$ $NP = 200$
Sphere	X	X	X	X	X
Ellipsoid 1	X				
Ellipsoid 2	X				
Rosenbrock	X	X	X	X	X
Schwefel	X	X	X	X	X
Griewank	X	X	X	X	X

Testifunktioita Ellipsoid 1 ja 2 käytettiin vain ensimmäisessä testitapauksessa, koska niiden antamat tulokset osoittautuivat odotetulla tavalla vastaaviksi kuin funktion Sphere, ja siten oli tarpeetonta tarkastella niitä käyttäen muita D :n ja NP :n arvoja. Kaikilla muilla testifunktioilla suoritettiin kaikki testitapaukset. Testitapaukset pyrittiin valitsemaan siten, että kokeellisesta tarkastelusta saatava hyöty olisi mahdollisimman suuri, ja eri testitapauksia voitaisiin verrata keskenään.

6 TULOKSET

Tässä kappaleessa tarkastellaan testiajojen tuloksia, ja analysoidaan keskeisimpiä niistä havaittavia ominaisuuksia. Kuvaajat testiajoista ovat liitteissä 2 - 6.

6.1 Testitapaus 1

Helpoimmalla testitapauksella, parametrien lukumäärän ollessa 2 ja populaation koon ollessa 40, ovat tulokset lähimpänä ideaalista ja tulosten tulkinta melko suoraviivaista. Suhteellinen monimuotoisuus lähtee noin arvosta 1, kuten kuuluukin suhteelliselle monimuotoisuudelle tehdyn skaalauksen vuoksi. Tyypillisesti suhteellinen monimuotoisuus laskee hieman ja pysyy hajonnan puitteissa likimain vakiona, pysytellen noin 0,8:n paikkeilla, kunnes se laskee huomattavasti tai katoaa kokonaan. Muutoksen syynä on laskentatarkkuuden loppuminen.

Testifunktiolla Sphere laskentatarkkuuden loppuessa suhteellinen monimuotoisuus asettuu noin 0,4:ään, kun taas molemmilla Ellipsoid-funktiolla se katoaa kokonaan. Testifunktion Ellipsoid 1 tuloksista (liite 2, kuva 2) nähdään, että funktion venyttäminen koordinaattiakseleiden suunnassa ei vaikuta mittarin toimintaan. Samoin testifunktion Ellipsoid 2 tuloksista (liite 2, kuva 3) huomataan, ettei funktion venyttäminen mielivaltaisessa suunnassa tai funktion kiertäminen koordinaattiakseleiden suhteen vaikuta mittarien antamiin tuloksiin.

Suhteellinen monimuotoisuus saattaa myös kasvaa huomattavasti juuri ennen katoamistaan, kuten esimerkiksi Rosenbrock'in ja Schwefel'in funktioiden kohdalla on käynyt. Lisäksi Schwefel'in funktion kohdalla suhteellinen monimuotoisuus käyttäytyy muista funktioista poiketen, sillä sen arvo lähtee aluksi kasvamaan, jonka jälkeen se kääntyy laskuun ja saavuttaa hetkeksi normaalin tason ja lopuksi kohoaa voimakkaasti ennen katoamistaan. Suhteellisen monimuotoisuuden kasvaminen johtuu luultavasti haun juuttumisesta yhteen tai useampaan lokaaliin optimiin, joita Schwefel'in funktiolla on runsaasti. Ennen suhteellisen monimuotoisuuden katoamista esiintyvä voimakas kasvu johtunee puolestaan globaalien optimien löytymisestä. Griewank'in funktiolla suhteellinen

monimuotoisuus pysyy aluksi noin 0,8:n tasolla, mutta kasvaa siitä hieman, noin 0,9:n tasolle. Tämäkin muutos näyttäisi tapahtuvan samaan aikaan globaalin optimin löytymisen kanssa.

Absoluuttinen monimuotoisuus näyttää ideaalitapauksessa pienenevän logaritmisesti, kunnes se saavuttaa arvon 0. Puolilogaritmisella asteikolle piirrettäessä siitä muodostuu laskeva suora, jonka kulmakerroin kuvaa populaation absoluuttisen monimuotoisuuden pienenemisnopeutta.

Schwefel'in funktiolla absoluuttinen monimuotoisuus nousee aluksi hieman ja lähtee sitten laskemaan, aivan kuten suhteellisellakin monimuotoisuudella tapahtui. Puolilogaritmisella asteikolla Schwefel'in funktiolla absoluuttisen monimuotoisuuden nähdään pysyvän ensin melko tasaisena, sitten laskevan melko jyrkästi hetken aikaa, pysyvän taas tasaisena ja laskevan lopulliselle tasolle. Tasaiset alueet käyrässä johtuvat haun juuttumisesta lokaaleihin optimeihin, kunnes globaali optimi on lopuksi löydetty. Griewank'in funktiolla absoluuttisen monimuotoisuuden käyttäytyminen on lähes samanlaista kuin Schwefel'in funktiolla, mutta täysin tasaisia alueita ei käyrässä ole, vaan käyrä laskee välillä loivemmin ja välillä jyrkemmin. Käyrän loivissa kohdissa haku etenee hitaammin lokaalien optimien tutkimisen vuoksi, ja kohti globaalia optimia supetessa haun nopeus kasvaa.

Tavoitefunktion arvo muodostaa puolilogaritmiselle asteikolle piirrettynä ideaalitapauksessa laskevan suoran, kuten myös absoluuttisen monimuotoisuuden tapauksessa tapahtui. Tämä laskeva suora ilmoittaa haun todellisen konvergoitumisnopeuden. Puolilogaritmiselle asteikolle piirrettyjä absoluuttisen monimuotoisuuden ja tavoitefunktion arvon muodostamia suoria verrattaessa havaitaan, että tavoitefunktion arvoa kuvaava suora laskee hieman jyrkemmin kuin absoluuttisen monimuotoisuuden suora.

Testifunktioista Schwefel ja Griewank havaitaan, että puolilogaritmiselle asteikolle piirrettyjen absoluuttisen monimuotoisuuden ja tavoitefunktion arvon (tai tavoitefunktion arvon ja funktion optimiarvon erotus) kuvaajat muistuttavat muodoltaan läheisesti toisiaan. Näin tapahtuu siitakin huolimatta, että kyseessä ei ole ideaalitapaus, ja siten kuvaajat eivät ole suoria. Siten niissä kohdissa, joissa absoluuttinen monimuotoisuus pysyy tasaisena tai

laskee hitaasti, tavoitefunktion arvo pysyy vakiona tai pienenee hitaasti. Tämä vahvistaa oletuksen lokaalien optimien vaikutuksesta absoluuttisen monimuotoisuuden kehittymiseen.

6.2 Testitapaus 2

Seuraavalla testitapauksella, jossa parametrien määrää on nostettu 10:een ja populaation kokoa 100:aan, havaitaan tulosten noudattavan samoja periaatteita kuin edellä. Helpoimmilla testifunktioilla, Sphere ja Rosenbrock (liite 3, kuvat 1-2), suhteellinen monimuotoisuus pysyy vakiona noin 0,75:ssä, mutta Rosenbrock:ssa noin 400 sukupolven kohdalla se ensin kasvaa, sitten laskee nopeasti ja häviää kokonaan.

Puolilogaritmiselle asteikolle piirretyt absoluuttinen monimuotoisuus ja tavoitefunktion arvo muodostavat laskevat suorat, mutta ne ovat huomattavasti loivemmat kuin helpoimmassa testitapauksessa. Lisäksi Rosenbrock:ssa tapahtuu kummassakin käyrässä jyrkkä lasku samassa kohdassa, jossa havaittiin suhteellisen monimuotoisuuden pieneneminen. Laskun syynä on konvergoituminen globaaliin optimiin. Schwefel'in ja Griewank'in tapauksessa konvergoituminen on myös huomattavasti hitaampaa kuin helpommalla testitapauksella, mikä on ymmärrettävää kun parametrien määrää ja populaation kokoa on kasvatettu.

Kuten kuvaajista (liite 3, kuva 3) käy ilmi, on Schwefel:iä jouduttu tällä ja tätä seuraavilla testitapauksilla ajamaan 5000 sukupolvea, jotta saataisiin käsitys haun edistymisestä. Kaikkia muita testifunktioita on ajettu 500 sukupolvea. Aivan kuten aiemmalla testitapauksella, suhteellinen monimuotoisuus kasvaa haun loppupuolella huomattavasti, mutta laskee ja jää hieman yli arvon yksi. Syynä tähän on konvergoituminen globaaliin optimiin. Myös Griewank'in funktiolla havaitaan tällä kertaa vastaavanlainen suhteellisen monimuotoisuuden kasvu, joka johtuu haun osumisesta lokaaliin optimiin. Vaarana on haun juuttuminen, mutta tässä tapauksessa haku pääsee pois lokaalista optimista, ja optimointi pääsee jatkumaan normaalisti.

6.3 Testitapaus 3

Seuraavassa testitapauksessa on käytetty 20 parametria ja populaation kokoa 50, jota voidaan pitää melko pienenä parametrien määrään verrattuna. Kaikissa neljässä testifunktiossa suhteellisen monimuotoisuuden arvo pysyttelee noin 0,8:ssa.

Schwefel'in funktiolla suhteellinen monimuotoisuus pysyy kuitenkin aluksi melko pitkään noin arvossa 1, jonka jälkeen se nopeasti kasvaa, laskee ja tasoittuu 0,8:aan. Testifunktioilla Sphere ja Rosenbrock (liite 4, kuvat 1-2) havaitaan, että puolilogaritmiselle asteikolle piirretyt suhteellisen monimuotoisuuden ja tavoitefunktion arvot eivät muodosta enää suoraa, vaan käyrän, joka aluksi laskee ja sitten tasoittuu vakioksi. Käyrän tasoittuminen merkitsee, ettei haku etene, eikä siten kummallakaan testifunktiolla saavuteta globaalia optimia. Kummassakaan tapauksessa haun juuttumista lokaaliin optimiin ei voida havaita millään suhteellisesta monimuotoisuudesta.

Schwefel'in funktiossa (liite 4, kuva 3) absoluuttista monimuotoisuutta tarkasteltaessa puolilogaritmisellä asteikolla havaitaan sen ensin pysyvän melko pitkään vakiona lähtötasolla, kunnes se laskee erittäin jyrkästi. Tämän jälkeen se kohoaa jälleen hiukan, ja jatkaa hieman vaihtelevasti ajon loppuun asti. Jyrkkä lasku on seurausta suppenemisesta lokaaliin optimiin, joka havaitaan myös tavoitefunktion kuvaajasta. Haku näyttää juuttuvan kyseiseen lokaaliin optimiin, sillä tavoitefunktion arvo ei enää tuon jälkeen parane, eikä globaalia optimia saavuteta. Myös Griewank'in funktiolla tapahtuu suppeneminen lokaaliin optimiin, eikä globaalia optimia löydetä. Myöskään tässä tapauksessa ei haun juuttuminen näy suhteellisen monimuotoisuuden arvossa millään lailla.

6.4 Testitapaus 4

Seuraavaan testitapaukseen on populaation koko muutettu 100:aan ja parametrien lukumäärä pidetty 20:ssä. Erona edelliseen havaitaan suhteellisen monimuotoisuuden olevan kaikilla funktioilla noin 0,75.

Funktiossa Sphere absoluuttinen monimuotoisuus pienenee hitaammin kuin edellisessä testitapauksessa, mutta tavoitefunktion arvo pienenee hieman nopeammin. Tavoitefunktion

arvo saavuttaa vain vähän edellistä testitapausta paremman arvon, ja globaali optimi jää siten löytymättä. Testifunktion Rosenbrock käyttäytyminen on aivan samanlaista kuin Sphere:llä.

Myöskään Schwefel:in funktion käyttäytyminen ei poikkea suuresti edellisen testitapauksen käyttäytymisestä, vain suhteellinen ja absoluuttinen monimuotoisuus säilyvät alussa tasaisina pitempään, ja puolilogaritmisella asteikolla absoluuttisen monimuotoisuuden notkahtamisen jälkeen kyseinen arvo pysyy korkeana pitempään. Schwefel:in funktiolla tavoitefunktion arvon ja funktion optimiarvon erotus on edellistä testitapausta hieman matalampi, eli parempi. Suhteellisen monimuotoisuuden arvo kääntyy arvon 1 yläpuolella, ja seurauksena on haun juuttuminen lokaaliin optimiin. Testifunktio Griewank poikkeaa edellisestä testitapauksesta kuten Sphere ja Schwefel. Myös Griewank:in funktiolla tapahtuu juuttuminen lokaaliin optimiin, mutta se ei ole havaittavissa mitenkään suhteellisesta monimuotoisuudesta.

Eri testitapauksia vertailtaessa on otettava huomioon, että kukin testifunktio on ajettu kullakin testitapauksella vain kerran, jolloin pienet eroavaisuudet eri testitapausten välillä saattavat johtua ajokohtaisista eroista.

6.5 Testitapaus 5

Viimeisessä testitapauksessa populaation koko on muutettu 200:aan ja parametrien lukumäärä on pidetty 20:ssä. Suhteellinen monimuotoisuus on edelleen pienentynyt kaikissa testifunktioissa, ja se on nyt noin 0,65. Oleellisimpana erona edelliseen testitapaukseen verrattuna voidaan Sphere:n ja Rosenbrock:in tapauksessa pitää sitä, että puolilogaritmiselle asteikolle piirretyt absoluuttisen monimuotoisuuden ja tavoitefunktion arvot muistuttavat jälleen suoraa, eikä juuttumista näytä tapahtuvan. Kyseisten funktioiden optimointiajoja voidaan pitää onnistuneina. Luultavasti hakua jatkettaessa juuttuminen olisi kuitenkin tapahtunut ennemmin tai myöhemmin. Tavoitefunktion arvot ovat molemmilla funktioilla huomattavasti paremmat kuin edellisellä testitapauksella, ja ne ovat lähellä globaalia optimia.

Schwefel:in funktiolla suhteellinen ja absoluuttinen monimuotoisuus säilyy alussa tasaisena vielä edellistäkin testitapausta pitempään. Lisäksi absoluuttinen monimuotoisuus säilyy ajon loppupuolella edellistä testiä korkeampana, mutta tavoitefunktioiden arvoissa ei ole havaittavissa merkittävää muutosta. Griewank:in funktion tärkeimmät muutokset edelliseen testitapaukseen verrattuna liittyvät myös tavoitefunktion arvoon, joka on jälkimmäisessä testissä huomattavasti parempi kuin edellisessä ja siten lähempänä globaalia optimia. Haku on hetken juuttuneena lokaaliin optimiin, mutta pääsee pois lokaalista optimista, ja haku pääsee jatkumaan normaalisti.

6.6 Tulosten tulkintaa

Suoritettujen testien perusteella havaittiin puolilogaritmiselle asteikolle piirrettyjen absoluuttisen monimuotoisuuden ja tavoitefunktion arvon olevan saman muotoiset niissä tapauksissa, joissa haku etenee hyvin ja globaali optimi löydetään. Testitapauksissa 3 ja 4 todettiin populaation koon olevan liian pieni parametrien määrään verrattuna, mikä johti haun juuttumiseen. Käytettäessä testitapauksessa 5 populaation kokoa 200, ei juuttumista ajettujen sukupolvien aikana enää tapahtunut. Tämä tukee populaation koon valinnassa käytettyä nyrkkisääntöä, jonka mukaan hyvä alkuarvaus populaation koolle saadaan yhtälöllä:

$$NP = 10 \cdot D \quad (42)$$

Testitapauksilla 3, 4 ja 5, joissa kaikissa parametrien lukumääränä oli 20 ja populaation koko vaihteli, havaittiin populaation koon vaikuttavan tasoon, johon suhteellinen monimuotoisuus asettuu. Testitapauksilla 2 ja 4, joissa populaation koko oli molemmissa 100 ja parametrien lukumäärä vaihteli, havaittiin ettei parametrien lukumäärä vaikuta tasoon, jolle suhteellinen monimuotoisuus asettuu. Tämä osoittaa, että suhteellisen monimuotoisuuden laskennassa tehty parametrien lukumäärän huomioon ottava skaalaus toimii ainakin kyseisillä testitapauksilla.

Evoluutioalgoritmien kontrolliparametrien ajonaikaista säätöä varten tulisi tuntea mittarien toiminta onnistuneen optimointiajon tapauksessa. Tyypillisessä onnistuneessa optimointiajossa suhteellinen monimuotoisuus muodostaa vaakatasossa olevan suoran

puolilogaritmisessa koordinaatistossa, ja absoluuttinen monimuotoisuus ja tavoitefunktion arvo ovat kumpikin laskevia suoria puolilogaritmisessa koordinaatistossa tarkasteltaessa.

Eräs mahdollinen säätöperiaate voisi olla, että havaittaessa poikkeama edellä esitetyistä piirteistä kasvatettaisiin populaation kokoa NP tai mutaatiovakiota F tai molempia. Vastaavasti haun sujuessa ideaalisten piirteiden mukaisesti, voitaisiin populaation kokoa tai mutaatiovakiota tai molempia pienentää. Seurauksena olisi haun nopeutuminen, mutta vaarana kyseisten arvojen liiallinen pieneneminen, jonka seurauksena jouduttaisiin tekemään korjausliike vastakkaiseen suuntaan. Tällöin säädössä tapahtuvan viiveen vuoksi voisi lopputuloksena olla alkuperäistä hitaampi ajo.

Toinen mahdollinen säätöperiaate liittyy suhteellisen monimuotoisuuden äkilliseen kasvamiseen yli arvon 1. Tämän havaittiin suoritetuissa testeissä ennakoivan äkillistä suppenemista optimiin. Suppenemisen osuessa lokaaliin optimiin on usein seurauksena haun juuttuminen. Havaittaessa suhteellisen monimuotoisuuden alkavan kasvaa nopeasti, voitaisiin säätää populaation kokoa NP , mutaatiovakiota F , risteytysvakiota CR tai näiden yhdistelmiä.

Toisaalta, kuten testitapauksia analysoitaessa havaittiin, ei haun juuttumista voida aina havaita suhteellisesta monimuotoisuudesta. Siten suhteelliseen monimuotoisuuteen perustuva säätömenetelmä ei toimisi kaikissa tapauksissa. Molempien säätömekanismien toteuttaminen ja käyttökelpoisuuden arviointi vaatisivat kuitenkin vielä tarkempaa tutkimista ja lisää kokeiden suorittamista.

7 JOHTOPÄÄTÖKSET

Kehitetyt suhteellisen ja absoluuttisen monimuotoisuuden mittarit todettiin suoritetun kokeellisen tarkastelun perusteella toimiviksi. Kokeiden perusteella havaittiin, että tavoitefunktion venyttäminen koordinaattiakseleiden suunnassa ei vaikuta mittareiden antamiin arvoihin. Myöskään tavoitefunktion venyttäminen muissa suunnissa ei vaikuttanut mittareiden tuloksiin. Tutkittaessa tavoitefunktion kiertämistä koordinaatistoon nähdén havaittiin, ettei sillä ole merkitystä mittarin toiminnan kannalta.

Tavoitefunktion dimensionaalisuuden muuttaminen otetaan huomioon suhteellista monimuotoisuutta laskettaessa, eikä sillä siten ole vaikutusta mittarin arvoihin. Absoluuttisen monimuotoisuuden tapauksessa dimensionaalisuuden muuttumista ei tarvitse ottaa mittarin arvoa laskettaessa huomioon. Dimensioiden lukumäärää D kasvatettaessa havaittiin absoluuttisen monimuotoisuuden laskevan hitaammin kuin pienempiä dimensioita käytettäessä.

Tarkasteltaessa populaation koon vaikutusta kehitettyjen mittareiden toimintaan, havaittiin sen vaikuttavan tasoon, jolle suhteellinen monimuotoisuus asettuu. Populaation kokoa NP kasvatettaessa havaittiin tason, jolle suhteellinen monimuotoisuus asettuu, laskevan. Kuten dimensioiden lukumäärää kasvatettaessa, myös populaation kokoa kasvatettaessa havaittiin absoluuttisen monimuotoisuuden laskevan hitaammin kuin pienempiä populaatioita käytettäessä.

Suhteellista monimuotoisuutta seuraamalla voidaan joissakin tapauksissa ennakoida tulossa olevia mahdollisia ongelmia. Monihuippuisilla funktioilla suhteellisen monimuotoisuuden nouseminen yli arvon 1 ennakoi nopeaa suppenemista. Nopea suppeneminen osuu todennäköisesti monihuippuisella funktiolla lokaaliin optimiin, josta on usein seurauksena juuttuminen. Suhteellista monimuotoisuutta seuraamalla voisi olla mahdollista havaita suppeneminen heti sen alkaessa, ja haun parametreja säätämällä se voitaisiin välttää. Tämä tosin vaatisi vielä lisää suhteellisen monimuotoisuuden käyttäytymisen tutkimista, ja lisää kokeiden suorittamista. Suhteellisen monimuotoisuuden

kohoaminen yli arvon 1 oli havaittavissa monissa monihuippuisten funktioiden kokeissa, joissa tapahtui juuttuminen lokaaliin optimiin.

Suhteellisen monimuotoisuuden mittarin soveltaminen evoluutioalgoritmin kontrolliparametrien, kuten esimerkiksi mutaatio- ja risteytysvakioiden, säätöön ei nykyisellään ole mielekästä, sillä joissakin tapauksissa se toimii tavalla, josta ei voida nähdä mahdollisia haussa tapahtuvia ongelmatilanteita millään lailla. Jotta kontrolliparametrien säätö olisi mahdollista, pitäisi suhteellisen monimuotoisuuden mittaria kehittää edelleen, ja tutkia lisää sen käyttäytymistä. Sen sijaan kehitettyä absoluuttisen monimuotoisuuden mittaria voidaan sellaisenaan käyttää evoluutioalgoritmin lopetusehdon toteuttamiseen. Tällöin absoluuttiselle monimuotoisuudelle määrätään alaraja, ja absoluuttisen monimuotoisuuden saavuttaessa tuon rajan voidaan haku lopettaa. Kontrolliparametrien säätöä ajatellen voisi olla hyödyllistä toteuttaa työssä kehitettyjä mittareita vastaavat mittarit myös tavoitefunktioavaruuteen.

Mittareiden käytännön sovellusten lisäksi niistä on hyötyä evoluutioalgoritmien toiminnan analysoinnissa. Mittarit antavat arvokasta tietoa suoritettavan haun etenemisestä ja mahdollisista ongelmista optimoinnin aikana. Mittareiden antamat tulokset ovat luettavissa näytöltä tai tiedostosta ajon aikana, ja ajon päätyttyä kaikkien kierrosten monimuotoisuuden arvot löytyvät tiedostosta. Tiedostoon tulostetuista arvoista saadaan helposti piirrettyä havainnolliset kuvaajat käyttämällä taulukkolaskentaa tai muuta kuvaajien piirtoon soveltuvaa ohjelmaa.

Mittareiden antamien tulosten tulkinnassa auttaa suuresti, jos tunnetaan funktion ominaisuudet. Tällöin mittareiden käyttäytymistä on helppo tulkita. Toisaalta ideaalisten ajojen tapauksessa tulkinta on helppoa, vaikkei olisikaan tietoa käytetystä funktiosta. Etenkin tieto käytetyn funktion yksi- tai monihuippuisuudesta auttaa haun aikana mahdollisesti tapahtuneiden ongelmatilanteiden ymmärtämisessä. Mittarin mielekkään käyttämisen kannalta on tunnettava mittarin käyttäytyminen, ja osattava tulkita sen antamia tuloksia.

Suhteellisen ja absoluuttisen monimuotoisuuden laskemiseen tarvittava aika kasvaa lineaarisesti populaation koon kasvaessa. Se on siten nopeampi kuin Šmuc'in menetelmä ja

samaa luokkaa Lopez Cruz *et al.*:n menetelmän kanssa. Etuna Lopez Cruz *et al.*:n menetelmään on se, että työssä esitetty menetelmä ottaa huomioon monimuotoisuuden määrän, eikä ole vain tietyn määritellyn monimuotoisuusrajan ylittävien yksilöiden lukumäärä. Lineaarisen aikakompleksisuuden vuoksi mittareiden käyttäminen on mahdollista myös suuria populaatioita vaativissa ongelmissa. Mittareiden toiminta ei vaadi ylimääräisiä parametreja, joten ne ovat melko helposti käytettävissä erilaisten evoluutioalgoritmien kanssa. Suhteellisen monimuotoisuuden antamat tulokset ovat vertailukelpoisia riippumatta parametrien lukumäärästä tai populaation koosta.

LÄHTEET

- [Bar00] Barker, Allen L. and Martin W.N. (2000). Dynamics of a Distance-Based Population Diversity Measure [verkkojulkaisu]. [viitattu 22.10.2002]. Saatavissa: <http://www.ieee.org>
- [Deb01] Deb, Kalyanmoy. (2001). Multi-Objective Optimization using Evolutionary Algorithms. 1st edition. John Wiley & Sons, Ltd. ISBN 0-471-87339-X.
- [Deb02] Deb, K., Agrawal, S., Pratap, A. and Meyarivan, T. (2002). A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II. Technical Report 200001, Indian Institute of Technology, Kanpur: Kanpur Genetic Algorithms Laboratory (KanGAL). Saatavissa: IEEE Trans. on Evolutionary Computation, 6(2): 182-197, April 2002 tai <http://www.work.caltech.edu/amrit/publications.html>.
- [DeJ75] De Jong, K. (1975). An Analysis of the Behaviour of a Class of Genetic Adaptive Systems. Ph.D. thesis, University of Michigan.
- [Lam00] Lampinen, Jouni. Multiobjective Nonlinear Pareto-Optimization, A Pre-Investigation Report [verkkojulkaisu]. Lappeenranta University of Technology, Finland, 2000 [viitattu 13.9.2002]. Saatavissa: <http://www.it.lut.fi/opetus/01-02/010778000/Pareto.pdf>
- [Lam00a] Lampinen, Jouni and Zelinka, Ivan. (2000). On Stagnation of the Differential Evolution Algorithm. In proceedings of the 6th International Conference on Soft Computing MENDEL 2000, Brno, Czech Republic, June 7-9, 2000.
- [Lam01] Lampinen, Jouni. (2001). Global Optimization by Differential Evolution [verkkojulkaisu]. Lappeenranta University of Technology, Finland, 2001 [viitattu 04.11.2002]. Saatavissa: <http://www.it.lut.fi/opetus/0102/010778000/DE.pdf>

- [Lam01a] Lampinen, Jouni. (2001). DE's Selection Rule for Multiobjective Optimization. [verkkojulkaisu]. Lappeenranta University of Technology, Finland, 2001. [viitattu 27.11.2002]. Saatavissa: <http://www.it.lut.fi/kurssit/02-03/010778000/MODE.pdf>
- [Lam01b] Lampinen, Jouni. (2001). Multi-Constrained Nonlinear Optimization by Differential Evolution Algorithm. [verkkojulkaisu]. Lappeenranta University of Technology, Finland, 2001. [viitattu 18.12.2002]. Saatavissa: <http://www.it.lut.fi/kurssit/02-03/010778000/DECONSTR.PDF>
- [Lop01] Lopez Cruz, I.L., Van Willigenburg, L.G. and Van Straten, G. (2001). Parameter Control Strategy in Differential Evolution Algorithm for Optimal Control. In proceedings of the IASTED International Conference Artificial Intelligence and Soft Computing, 2001, Cancun, Mexico.
- [Luu02] Luukka, Pasi and Saastamoinen, Kalle. (2002). Fuzzy Similarity Based Mutation in Genetic Algorithm. Lappeenranta University of Technology, Finland, 2002. In the proceedings of the Joint 1st International Conference on Soft Computing and Intelligent Systems, SCIS & ISIS 2002.
- [Loc03] Locatelli, M. (2003). A Note on the Griewank Test Function. Journal of Global Optimization 25: 169-174, 2003. Kluwer Academic Publishers, 2003, Netherlands.
- [Poh94] Pohlheim, Hartmut. (1994). Genetic Algorithm Toolbox Test Functions. [verkkojulkaisu]. Department of Automatic Control and Systems Engineering, University of Sheffield. [viitattu 30.01.2003.] Saatavissa: http://clio.mit.csu.edu.au/subjects/itc554/test_fns.pdf
- [Sch95] Schott, J.R. (1995). Fault Tolerant Design Using Single and Multi-Criteria Genetic Algorithms. M.Sc. thesis, Boston, MA: Department of Aeronautics and Astronautics, Massachusetts Institute of Technology.

- [Šmu02] Šmuc, Tomislav. (2002). Improving Convergence Properties of the Differential Evolution. In the proceedings of the 8th International Conference on Soft Computing MENDEL 2002, Brno, Czech Republic, 2002.
- [Sto95] Storn, Rainer and Price, Kenneth. (1995). Differential Evolution – A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces. Technical report TR-95-012. ICSI. [viitattu 26.11.2002].
Saatavissa: <ftp://ftp.icsi.berkeley.edu/pub/techreports/1995/tr-95-012.pdf>
- [Sto97] Storn, Rainer and Price, Kenneth. (1997). Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization 11: 341-359, 1997. Kluwer Academic Publishers, 1997.
- [Whi96] Whitley, D. *et al.* (1996) Evaluating Evolutionary Algorithms. Artificial Intelligence 85 (1996) 245-176. Elsevier, 1996.
- [Yur94] Yuret, Deniz. (1994). From Genetic Algorithms to Efficient Optimization. M.Sc. thesis. Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1994.
- [Zah02] Zaharie, Daniela. (2002). Critical Values for the Control Parameters of Differential Evolution Algorithms. In proceedings of the 8th International Conference on Soft Computing MENDEL 2002, Brno, Czech Republic, June 5-7, 2002.
- [Zit99] Zitzler, E. (1999). Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. Ph. D. thesis, Zürich, Switzerland: Swiss Federal Institute of Technology (ETH) (Dissertation ETH No. 13398). Shaker Verlag, Germany, ISBN 3-8265-6831-1. [viitattu 07.11.2002]. Saatavissa: <http://www.tik.ee.ethz.ch/~zitzler/>

Liite 1. Populaation suhteellista ja absoluuttista monimuotoisuutta mittaavan mittarin C-lähdekielinen ohjelma.

```
#include "de.h"

// Globals
float (*pold)[MAXPOP][MAXDIM]; /* Pointer to the population */

/* Function diversity(input_conf *input, diversity_info *div_ptr)
/*
/*   Programmed by Esa Ruuth
/*   2003
/*
/*   Function call: diversity(input,&divers);
/*
/*   The first parameter is a struct which have to contain (at least):
/*
/*       int D    - Dimension of population (Number of rows of
/*                population)
/*       int NP   - Number of population members (Number of columns of
/*                population)
/*
/*   And the second parameter is a struct which contains:
/*
/*       float X_min[MAXDIM]      - lowest value for each parameter D
/*                                (row) in the initial population
/*       float X_max[MAXDIM];     - highest value for for each
/*                                parameter D (row) in the initial
/*                                population
/*       float relative_diversity; - variable for relative diversity,
/*                                initialized as 0
/*       float absolute_diversity; - variable for absolute diversity,
/*                                initialized as 0
*/
*/

void diversity(input_conf *input, diversity_info *div_ptr)
{
    int i=0,j=0; /* counter variables */
    float population[MAXPOP][MAXDIM]; /* temp table for population */
    float mean[MAXDIM]; /* mean values of each row of
                        the population */
    float stddev[MAXDIM]; /* standard deviation of each
                        row of the population */
    float rdiv[MAXDIM],adiv[MAXDIM]; /* temp variables used in
                        counting of relative and
                        absolute diversity */
    float Xmin[MAXDIM], Xmax[MAXDIM]; /* lowest and highest values
                        of each row in the current
                        population */

    /* initialize diversity variables */
    div_ptr->relative_diversity = 0;
    div_ptr->absolute_diversity = 0;
}
```

(jatkuu)

(liite 1 jatkoa)

```
/* initialize arrays */
for (j=0;j<input->D;j++)
{
    mean[j]=0;
    stddev[j]=0;
    rdiv[j]=0;
    adiv[j]=0;
    Xmin[j]=0;
    Xmax[j]=0;
    for (i=0;i<input->NP;i++)
        population[i][j]=0;
}

/* initialize the array for the use of relative diversity */
for (j=0;j<MAXDIM;j++)
{
    Xmin[j]=HUGE; /* biggest possible number */
    Xmax[j]=-HUGE; /* smallest possible number */
}

/* population is copied into new table */
for (i=0;i<input->NP;i++)
{
    for(j=0;j<input->D;j++)
    {
        population[i][j]=(*pold)[i][j];
    }
}

/* mean values of each parameter row */
for (j=0;j<input->D;j++)
{
    for (i=0;i<input->NP;i++)
    {
        mean[j]=mean[j]+population[i][j];
    }
    mean[j]=mean[j]/input->NP;
}

/* calculation of standard deviation */
for (j=0;j<input->D;j++)
{
    for (i=0;i<input->NP;i++)
    {
        /* find lowest and highest value for each parameter
        row of the current population */
        if (population[i][j]<Xmin[j])
            Xmin[j]=population[i][j];
        if (population[i][j]>Xmax[j])
            Xmax[j]=population[i][j];
        /* standard deviation */
        stddev[j]= stddev[j] + pow((population[i][j] -
            mean[j]),2);
    }
    stddev[j]=stddev[j]/(input->NP);
    stddev[j]=sqrt(stddev[j]);
}
}
```

(jatkuu)

(liite 1 jatkoa)

```
/* calculate relative and absolute diversity */
for (j=0;j<input->D;j++)
{
    /* prevents zero dividing in case of relative diversity */
    if (Xmax[j] - Xmin[j] != 0)

        /* standard deviation is divided by differential of
           highest and lowest value of current parameter row
           of the current population */
        rdiv[j] = stddev[j] / (Xmax[j] - Xmin[j]);
    else
        rdiv[j] = 0;

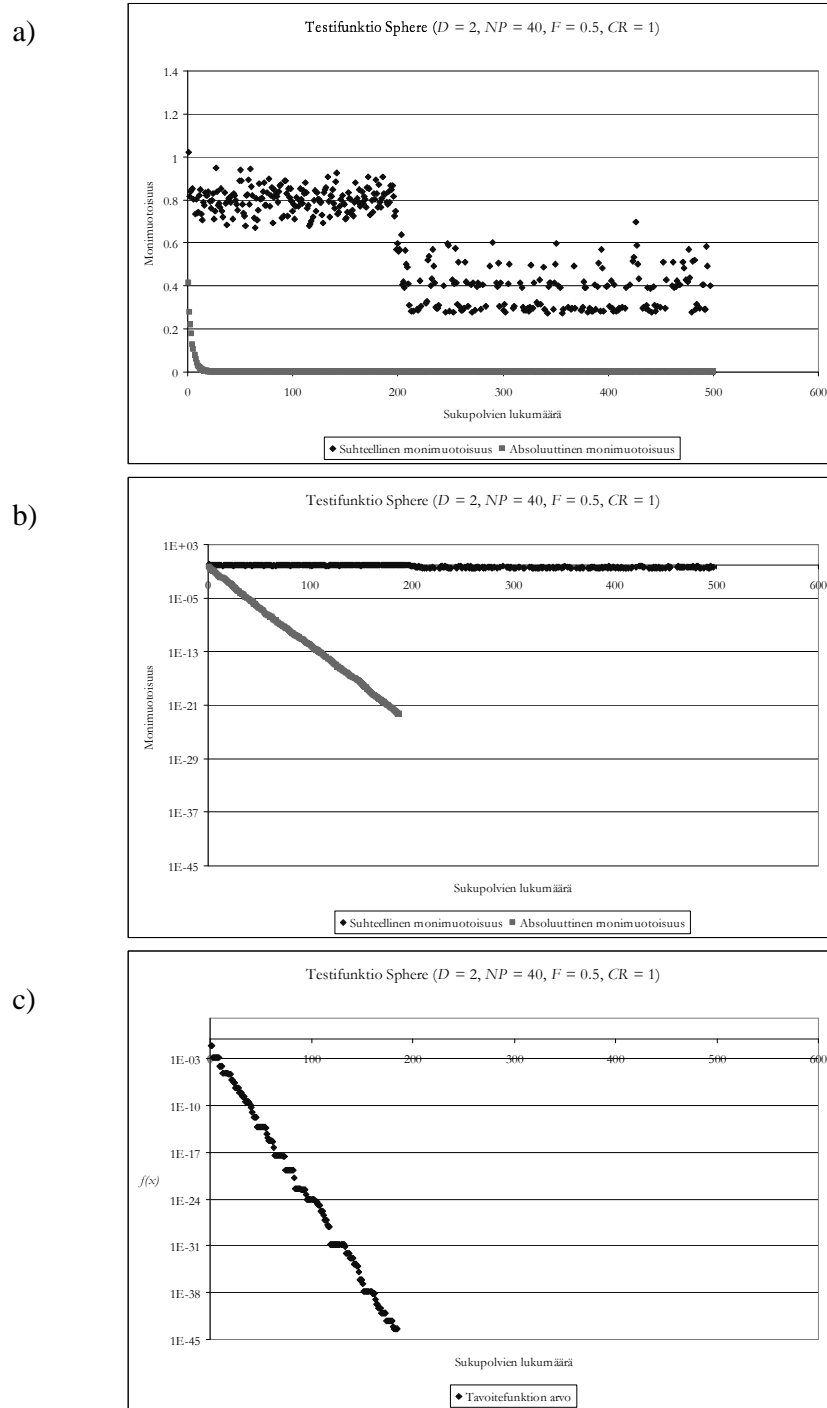
    /* standard deviation is divided by differential of highest
       and lowest value of current parameter row of the initial
       population */
    adiv[j] = stddev[j] / (div_ptr->X_max[j] -
        div_ptr->X_min[j]);

    /* elements of diversity vectors are squared and summed */
    div_ptr->relative_diversity = div_ptr->relative_diversity +
        pow(rdiv[j],2);
    div_ptr->absolute_diversity = div_ptr->absolute_diversity +
        pow(adiv[j],2);
}

/* square root is taken from squared and summed values */
div_ptr->relative_diversity = sqrt(div_ptr->relative_diversity);
div_ptr->absolute_diversity = sqrt(div_ptr->absolute_diversity);

/* relative diversity is scaled so that it is multiplied with the
   coefficient sqrt(12)/sqrt(D) */
div_ptr->relative_diversity = (sqrt(12)/sqrt(input->D))*
    div_ptr->relative_diversity;
}
```

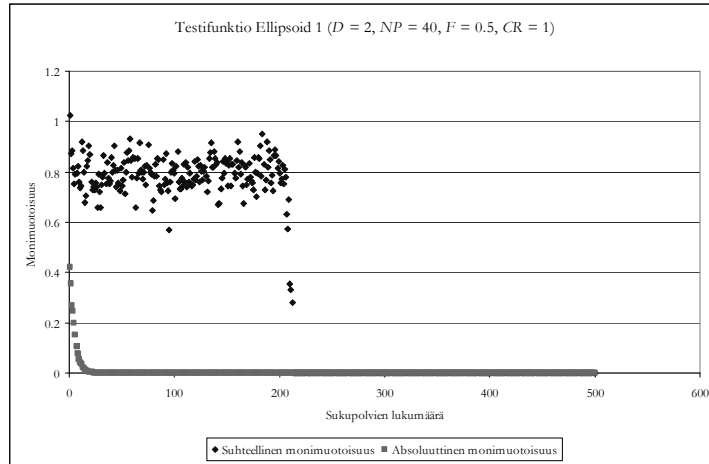
Liite 2. Testitapaus1: testifunktiot Sphere, Ellipsoid 1, Ellipsoid 2, Rosenbrock, Schwefel ja Griewank parametrien arvoilla $D = 2$, $NP = 40$, $CR = 1.0$ ja $F = 0.5$.



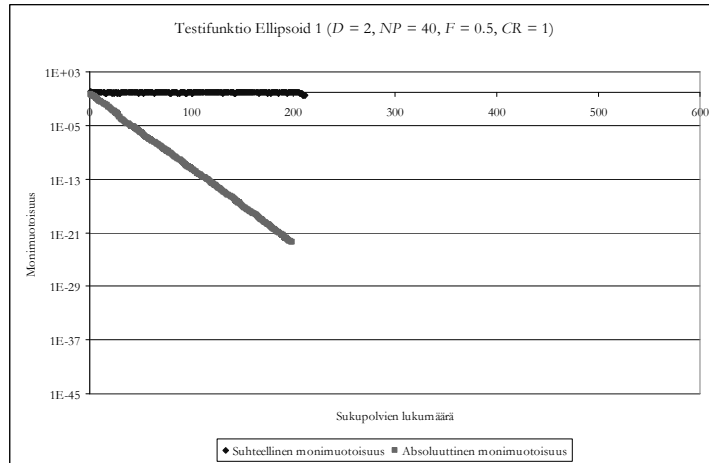
Kuva 12. a) suhteellinen ja absoluuttinen monimuotoisuus b) suhteellinen ja absoluuttinen monimuotoisuus puolilogaritmisella asteikolla c) tavoitefunktion arvo puolilogaritmisella asteikolla

(jatkuu)

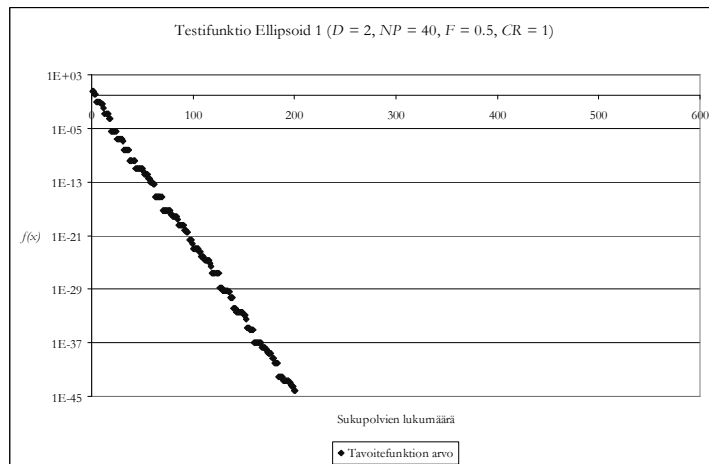
a)



b)

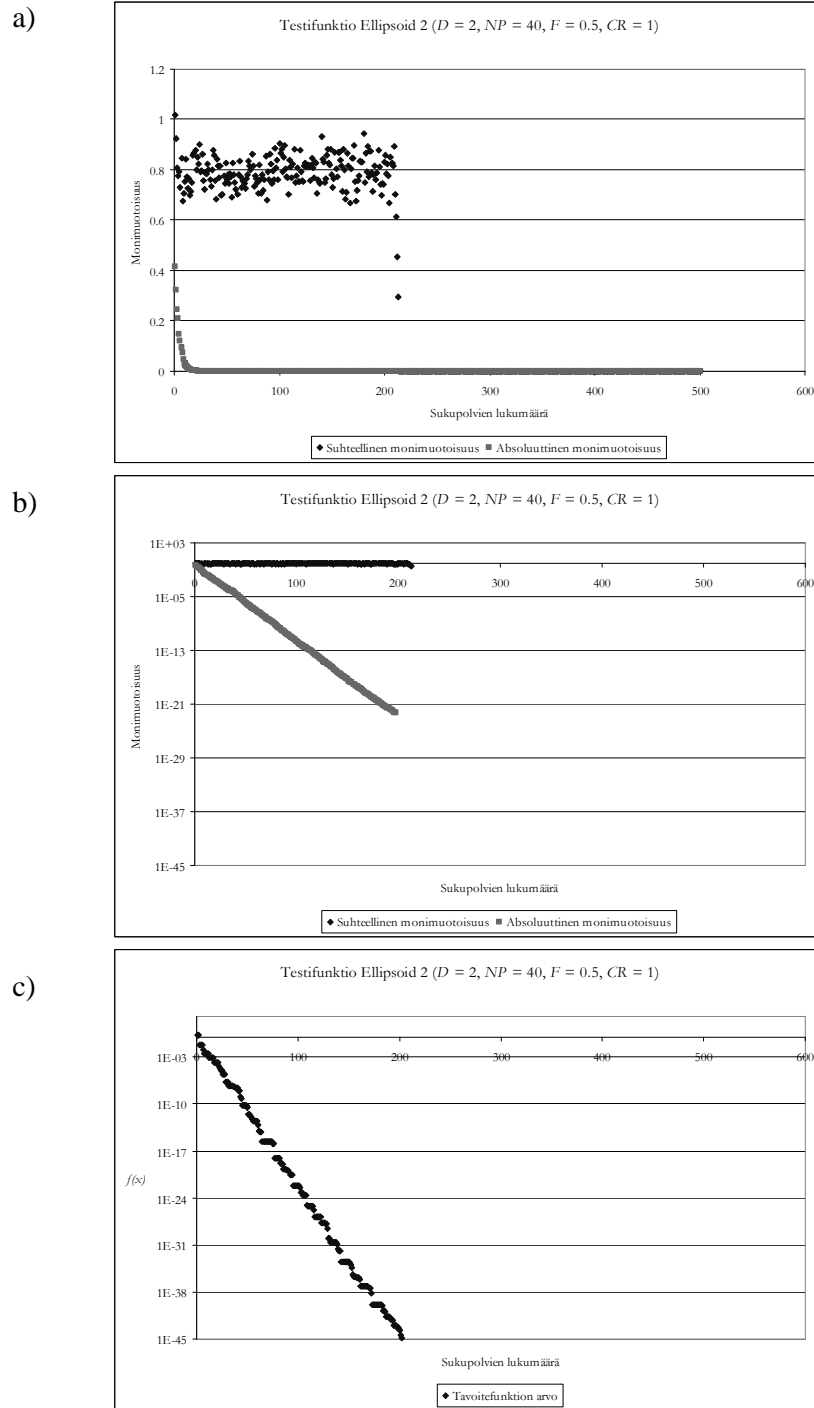


c)



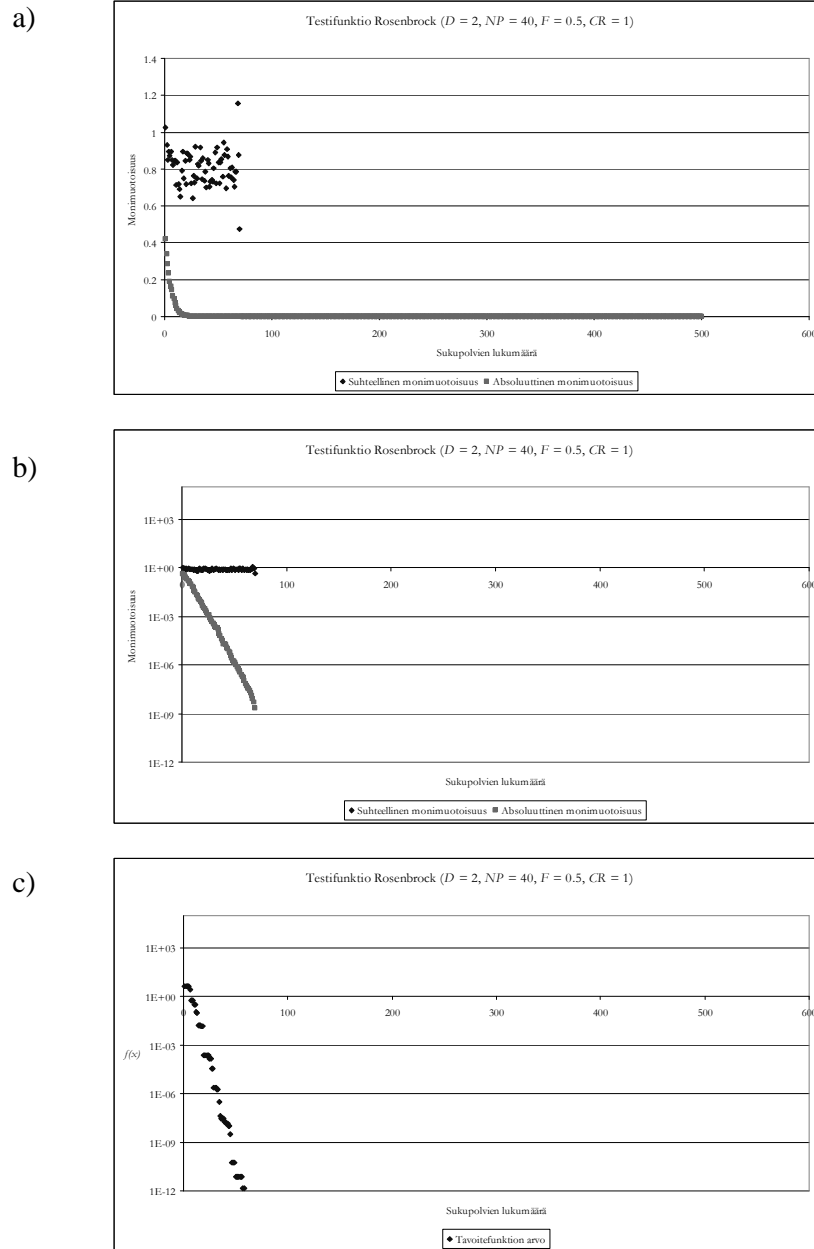
Kuva 13. a) suhteellinen ja absoluuttinen monimuotoisuus b) suhteellinen ja absoluuttinen monimuotoisuus puolilogaritmisella asteikolla c) tavoitefunktion arvo puolilogaritmisella asteikolla

(jatkuu)



Kuva 14. a) suhteellinen ja absoluuttinen monimuotoisuus b) suhteellinen ja absoluuttinen monimuotoisuus puolilogaritmisella asteikolla c) tavoitefunktion arvo puolilogaritmisella asteikolla

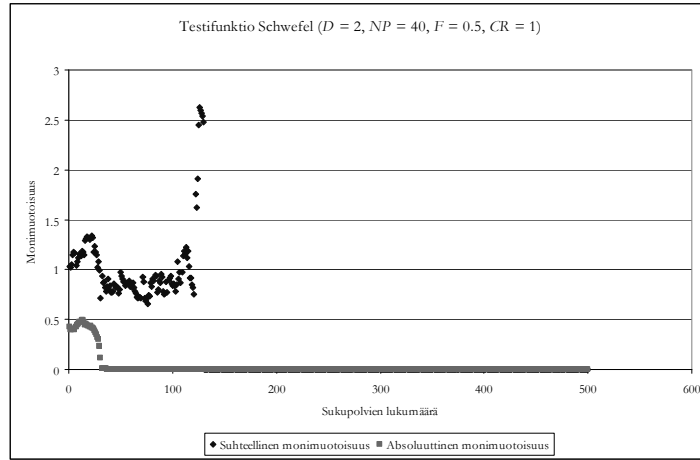
(jatkuu)



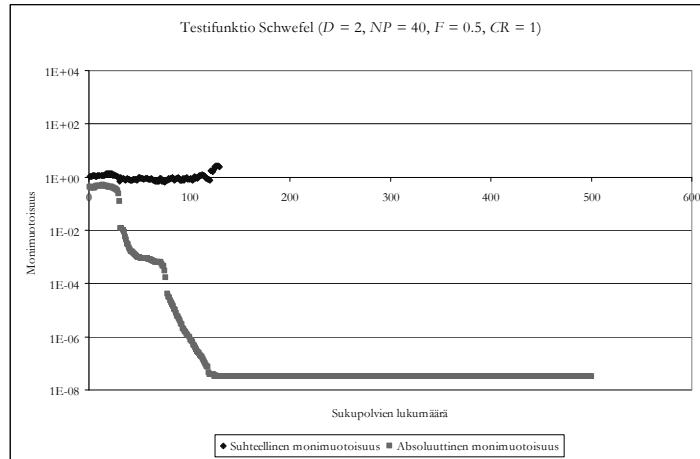
Kuva 15. a) suhteellinen ja absoluuttinen monimuotoisuus b) suhteellinen ja absoluuttinen monimuotoisuus puolilogaritmisella asteikolla c) tavoitefunktion arvo puolilogaritmisella asteikolla

(jatkuu)

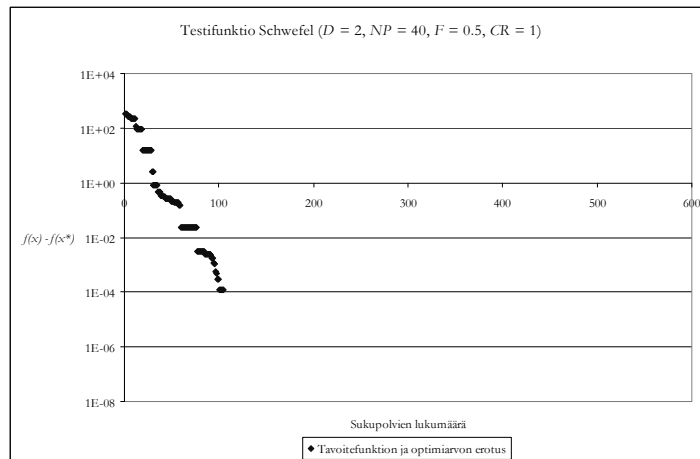
a)



b)



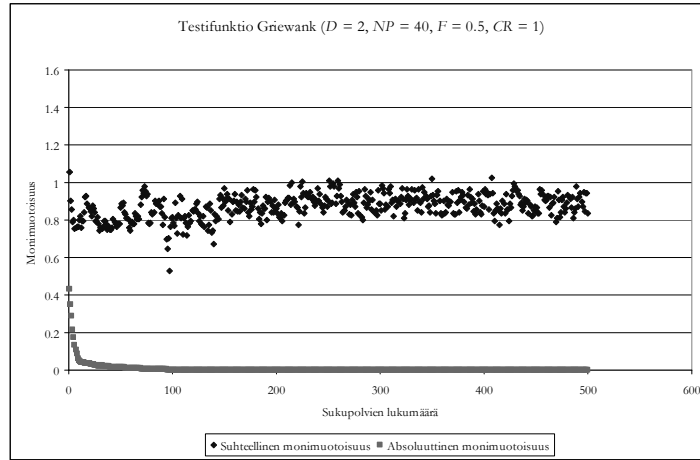
c)



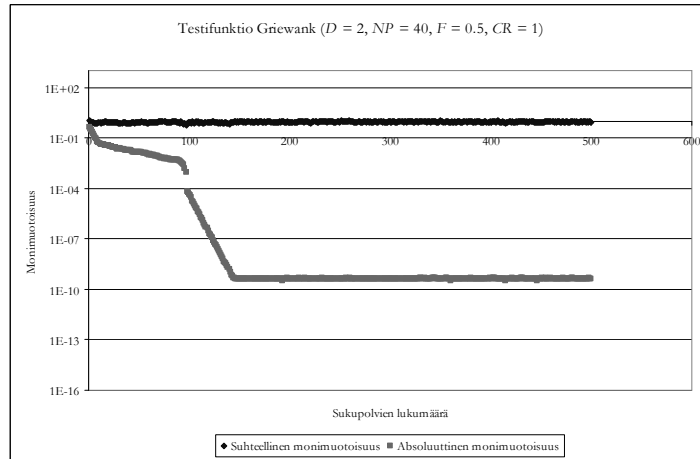
Kuva 16. a) suhteellinen ja absoluuttinen monimuotoisuus b) suhteellinen ja absoluuttinen monimuotoisuus puolilogaritmisella asteikolla c) tavoitefunktion arvon ja funktion optimiarvon erotus puolilogaritmisella asteikolla

(jatkuu)

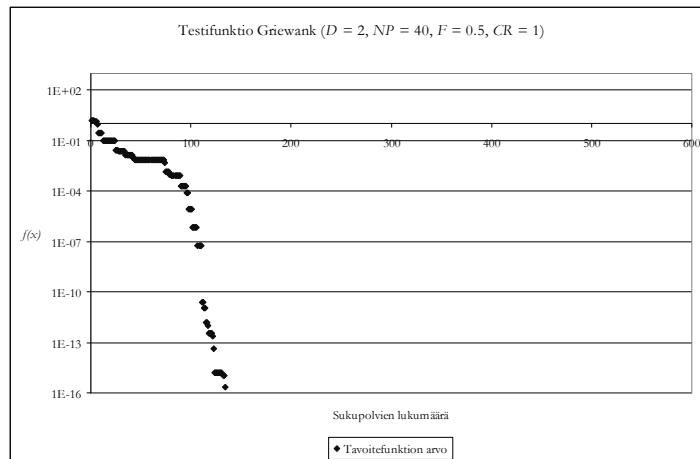
a)



b)



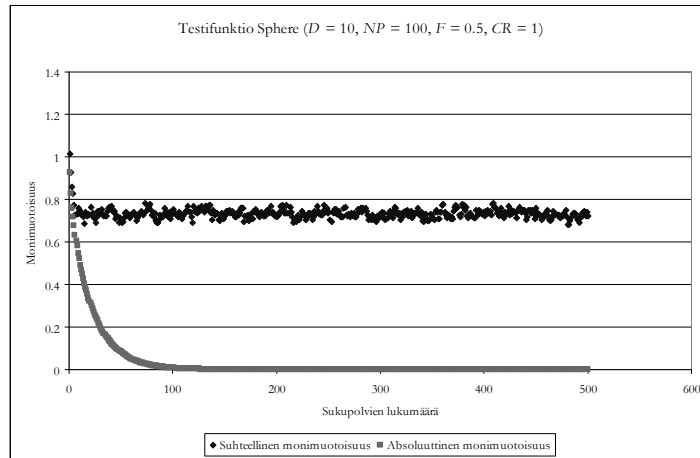
c)



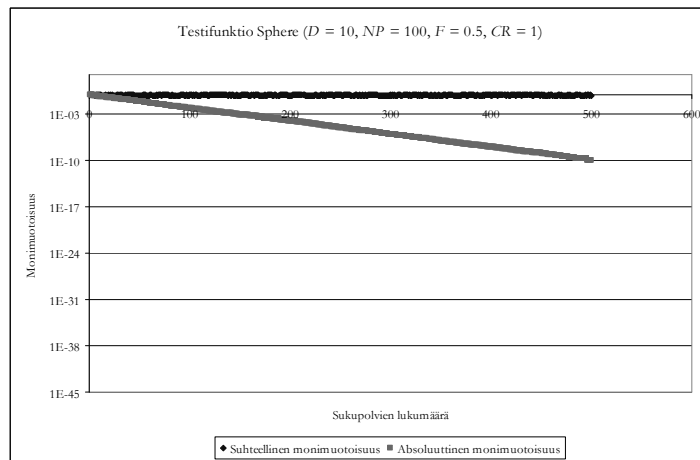
Kuva 17. a) suhteellinen ja absoluuttinen monimuotoisuus b) suhteellinen ja absoluuttinen monimuotoisuus puolilogaritmisella asteikolla c) tavoitefunktion arvo puolilogaritmisella asteikolla

Liite 3. Testitapaus 2: testifunktiot Sphere, Rosenbrock, Schwefel ja Griewank parametrien arvoilla $D = 10$, $NP = 100$, $CR = 1.0$ ja $F = 0.5$.

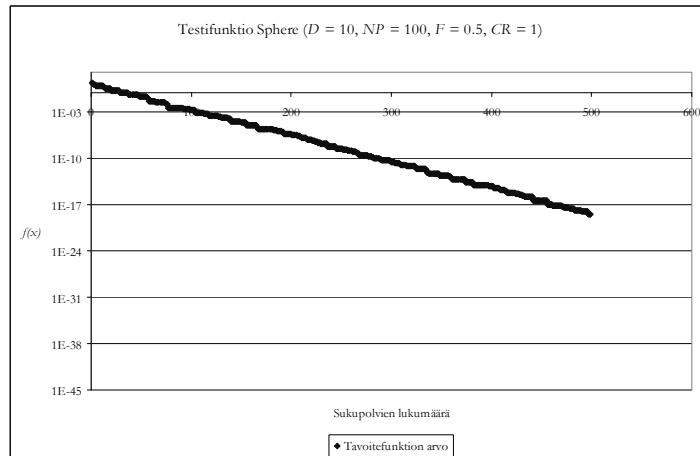
a)



b)

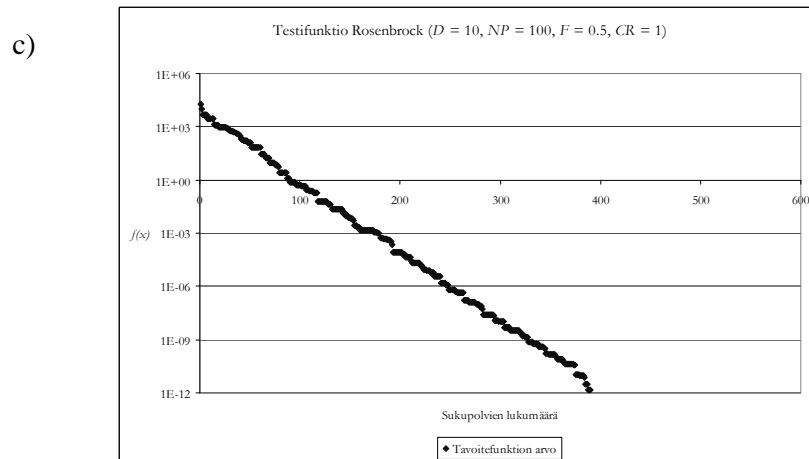
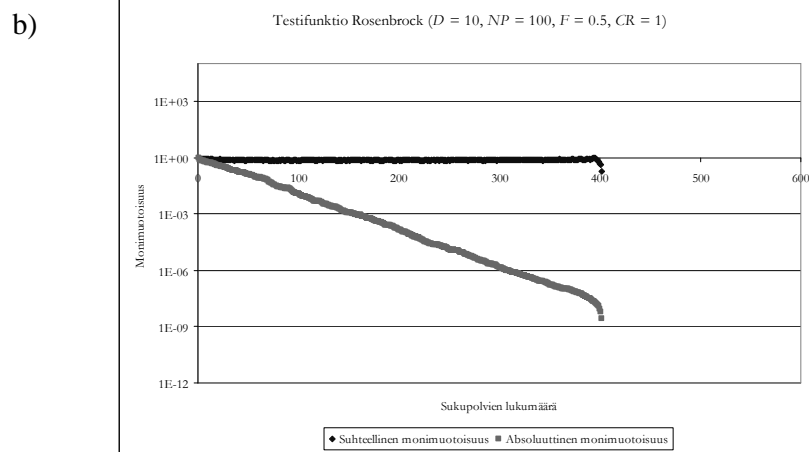
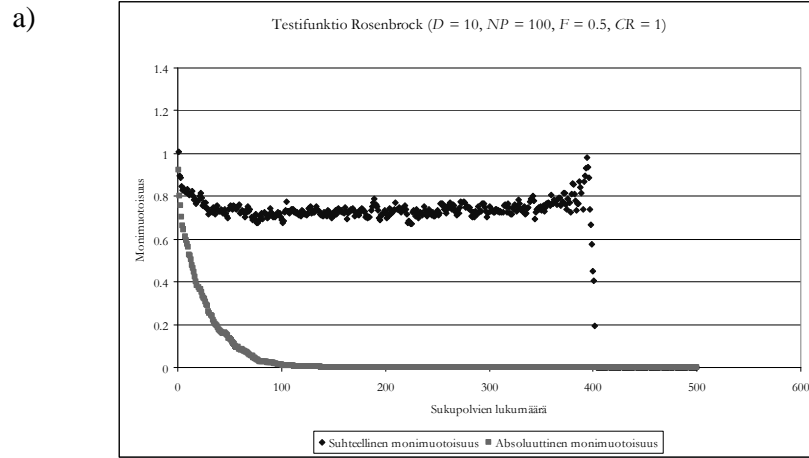


c)



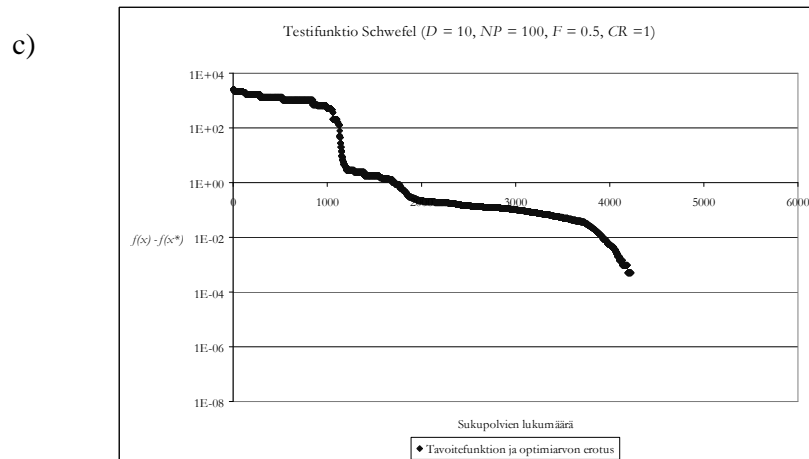
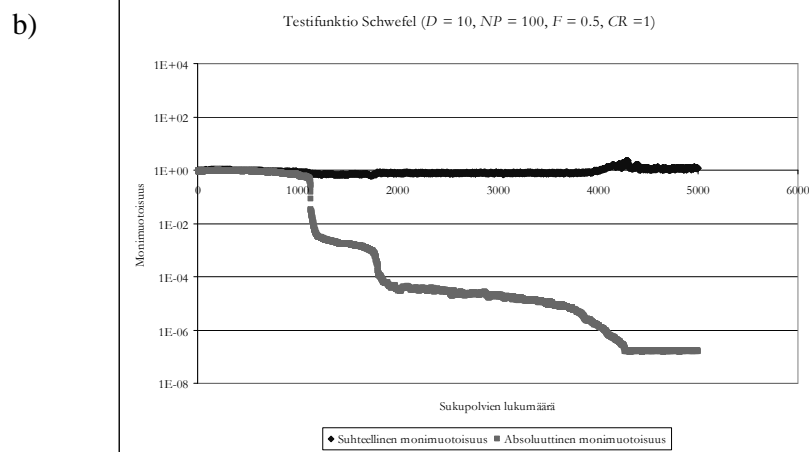
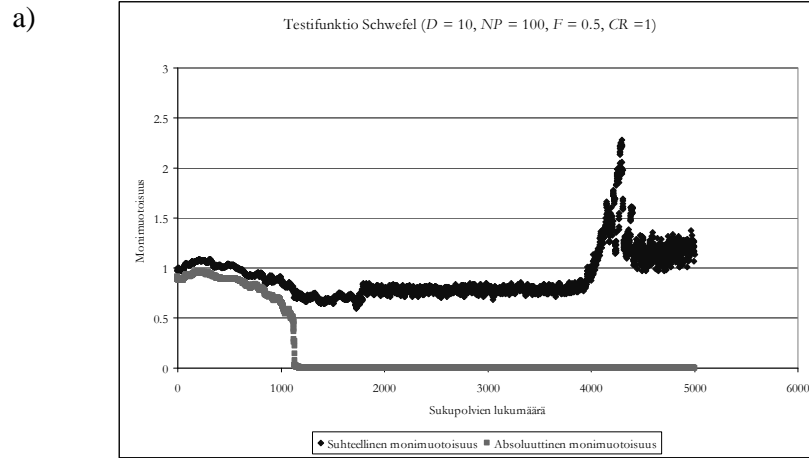
Kuva 18. a) suhteellinen ja absoluuttinen monimuotoisuus b) suhteellinen ja absoluuttinen monimuotoisuus puolilogaritmisella asteikolla c) tavoitefunktion arvo puolilogaritmisella asteikolla

(jatkuu)



Kuva 19. a) suhteellinen ja absoluuttinen monimuotoisuus b) suhteellinen ja absoluuttinen monimuotoisuus puolilogaritmisella asteikolla c) tavoitefunktion arvo puolilogaritmisella asteikolla

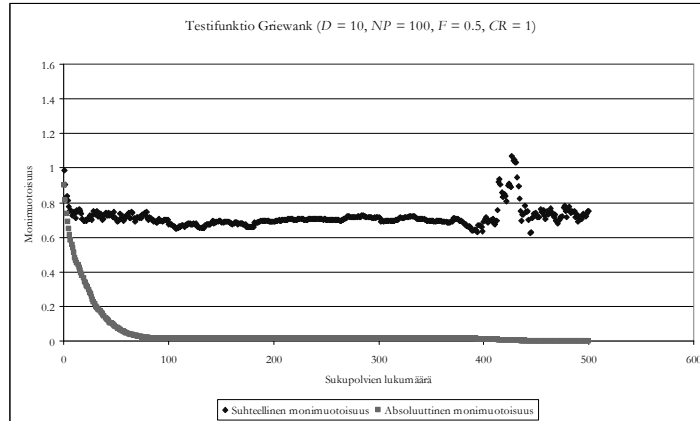
(jatkuu)



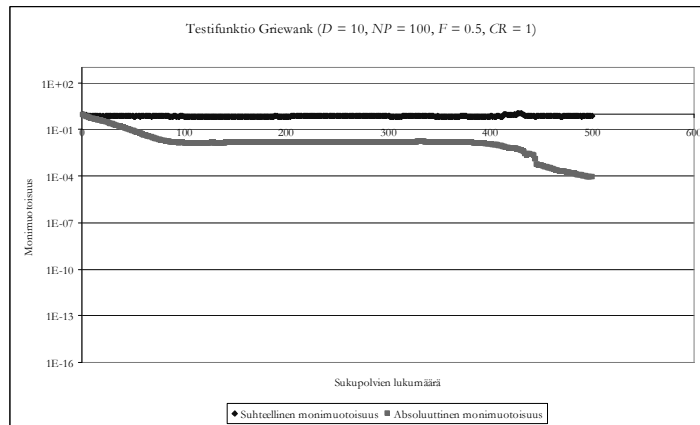
Kuva 20. a) suhteellinen ja absoluuttinen monimuotoisuus b) suhteellinen ja absoluuttinen monimuotoisuus puolilogaritmisella asteikolla c) tavoitefunktion arvon ja funktion optimiarvon erotus puolilogaritmisella asteikolla

(jatkuu)

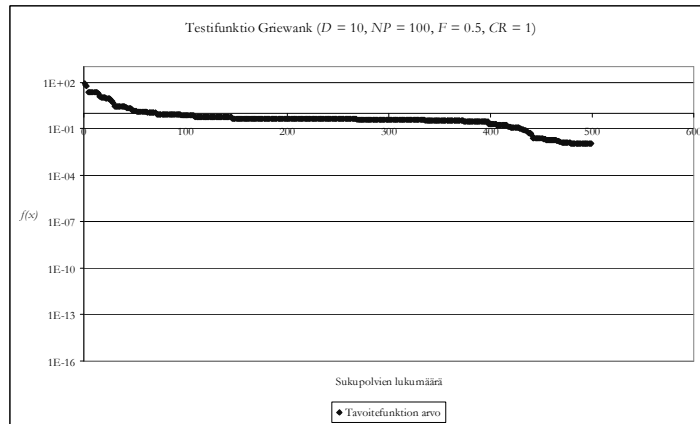
a)



b)



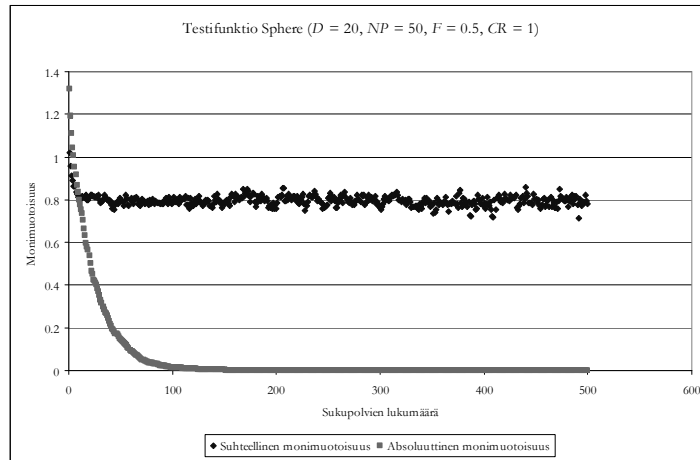
c)



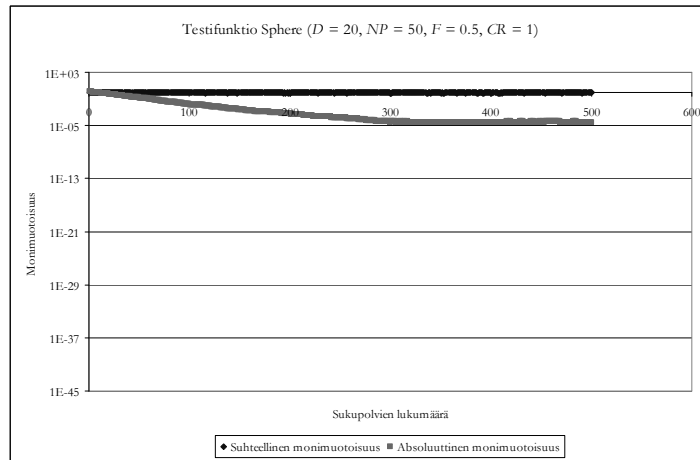
Kuva 21. a) suhteellinen ja absoluuttinen monimuotoisuus b) suhteellinen ja absoluuttinen monimuotoisuus puolilogaritmisella asteikolla c) tavoitefunktion arvo puolilogaritmisella asteikolla

Liite 4. Testitapaus 3: testifunktiot Sphere, Rosenbrock, Schwefel ja Griewank parametrien arvoilla $D = 20$, $NP = 50$, $CR = 1.0$ ja $F = 0.5$.

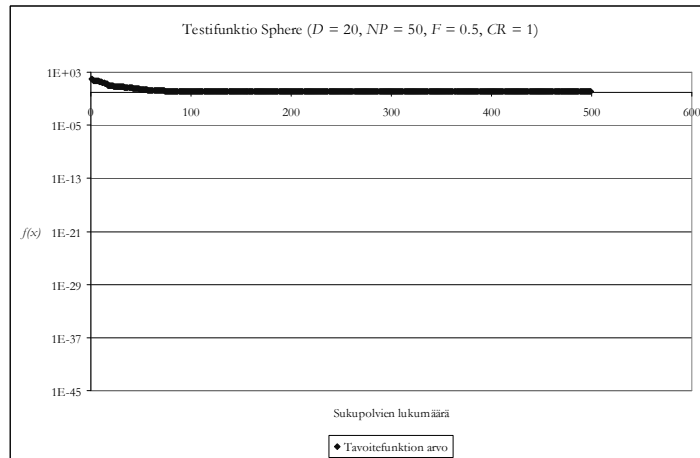
a)



b)



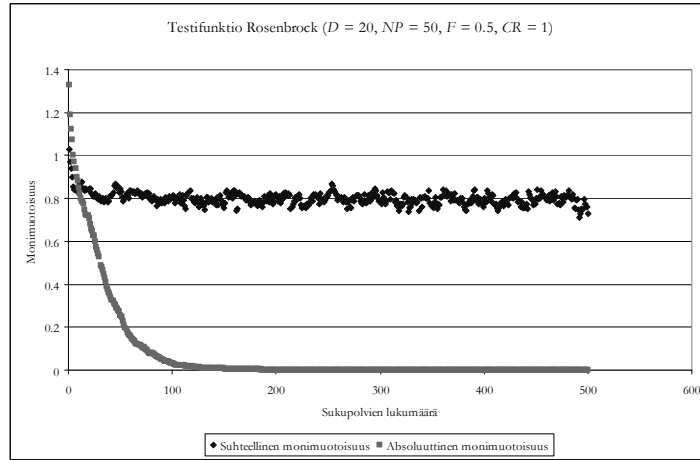
c)



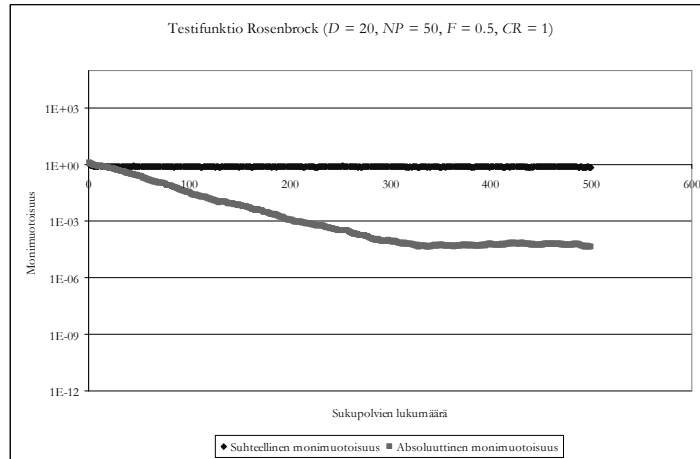
Kuva 22. a) suhteellinen ja absoluuttinen monimuotoisuus b) suhteellinen ja absoluuttinen monimuotoisuus puolilogaritmisella asteikolla c) tavoitefunktion arvo puolilogaritmisella asteikolla

(jatkuu)

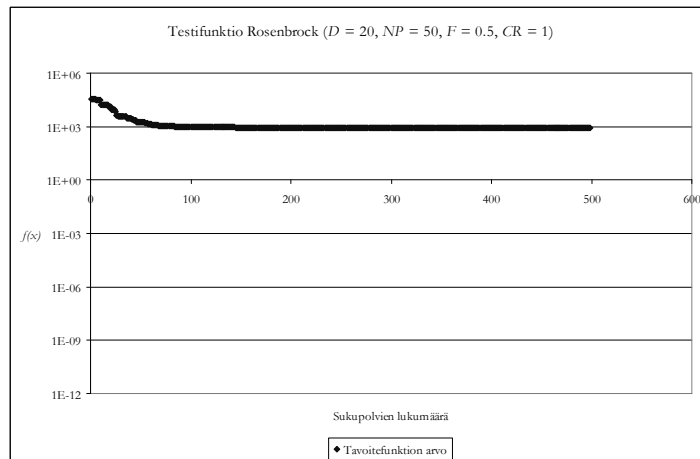
a)



b)

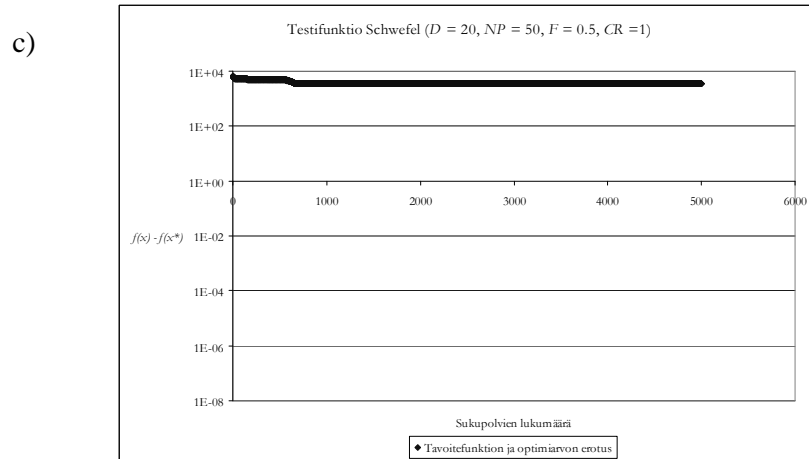
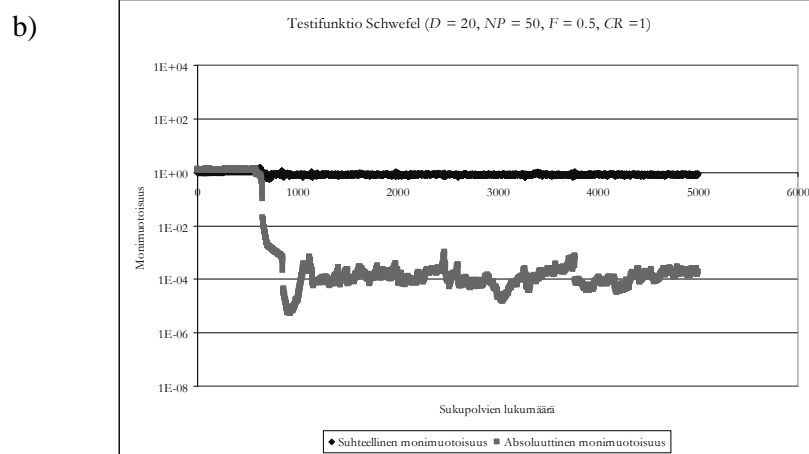
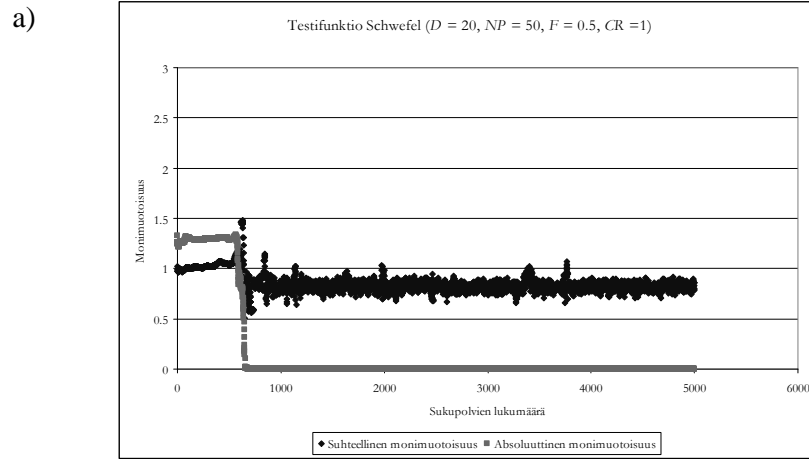


c)



Kuva 23. a) suhteellinen ja absoluuttinen monimuotoisuus b) suhteellinen ja absoluuttinen monimuotoisuus puolilogaritmisella asteikolla c) tavoitefunktion arvo puolilogaritmisella asteikolla

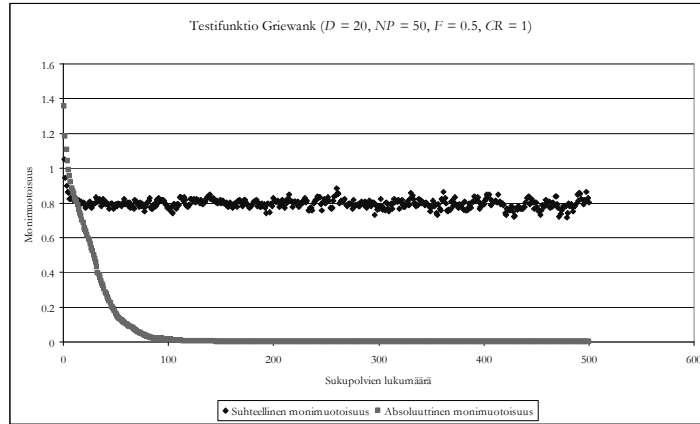
(jatkuu)



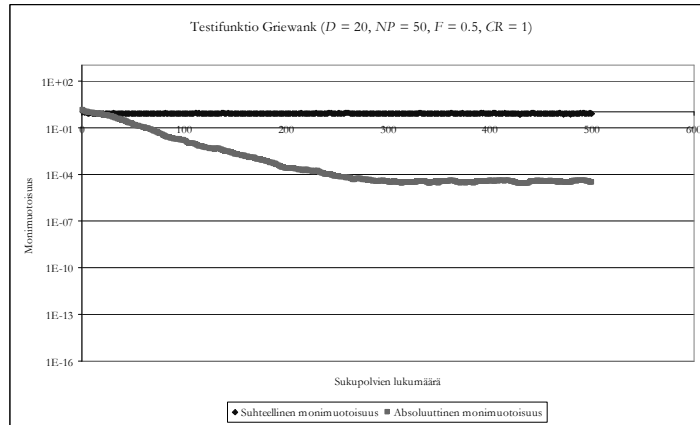
Kuva 24. a) suhteellinen ja absoluuttinen monimuotoisuus b) suhteellinen ja absoluuttinen monimuotoisuus puolilogaritmisella asteikolla c) tavoitefunktion arvon ja funktion optimiarvon erotus puolilogaritmisella asteikolla

(jatkuu)

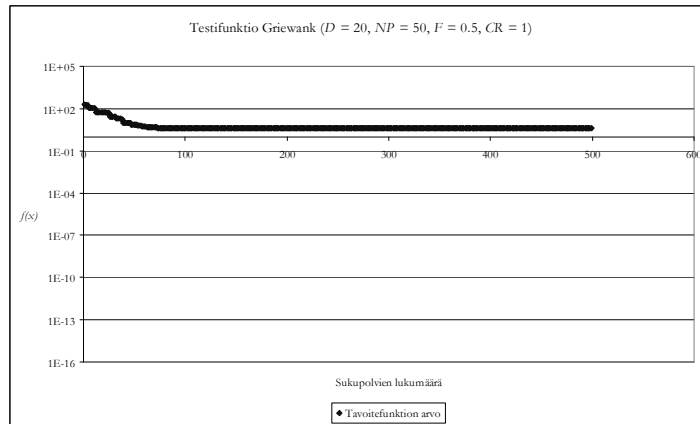
a)



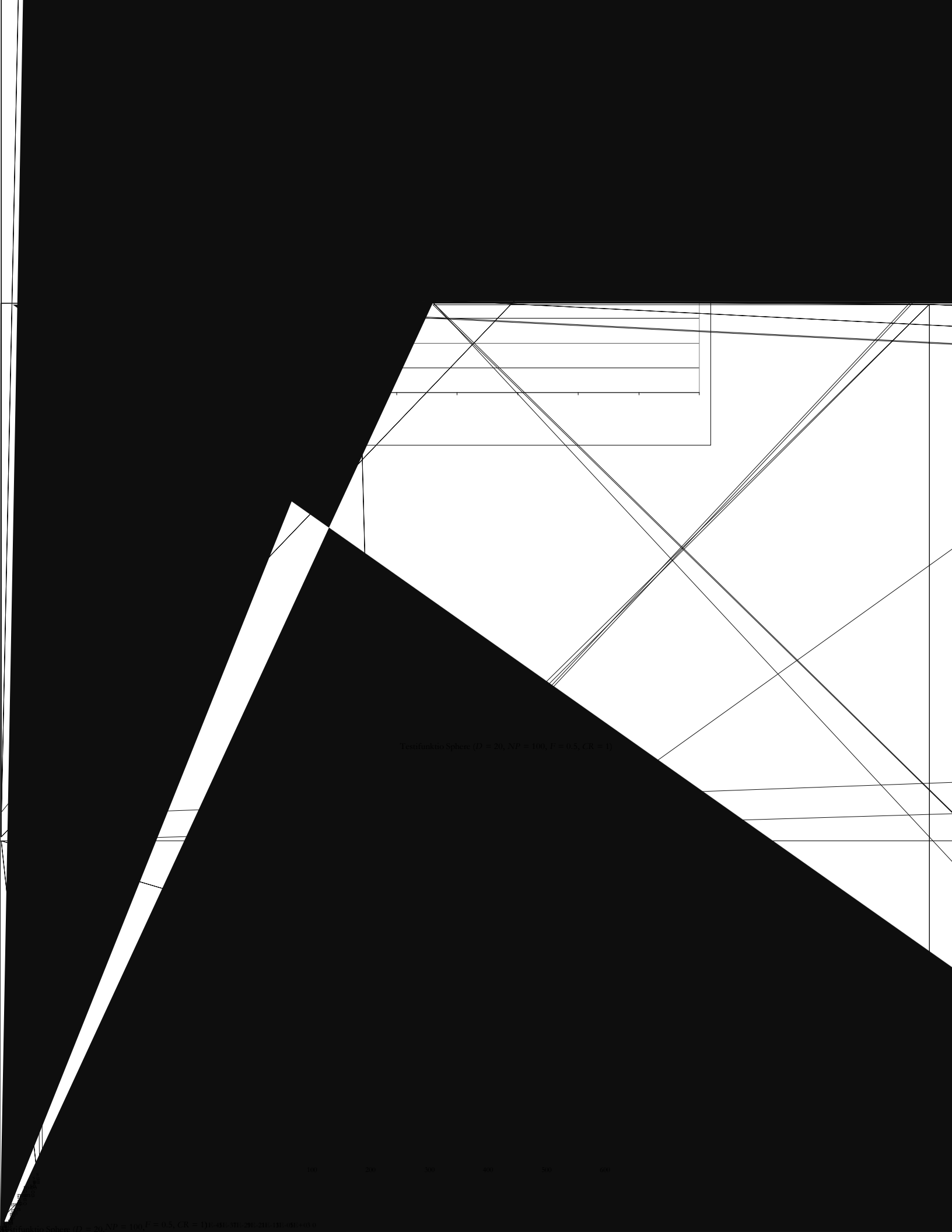
b)

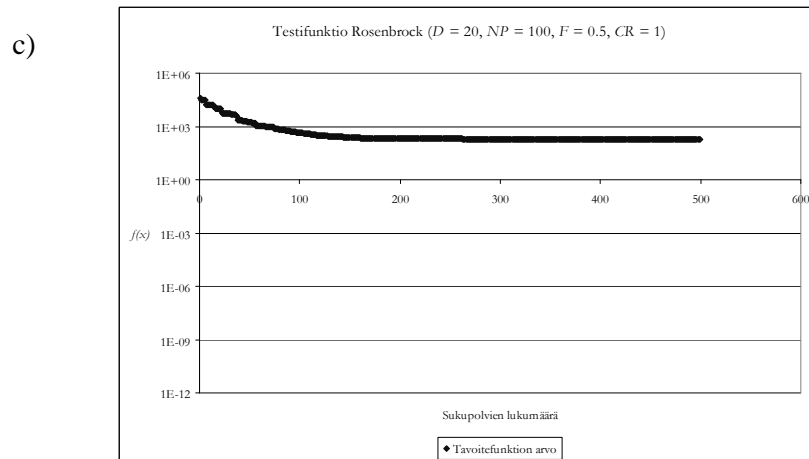
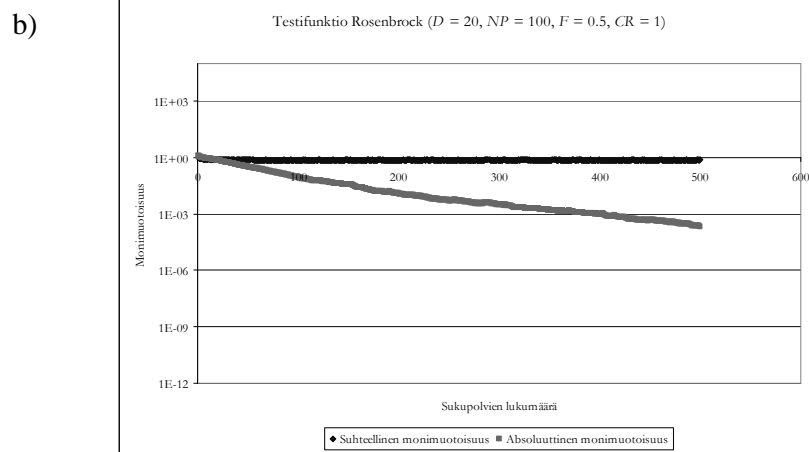
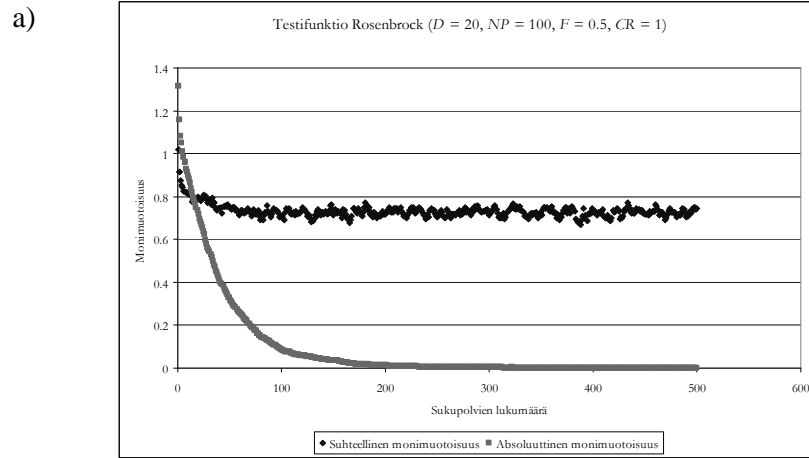


c)



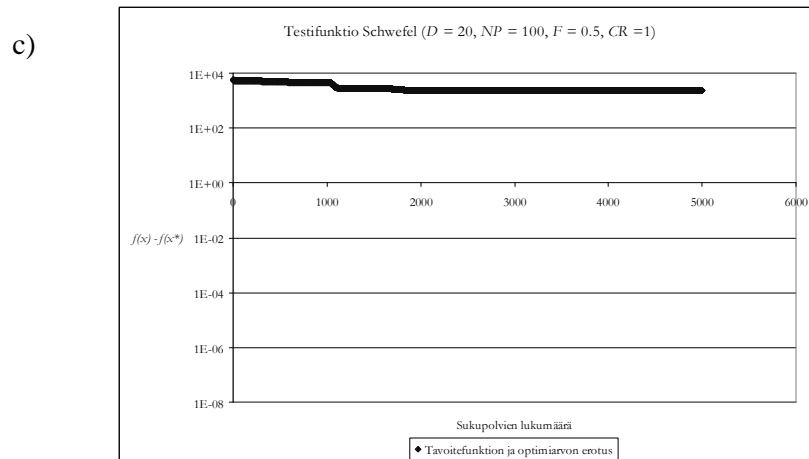
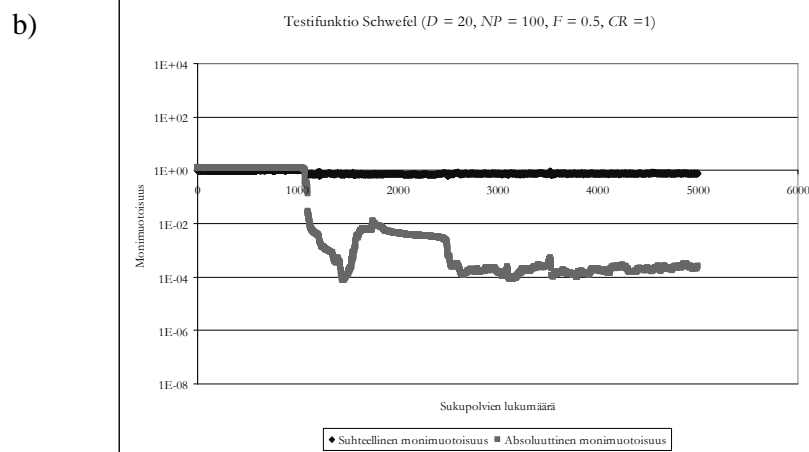
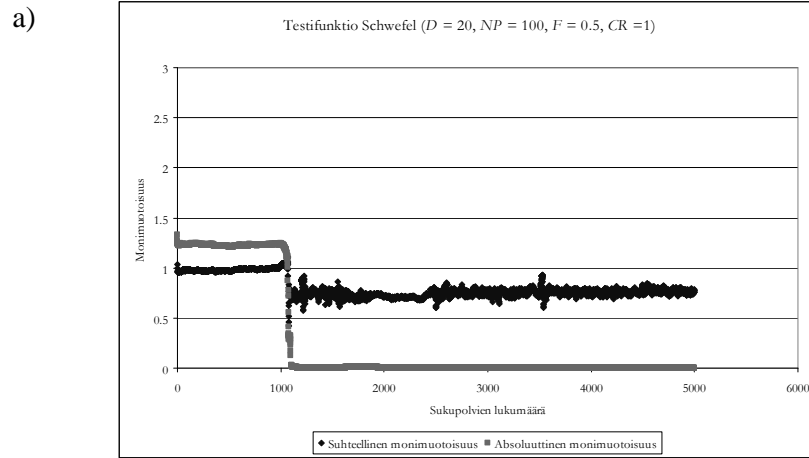
Kuva 25. a) suhteellinen ja absoluuttinen monimuotoisuus b) suhteellinen ja absoluuttinen monimuotoisuus puolilogaritmisella asteikolla c) tavoitefunktion arvo puolilogaritmisella asteikolla





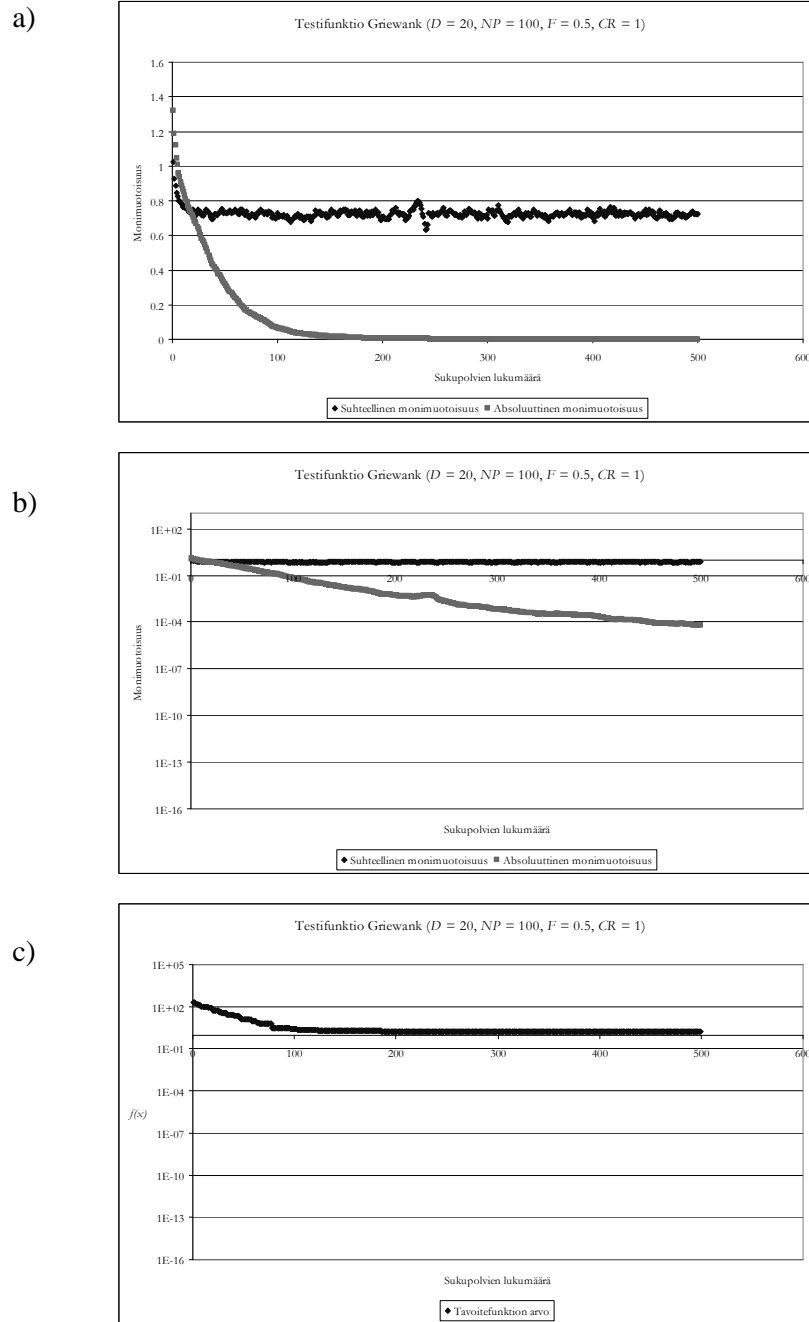
Kuva 27. a) suhteellinen ja absoluuttinen monimuotoisuus b) suhteellinen ja absoluuttinen monimuotoisuus puolilogaritmisella asteikolla c) tavoitefunktion arvo puolilogaritmisella asteikolla

(jatkuu)



Kuva 28. a) suhteellinen ja absoluuttinen monimuotoisuus b) suhteellinen ja absoluuttinen monimuotoisuus puolilogaritmisella asteikolla c) tavoitefunktion arvon ja funktion optimaarvon erotus puolilogaritmisella asteikolla

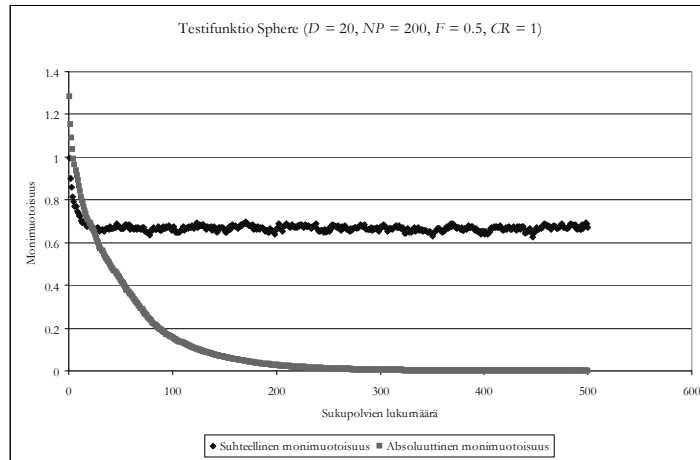
(jatkuu)



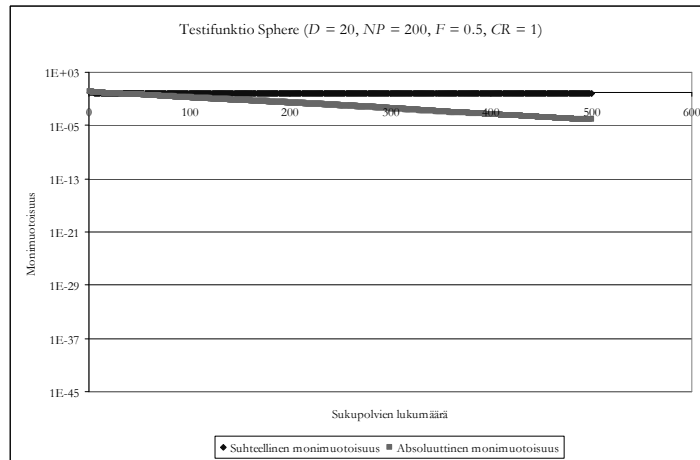
Kuva 29. a) suhteellinen ja absoluuttinen monimuotoisuus b) suhteellinen ja absoluuttinen monimuotoisuus puolilogaritmisella asteikolla c) tavoitefunktion arvo puolilogaritmisella asteikolla

Liite 6. Testitapaus 5: testifunktiot Sphere, Rosenbrock, Schwefel ja Griewank parametrien arvoilla $D = 20$, $NP = 200$, $CR = 1.0$ ja $F = 0.5$.

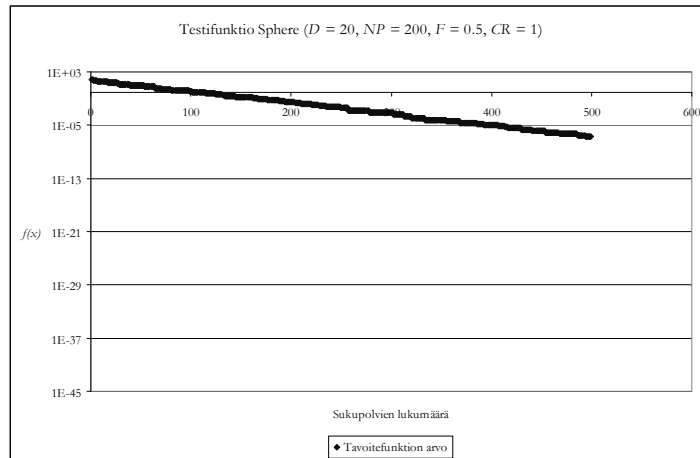
a)



b)



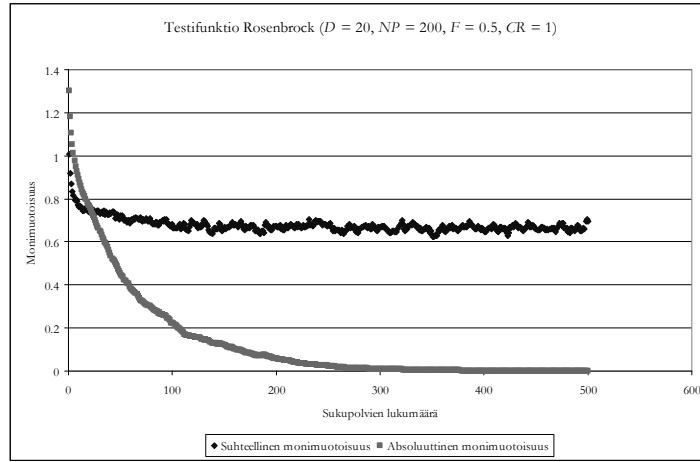
c)



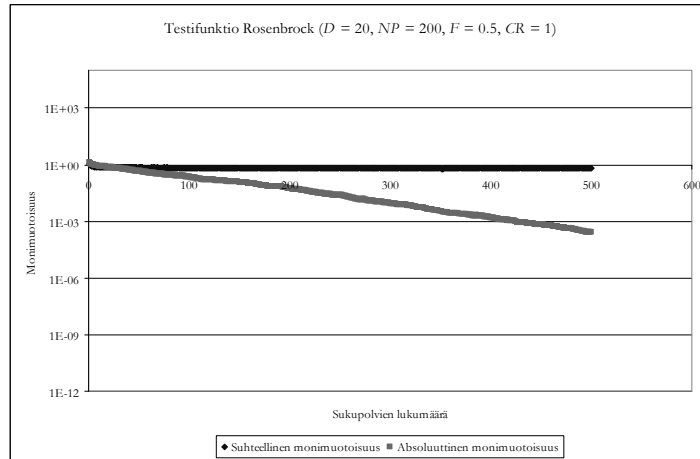
Kuva 30. a) suhteellinen ja absoluuttinen monimuotoisuus b) suhteellinen ja absoluuttinen monimuotoisuus puolilogaritmisella asteikolla c) tavoitefunktion arvo puolilogaritmisella asteikolla

(jatkuu)

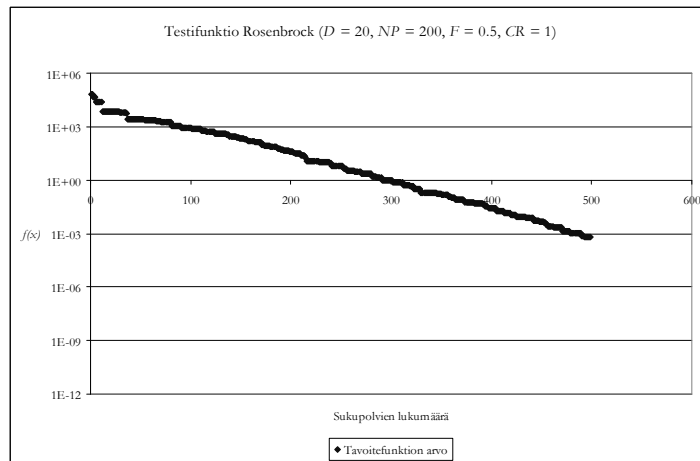
a)



b)

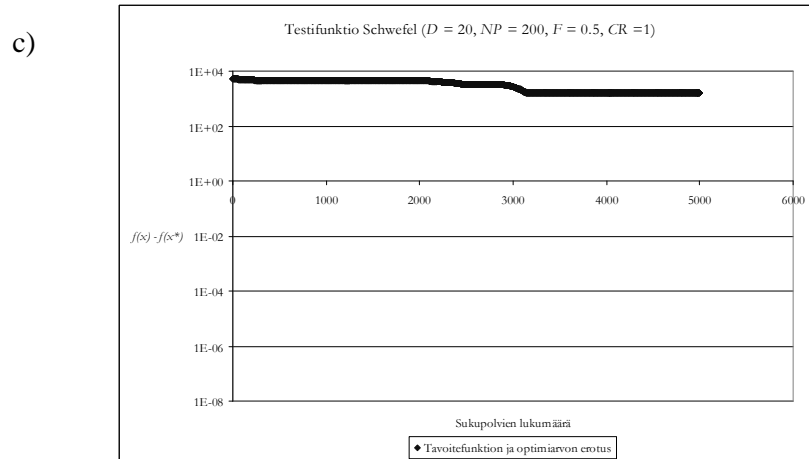
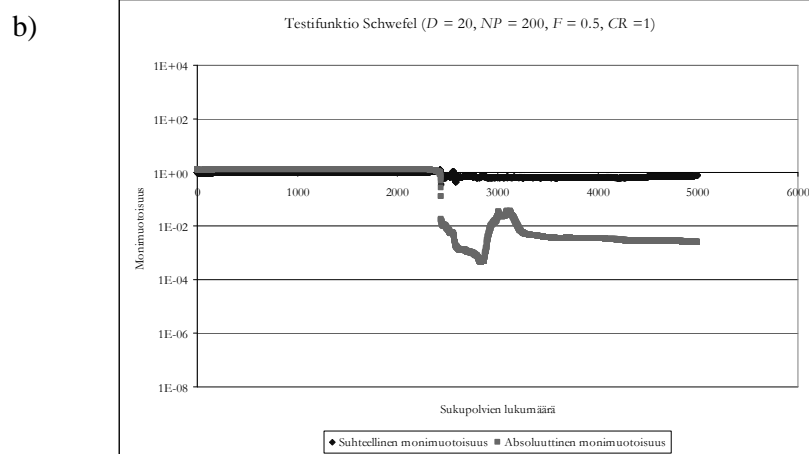
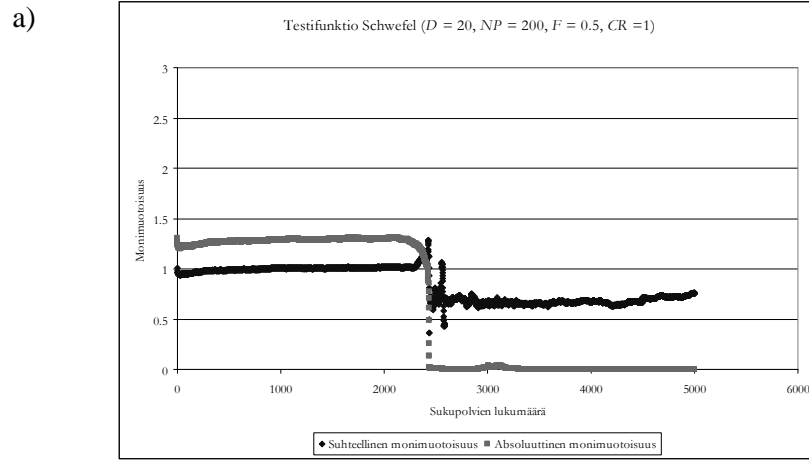


c)



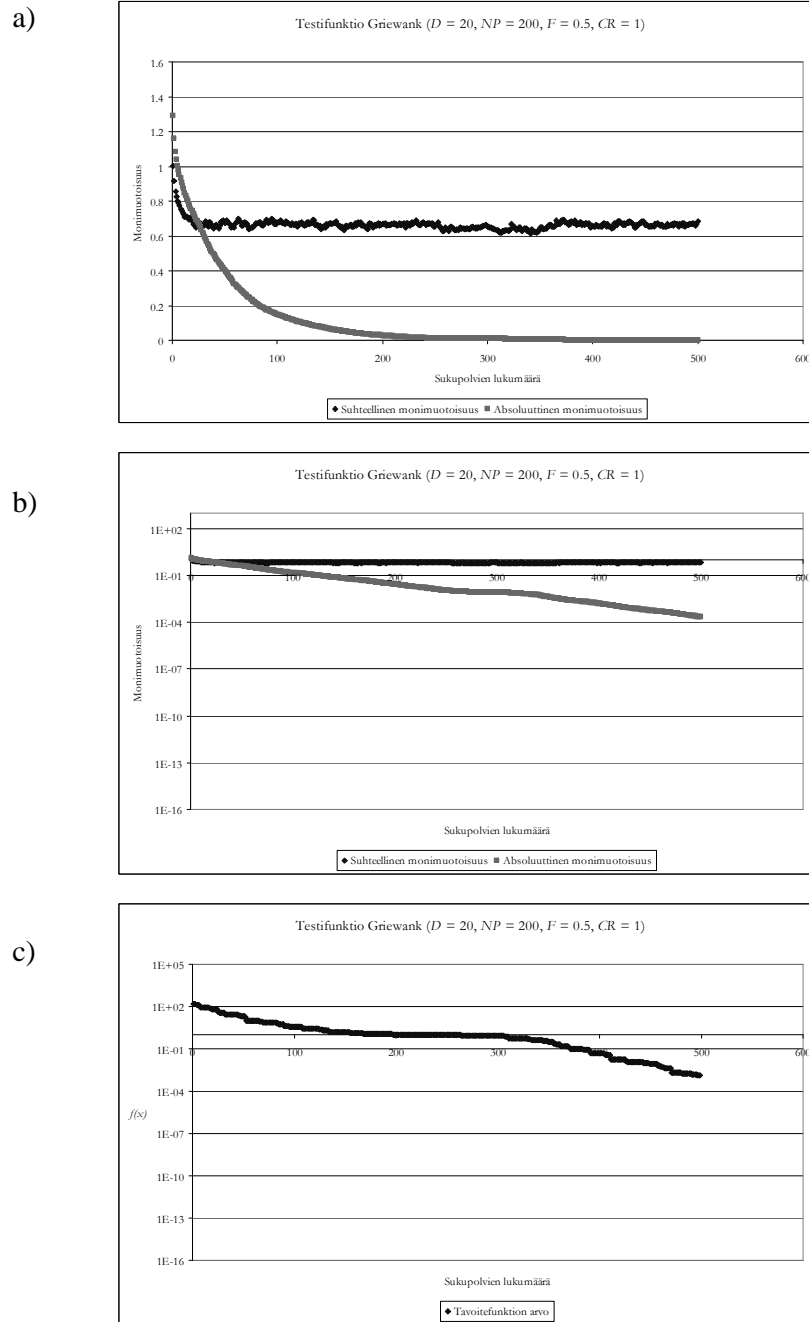
Kuva 31. a) suhteellinen ja absoluuttinen monimuotoisuus b) suhteellinen ja absoluuttinen monimuotoisuus puolilogaritmisella asteikolla c) tavoitefunktion arvo puolilogaritmisella asteikolla

(jatkuu)



Kuva 32. a) suhteellinen ja absoluuttinen monimuotoisuus b) suhteellinen ja absoluuttinen monimuotoisuus puolilogaritmisella asteikolla c) tavoitefunktion arvon ja funktion optimiarvon erotus puolilogaritmisella asteikolla

(jatkuu)



Kuva 33. a) suhteellinen ja absoluuttinen monimuotoisuus b) suhteellinen ja absoluuttinen monimuotoisuus puolilogaritmisella asteikolla c) tavoitefunktion arvo puolilogaritmisella asteikolla