

UNIVERZA V MARIBORU

FAKULTETA ZA ELEKTROTEHNIKO,
RAČUNALNIŠTVO IN INFORMATIKO

**Samoprilagajanje krmilnih parametrov
pri algoritmu diferencialne evolucije za
večkriterijsko optimizacijo**

MAGISTRSKO DELO

Maribor, julij 2008

Aleš Zamuda



UNIVERZA V MARIBORU



FAKULTETA ZA ELEKTROTEHNIKO,
RAČUNALNIŠTVO IN INFORMATIKO
2000 Maribor, Smetanova ulica 17

Številka: 1/64-podipl.-2008

Datum: 28. 04. 2008

TEMA ZA MAGISTRSKO DELO

V skladu s statutom Univerze v Mariboru je Senat Fakultete za elektrotehniko, računalništvo in informatiko na 10. seji dne 25. aprila 2008

s p r e j e l
temo za magistrsko delo

»SAMOPRILAGAJANJE KRMILNIH PARAMETROV PRI ALGORITMU DIFERENCIJALNE
EVOLUCIJE ZA VEČKRITERIJSKO OPTIMIZACIJO«,

ki ga bo izdelal
Aleš ZAMUDA, univ. dipl. inž. rač. in inf.

Mentor: izred. prof. dr. Janez BREST



PREJMEJO:

- dr. Brest
- A. Zamuda
- Univerza v Mariboru
- odloga

Avtor: Aleš Zamuda

Naslov: Samoprilagajanje krmilnih parametrov pri algoritmu diferencialne evolucije za večkriterijsko optimizacijo

UDK: 519.8:004.8

Ključne besede: evolucijski algoritmi, diferencialna evolucija, samoprilagajanje krmilnih parametrov, večkriterijska optimizacija

ZAHVALA

Zahvala mentorju izr. prof. dr. Janezu Brestu za številne napotke in vsakokratno pripravljenost pomagati ob nastalih preprekah v času študija in pri izdelavi magistrske naloge.

Posebna zahvala družini in Darji za moralno vzpodbudo.

Zahvala doc. dr. Bogdanu Filipiču za napotke med nastajanjem naloge. Zahvala Simonu Tuteku za pomoč pri zapisu enačb testnih funkcij v L^AT_EX kod. Zahvala Borku Boškoviću za konstruktivne predloge ob nastajanju naloge. Zahvala Sonji Bošković za pomoč pri lektoriranju naloge. Zahvala še mag. Sašu Greinerju in prof. dr. Viljemu Žumerju za izkazano pomoč.

Zahvala tudi nesebičnim avtorjem za objavo prosto dostopnega programskega koda osnovnih algoritmov, ki so bili uporabljeni pri izdelavi tega dela:

- Tei Tušar za program DEMO,
- Ling Huang za sestavo ogrodja za testiranje večkriterijskih problemov in Pon-nuthurai Nagaratnamu Suganthanu za razpoložljivost ob delu s tem ogrodjem,
- Carlosu Fonseci za objavo programa za izračun ploskev dosega,
- Joshui Knowlesu za izris dobljenih ploskev dosega in izračun indikatorja I_R ,
- Eckartu Zitzlerju za algoritem izračuna indikatorja hipervolumna,
- Mikeu Preussu, Eckartu Zitzlerju in Günterju Rudolphu za izračun metrike pokritosti podmnožic,
- vsej odprtokodni skupnosti za odprtost in dostopnost GNU/Linux orodij.

Zahvaljujem se še vsem drugim in vsakemu posamezniku, ki je na kakršenkoli način prispeval k izvedbi tega dela.

<i>Naslov:</i>	Samoprilagajanje krmilnih parametrov pri algoritmu diferencialne evolucije za večkriterijsko optimizacijo
<i>UDK:</i>	519.8:004.8
<i>Ključne besede:</i>	evolucijski algoritmi, diferencialna evolucija, samoprilagajanje krmilnih parametrov, večkriterijska optimizacija

POVZETEK

Predstavljamo nov algoritem diferencialne evolucije za večkriterijsko optimizacijo, ki ga krmili samoprilagoditveni mehanizem, predstavljen v evolucijskih strategijah.

Posebej se posvetimo načrtu algoritma, tj. matematično formalnemu zapisu glavnih sestavnih delov algoritma in medsebojni povezavi teh delov. Potek algoritma opišemo s psevdokodom. Sklepamo na časovno zahtevnost algoritma in podamo nekaj empiričnih meritev, da bi jo potrdili. Posebej preučimo tudi dinamiko samoprilagajanja krmilnih parametrov.

Navedemo sodobne testne probleme in indikatorje kakovosti iz literature za oceno zmogljivosti večkriterijskih optimizacijskih algoritmov. S pomočjo teh dobimo ocene zmogljivosti algoritma, ki pokažejo številna statistično signifikantna izboljšanja. Dobljene rezultate našega algoritma primerjamo tudi s sorodnimi algoritmi in na empiričnih rezultatih pokažemo, kje je algoritem statistično signifikantno boljši.

Title: Self-adaptation of Control Parameters in Differential Evolution for Multiobjective Optimization Algorithm

UDK: 519.8:004.8

Keywords: evolutionary algorithms, differential evolution, control parameters self-adaptation, multiobjective optimization

ABSTRACT

We present a new differential evolution algorithm for multiobjective optimization controlled by the self-adaptation mechanism introduced in evolution strategies.

Algorithm design is presented with mathematically formal notation of algorithm's main parts and their assembly. The algorithm is described using a pseudocode. Computational complexity of the algorithm is given and some empirical measurement are given for evidence. Self-adaptation dynamics of control parameters is also studied.

State of the art test problems and quality indicators from literature for performance assessment of multiobjective optimization algorithms are listed. Using these, performance assessments of the algorithm are obtained showing numerous statistically significant improvements. Obtained results with the algorithm are also compared with related algorithms and statistically significant differences of the compared algorithms are pointed out on empirical results.

Kazalo

1	Uvod	1
2	Sorodna dela	3
2.1	Večkriterijska optimizacija	3
2.1.1	Klasične optimizacijske metode	6
2.2	Evolucijski algoritmi	7
2.2.1	Diferencialna evolucija	9
2.2.2	Samoprilagajanje krmilnih parametrov	12
2.3	Evolucijski algoritmi za večkriterijsko optimizacijo	12
3	Diferencialna evolucija za večkriterijsko optimizacijo s samoprilagajanjem	14
3.1	Predstavitev samoprilagodljivih krmilnih parametrov	15
3.2	Spreminjanje samoprilagodljivih krmilnih parametrov	15
3.3	Zapis v psevdokodu	16
4	Rezultati	24
4.1	Primer zagona algoritma	24
4.2	Časovna analiza algoritma	26
4.2.1	Analitična ocena računske zahtevnosti algoritma	27
4.3	Analiza kakovosti algoritma	29
4.3.1	Testne funkcije	29
4.3.2	Metrike učinkovitosti	30
4.3.3	Rezultati simulacij	31
4.4	Statistična analiza rezultatov algoritmov	40
4.4.1	Algoritem DEMO	40
4.4.2	Algoritem MOSaDE	43
4.4.3	Algoritem GDE3	46
4.4.4	Algoritem NSGAII-SBX	49
4.4.5	Algoritem NSGAII-PCX	52

4.4.6	Algoritem MTS	55
4.4.7	Algoritem MO_DE	58
4.4.8	Algoritem MO_PSO	61
4.4.9	Povzetek rezultatov primerjav	64
4.5	Analiza dinamike samoprilagodljivih parametrov	65
5	Zaključek	71
A	Enovit psevdokod algoritma DEMOwSA	73
B	Testne funkcije CEC 2007	79
B.1	Novejše funkcije	81
B.2	Funkcije ZDT in DTLZ	81
B.3	Funkcije WFG	88
C	Ogrodje za paralelno procesiranje	91
C.1	Faza <i>test</i>	92
C.2	Faza <i>metrics</i>	92
C.3	Faza <i>latex</i>	94
D	Slovar	95
	Literatura	97
	Življjenjepis	109
	Bibliografija	110

Slike

2.1 Pojem dominantnosti na primeru večkriterijske funkcije $\mathbf{f}(\mathbf{x})$	4
3.1 Predstavitev samoprilagodljivih krmilnih parametrov in ovrednotenja v posamezniku	15
3.2 Prikaz algoritma za premešanje seznama na preprostem primeru	18
4.1 Aproksimacijske množice DEMOwSA po različnem številu ovrednotenj funkcije OKA2, z $RNi_{seed} = 1$ in $NP = 20$	25
4.2 Drevo klicev glavnih funkcij programa za 5000 ovrednotenj kriterijske funkcije WFG1_M5.	27
4.3 Ploskve 0%, 50% in 100% dosega vseh 25 zagonov na testni funkciji OKA2 po 5e+5 ovrednotenjih.	36
4.4 Ploskve 0%, 50% in 100% dosega vseh 25 zagonov na testni funkciji SYMPART po 5e+5 ovrednotenjih.	36
4.5 Ploskve 0%, 50% in 100% dosega vseh 25 zagonov na testni funkciji S_ZDT1 po 5e+5 ovrednotenjih.	36
4.6 Ploskve 0%, 50% in 100% dosega vseh 25 zagonov na testni funkciji S_ZDT2 po 5e+5 ovrednotenjih.	37
4.7 Ploskve 0%, 50% in 100% dosega vseh 25 zagonov na testni funkciji S_ZDT4 po 5e+5 ovrednotenjih.	37
4.8 Ploskve 0%, 50% in 100% dosega vseh 25 zagonov na testni funkciji R_ZDT4 po 5e+5 ovrednotenjih.	37
4.9 Ploskve 0%, 50% in 100% dosega vseh 25 zagonov na testni funkciji S_ZDT6 po 5e+5 ovrednotenjih.	38
4.10 Ploskve 50% dosega vseh 25 zagonov na testnih funkcijah 8–13 z $M = 3$ po 5e+5 ovrednotenjih.	38
4.11 Po kriterijih projicirane aproksimacijske množice pri zagonu z mediano vrednostjo indikatorja I_R po 5e+5 ovrednotenjih funkcij 12 in 13.	39

4.12 Na 25 generacijah povprečena vrednost povprečne izboljšane vrednosti krmilnega parametra F za 25 neodvisnih zagonov (funkcije 1–7 in DTLZ z $M = 3$).	67
4.13 Na 25 generacijah povprečena vrednost povprečne izboljšane vrednosti krmilnega parametra F za 25 neodvisnih zagonov (funkcije 8–13 in ostale z $M = 5$).	68
4.14 Na 25 generacijah povprečena vrednost povprečne izboljšane vrednosti krmilnega parametra CR za 25 neodvisnih zagonov (funkcije 1–7 in DTLZ z $M = 3$).	69
4.15 Na 25 generacijah povprečena vrednost povprečne izboljšane vrednosti krmilnega parametra CR za 25 neodvisnih zagonov (funkcije 8–13 in ostale z $M = 5$).	70
B.1 Ločene enakovredne Pareto množice za funkcijo SYM-PART.	82

Tabele

2.1	Diferencialna evolucija – kombinacije operatorjev križanja in mutacije ter njihovo poimenovanje v [1].	11
4.1	Časovna zahtevnost algoritma.	27
4.2	Ocene računske zahtevnosti algoritma DEMOwSA.	28
4.3	Čas izvajanja algoritma DEMOwSA pri različnih <i>MAXFEs</i>	28
4.4	Čas izvajanja algoritma DEMOwSA pri različnih <i>NP</i>	29
4.5	Vrednosti indikatorja pokritosti množic za testno funkcijo SYM-PART.	32
4.6	Vrednosti indikatorja I_R na testnih funkcijah 1–7.	34
4.7	Vrednosti indikatorja I_R na testnih funkcijah 8–13 z $M = 3$	34
4.8	Vrednosti indikatorja I_R na testnih funkcijah 8–13 z $M = 5$	34
4.9	Vrednosti indikatorja $I_{\overline{H}}$ na testnih funkcijah 1–7.	35
4.10	Vrednosti indikatorja $I_{\overline{H}}$ na testnih funkcijah 8–13 z $M = 3$	35
4.11	Vrednosti indikatorja $I_{\overline{H}}$ na testnih funkcijah 8–13 z $M = 5$	35
4.12	Razlika vrednosti indikatorja I_R za algoritma DEMO in DEMOwSA na testnih funkcijah 1–7.	41
4.13	Razlika vrednosti indikatorja I_R za algoritma DEMO in DEMOwSA na testnih funkcijah 8–13 z $M = 3$	41
4.14	Razlika vrednosti indikatorja I_R za algoritma DEMO in DEMOwSA na testnih funkcijah 8–13 z $M = 5$	41
4.15	Razlika vrednosti indikatorja $I_{\overline{H}}$ za algoritma DEMO in DEMOwSA na testnih funkcijah 1–7.	42
4.16	Razlika vrednosti indikatorja $I_{\overline{H}}$ za algoritma DEMO in DEMOwSA na testnih funkcijah 8–13 z $M = 3$	42
4.17	Razlika vrednosti indikatorja $I_{\overline{H}}$ za algoritma DEMO in DEMOwSA na testnih funkcijah 8–13 z $M = 5$	42
4.18	Razlika vrednosti indikatorja I_R za algoritma MOSaDE in DEMOwSA na testnih funkcijah 1–7.	44
4.19	Razlika vrednosti indikatorja I_R za algoritma MOSaDE in DEMOwSA na testnih funkcijah 8–13 z $M = 3$	44

4.20 Razlika vrednosti indikatorja I_R za algoritma MOSaDE in DEMOwSA na testnih funkcijah 8–13 z $M = 5$	44
4.21 Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma MOSaDE in DEMOwSA na testnih funkcijah 1–7.	45
4.22 Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma MOSaDE in DEMOwSA na testnih funkcijah 8–13 z $M = 3$	45
4.23 Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma MOSaDE in DEMOwSA na testnih funkcijah 8–13 z $M = 5$	45
4.24 Razlika vrednosti indikatorja I_R za algoritma GDE3 in DEMOwSA na testnih funkcijah 1–7.	47
4.25 Razlika vrednosti indikatorja I_R za algoritma GDE3 in DEMOwSA na testnih funkcijah 8–13 z $M = 3$	47
4.26 Razlika vrednosti indikatorja I_R za algoritma GDE3 in DEMOwSA na testnih funkcijah 8–13 z $M = 5$	47
4.27 Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma GDE3 in DEMOwSA na testnih funkcijah 1–7.	48
4.28 Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma GDE3 in DEMOwSA na testnih funkcijah 8–13 z $M = 3$	48
4.29 Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma GDE3 in DEMOwSA na testnih funkcijah 8–13 z $M = 5$	48
4.30 Razlika vrednosti indikatorja I_R za algoritma NSGAII-SBX in DEMOwSA na testnih funkcijah 1–7.	50
4.31 Razlika vrednosti indikatorja I_R za algoritma NSGAII-SBX in DEMOwSA na testnih funkcijah 8–13 z $M = 3$	50
4.32 Razlika vrednosti indikatorja I_R za algoritma NSGAII-SBX in DEMOwSA na testnih funkcijah 8–13 z $M = 5$	50
4.33 Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma NSGAII-SBX in DEMOwSA na testnih funkcijah 1–7.	51
4.34 Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma NSGAII-SBX in DEMOwSA na testnih funkcijah 8–13 z $M = 3$	51
4.35 Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma NSGAII-SBX in DEMOwSA na testnih funkcijah 8–13 z $M = 5$	51
4.36 Razlika vrednosti indikatorja I_R za algoritma NSGAII-PBX in DEMOwSA na testnih funkcijah 1–7.	53
4.37 Razlika vrednosti indikatorja I_R za algoritma NSGAII-PBX in DEMOwSA na testnih funkcijah 8–13 z $M = 3$	53
4.38 Razlika vrednosti indikatorja I_R za algoritma NSGAII-PBX in DEMOwSA na testnih funkcijah 8–13 z $M = 5$	53

4.39 Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma NSGAII-PBX in DEMoWSA na testnih funkcijah 1–7	54
4.40 Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma NSGAII-PBX in DEMoWSA na testnih funkcijah 8–13 z $M = 3$	54
4.41 Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma NSGAII-PBX in DEMoWSA na testnih funkcijah 8–13 z $M = 5$	54
4.42 Razlika vrednosti indikatorja I_R za algoritma MTS in DEMoWSA na testnih funkcijah 1–7	56
4.43 Razlika vrednosti indikatorja I_R za algoritma MTS in DEMoWSA na testnih funkcijah 8–13 z $M = 3$	56
4.44 Razlika vrednosti indikatorja I_R za algoritma MTS in DEMoWSA na testnih funkcijah 8–13 z $M = 5$	56
4.45 Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma MTS in DEMoWSA na testnih funkcijah 1–7	57
4.46 Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma MTS in DEMoWSA na testnih funkcijah 8–13 z $M = 3$	57
4.47 Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma MTS in DEMoWSA na testnih funkcijah 8–13 z $M = 5$	57
4.48 Razlika vrednosti indikatorja I_R za algoritma MO_DE in DEMoWSA na testnih funkcijah 1–7	59
4.49 Razlika vrednosti indikatorja I_R za algoritma MO_DE in DEMoWSA na testnih funkcijah 8–13 z $M = 3$	59
4.50 Razlika vrednosti indikatorja I_R za algoritma MO_DE in DEMoWSA na testnih funkcijah 8–13 z $M = 5$	59
4.51 Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma MO_DE in DEMoWSA na testnih funkcijah 1–7	60
4.52 Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma MO_DE in DEMoWSA na testnih funkcijah 8–13 z $M = 3$	60
4.53 Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma MO_DE in DEMoWSA na testnih funkcijah 8–13 z $M = 5$	60
4.54 Razlika vrednosti indikatorja I_R za algoritma MO_PSO in DEMoWSA na testnih funkcijah 1–7	62
4.55 Razlika vrednosti indikatorja I_R za algoritma MO_PSO in DEMoWSA na testnih funkcijah 8–13 z $M = 3$	62
4.56 Razlika vrednosti indikatorja I_R za algoritma MO_PSO in DEMoWSA na testnih funkcijah 8–13 z $M = 5$	62
4.57 Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma MO_PSO in DEMoWSA na testnih funkcijah 1–7	63

4.58 Razlika vrednosti indikatorja $I_{\overline{H}}$ za algoritma MO_PSO in DEMOwSA na testnih funkcijah 8–13 z $M = 3$	63
4.59 Razlika vrednosti indikatorja $I_{\overline{H}}$ za algoritma MO_PSO in DEMOwSA na testnih funkcijah 8–13 z $M = 5$	63
4.60 Povzetek rezultatov primerjanih algoritmov pri 5e+5 ovrednotenjih kriterijske funkcije.	64
4.61 Vrednosti krmilnih parametrov po 5e+5 ovrednotenjih kriterijske funkcije.	65
B.1 Nastavitev parametrov v testnih funkcijah [2].	80

Seznam algoritmov

1	Ovrednoti _{OKA2} : primer testne funkcije OKA2.	17
2	Primerjaj rešitvi.	17
3	Premešaj populacijo.	18
4	Incializiraj nov primerek.	19
5	Nov primerek _{DEMOwSA/rand/1/bin}	20
6	Poreži _{SPEA-II} .	21
7	Odstrani prenakopičene rešitve _{SPEA-II} .	21
8	Nakopičenost _{SPEA-II} .	22
9	Diferencialna evolucija za večkriterijsko optimizacijo s samoprilagajanjem.	23

Uporabljeni simboli

F	faktor skaliranja (nadzoruje stopnjo mutacije algoritma DE)
CR	faktor križanja (nadzoruje stopnjo križanja algoritma DE)
NP	velikost populacije
\mathbf{x}	iskalni parametri
x_i	i -ti iskalni parameter
\mathbf{a}	kodiranje poljubnega posameznika iz populacije
\mathbf{s}	kodirani krmilni parametri posameznika
$\mathbf{f}(\mathbf{x})$	večkriterijska funkcija z vektorskim parametrom \mathbf{x}
D	dimenzija prostora spremenljivk
M	dimenzija prostora kriterijev
\mathbf{I}	množica vseh parametrov, vsebovanih v kriterijski funkciji
$G(g)$	generacija (števec)
$g_i(\mathbf{x})$	omejitve kriterijske funkcije
P	Pareto množica
w_i	uteži pri metodi utežene vsote
ϵ	okolica (npr. enakosti pretvorjenih enakostnih omejitev)
$\mathbf{v}_{i,G+1}$	mutiran vektor
r_1, r_2, r_3	naključni indeksi
$\mathbf{x}_{best,G}$	najboljši posameznik v populaciji generacije G (pri enokriterijski optimizaciji)
$u_{i,j,G+1}$	poskusni vektor
$\text{rand}(0, 1)$	uniformno naključno porazdeljeno število
j	naključni indeks pri križanju v algoritmu DE
L	tekovični indeks pri križanju v algoritmu DE
e	Eulerjevo število
$N(0, 1)$	normalno porazdeljena naključna spremenljivka
τ	stopnja učenja
ND	nedominiranost
d_{new}, d_{parent}	števili dominiranih primerkov

\mathbf{P}_g	populacija primerkov v generaciji g
$\mathbf{p}_{g,i}$	i -ti primerek iz populacije g
RNi	trenutno stanje generatorja enakomerno porazdeljenih naključnih števil
$func_type$	zaporedna številka funkcije, ki jo optimiziramo
sel_type_{MO}	strategija večkriterijske selekcije (strategija sekanja)
s_{DE}	številka strategije mutacije algoritma DE (strategija preiskovanja)
RNj	seme generatorja normalno porazdeljenih naključnih števil
F_{init}	začetna vrednost parametrov F
CR_{init}	začetna vrednost parametrov CR
$F_{min} (F_{max})$	najmanjša (največja) možna vrednost parametra F
$CR_{min} (CR_{max})$	najmanjša (največja) možna vrednost parametra CR
front_gen	množica intervalov vzorčenj populacije \mathbf{P}_g po FE
FEs	število ovrednotenj kriterijske funkcije
$MAXFEs$	največje dovoljeno število ovrednotenj kriterijske funkcije
T	čas zagona algoritma
$I_{\overline{H}}$	indikator hipervolumna
I_R	indikator R
Z	zvezni prostor kriterijev
Ω	diskretiziran prostor kriterijev
RNi_{seed}	seme psevdonaključnega generatorja števil
\overline{N}	velikost arhiva nedominiranih (elitnih) rešitev

Uporabljene kratice

CI	<i>Computational Intelligence</i>
EC	<i>Evolutionary Computation</i>
DE	<i>Differential Evolution</i>
ES	<i>Evolution Strategies</i>
PSO	<i>Particle Swarm Optimization</i>
ACO	<i>Ant Colony Optimization</i>
MO	<i>Multiobjective Optimization</i>
MOP	<i>Multiobjective Optimization Problem</i>
MOEA	<i>Multiobjective Optimization Evolutionary Algorithm</i>
ZDT	<i>Zitzler, Deb, Thiele</i>
DTLZ	<i>Deb, Thiele, Laumanns, Zitzler</i>
WFG	<i>Walking Fish Group</i>
CS	<i>Covered Sets</i>

Poglavlje 1

Uvod

Pri poskušanju boljšega razumevanja dobrih in slabih lastnosti poljubnega algoritma je pomembno dobro razumeti problem, ki ga algoritmom rešuje. To je res tako za področje večkriterijskih evolucijskih algoritmov kot tudi za vsako drugo področje. Da bi bolje razumeli algoritme za večkriterijsko optimizacijo, je potrebno poznati razloge, ki pripeljejo do potrebe po njej. Eden od razlogov je, da se v praksi pogosto srečamo z odločanjem, ki zahteva izbiro ene izmed različnih danih možnosti. To lahko opravimo s pomočjo kriterijev, ki si jih postavimo za te možnosti. Vendar pa so si kriteriji pogosto v nasprotju, kar pomeni, da izboljšanje neke odločitve po enem kriteriju povzroči njeno poslabšanje po drugih kriterijih. Takrat nimamo opravka samo z eno optimalno možno rešitvijo oz. odločitvijo, temveč z množico optimalnih rešitev. Za izbiro najugodnejših rešitev problema pogosto želimo dobro spoznati, kakšne možnosti sploh imamo, oz. kakšne najboljše možnosti imamo ter se šele kasneje odločiti za eno od njih. Ta pristop opisuje večkriterijsko optimizacijo, tj. sočasno optimizacijo medsebojno konfliktnih kriterijev.

Glavna cilja magistrskega dela sta (1) predstaviti izboljšan algoritmom za večkriterijsko optimizacijo, ki v povprečju dosega boljše rezultate, kot jih dosegajo nekateri obstoječi algoritmi in (2) vgradili samoprilagajanje nekaterih krmilnih parametrov, tako da jih uporabniku ni potrebno več nastavljati, in s tem povečati robustnost algoritma za kasnejšo uporabo na novih optimizacijskih problemih.

V delu predstavljamo algoritmom za reševanje večkriterijskih optimizacijskih problemov, ki je po kakovosti dobljenih rešitev primerljiv z najsodobnejšimi konkurenčnimi algoritmi. Algoritmom temelji na preiskovanju z evolucijskim algoritmom diferencialne evolucije. Ta vsebuje populacijo primerkov, ki skozi evolucijski proces s stohastičnim vzorčenjem kriterijske funkcije iščejo optimalne vrednosti iskalnih parametrov. V posamezne vektorje diferencialne evolucije za večkriterijsko optimizacijo smo dodali mehanizem samoprilagajanja. Tako smo dobili algoritmom, ki izkazuje statistično signifikantne izboljšave kakovosti na številnih aktualnih testnih

funkcijah.

V nadaljevanju podajamo sorodna dela, od večkriterijske optimizacije, preko evolucijskih algoritmov do evolucijskih algoritmov za večkriterijsko optimizacijo. Po podanem teoretičnem ozadju podamo načrt novega algoritma v tretjem poglavju in podamo njegov psevdokod. Rezultati sledijo v četrtem poglavju, kjer najprej navedemo preprost zagon algoritma in nadaljujemo s časovno analizo algoritma. Podamo tudi oceno kakovosti algoritma, z izračunom indikatorjev za testirane funkcije in izrisom ploskev dosega. Poglavlje eksperimentalnih rezultatov zaključimo s statistično primerjavo kakovosti algoritma in analizo dinamike samoprilagodljivih parametrov, ki pokažeta prednosti in slabosti algoritma ter obnašanje mehanizma samoprilagajanja. Sledi zaključek z napotki za nadaljnje delo. V dodatkih zberemo še enovit psevdokod predstavljenega algoritma, uporabljane testne funkcije, opis ogrodja za izvedbo eksperimentov in krajši slovar. Na koncu so navedeni še literatura, življenjepis in bibliografija.

Poglavlje 2

Sorodna dela

Delo spada na področje umetne inteligece (*Artificial Intelligence*) oz. računske inteligece (*Computational Intelligence*) [3], v ožjem smislu pa na področje evolucijskega računanja (*Evolutionary Computation – EC*). V tem poglavju predstavljamo osnovne gradnike, ki dajejo temelje temu delu. Najprej predstavljamo pojem večkriterijske optimizacije in nekaj z njo povezanih pojmov, nato podajamo pristope za njeno reševanje. Sledi razčlenba evolucijskih algoritmov in pregled nekaj njihovih primerov za reševanje večkriterijskih problemov. Posebej osvetlimo algoritom diferencialne evolucije in njegove evolucijske operatorje in samoprilagajanje.

2.1 Večkriterijska optimizacija

Da bomo lažje razumeli, kaj je večkriterijska optimizacija (ang. *Multiobjective Optimization – MO*), definirajmo njen cilj. Cilj MO je reševanje večkriterijskega optimacijskega problema (ang. *Multiobjective Optimization Problem – MOP*). MOP definiramo kot iskanje dopustnega vektorja spremenljivk oziroma iskalnih parametrov $\mathbf{x} = (x_1, x_2, \dots, x_D)$, ki optimizira (minimizira) vektorsko funkcijo $\mathbf{f}(\mathbf{x})$:

$$\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})).$$

Pri tem vsakemu vektorju spremenljivk \mathbf{x} pripada njegov **kriterijski vektor** $\mathbf{f}(\mathbf{x})$, ki ga v prostoru kriterijev dobimo z ovrednotenjem rešitve iz prostora spremenljivk. Kriterijski vektor \mathbf{x} pri tem mora zadoščati m omejitvam:

$$g_i(\mathbf{x}) \geq 0; \quad i = 1, 2, \dots, m,$$

s katerimi podamo enostavne omejitve. Omejitve $g_i(\mathbf{x})$ lahko določajo npr. le definijsko območje in zalogo vrednosti optimizirane funkcije:

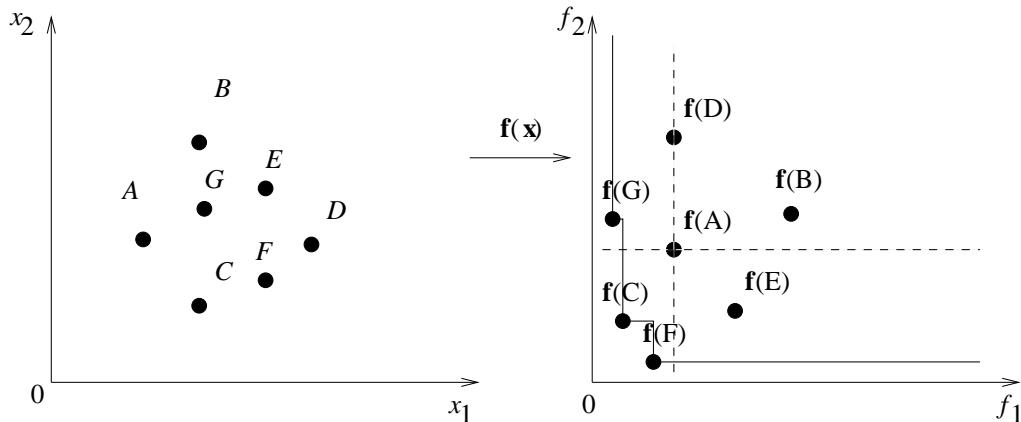
$$\mathbf{g}(\mathbf{x}) \geq 0 \Leftrightarrow (x_{1,min}, x_{1,max}, \dots, x_{D,min}, x_{D,max}, f_{1,min}, f_{1,max}, \dots, f_{M,min}, f_{M,max}).$$

Ob tem lahko povemo, da lahko vsako maksimizacijo kriterijev pretvorimo na problem minimizacije, tako da kriterije množimo z -1 . Od tu naprej bomo na optimizacijo vedno gledali kot problem minimizacije.

Definirajmo še pojem dominiranja. Rešitev večkriterijskega optimizacijskega problema \mathbf{x} **dominira** rešitev \mathbf{y} ($\mathbf{x} \preceq \mathbf{y}$), če velja:

1. rešitev \mathbf{x} ni slabša od rešitve \mathbf{y} po vseh kriterijih ($f_i(\mathbf{x}) \leq f_i(\mathbf{y}), \forall i = 1, \dots, M$),
2. rešitev \mathbf{x} je boljša od rešitve \mathbf{y} po vsaj enim kriteriju ($\exists j \in \{1, \dots, M\}: f_j(\mathbf{x}) < f_j(\mathbf{y})$).

Prostor kriterijev \Re^M (M je število kriterijev) je z relacijo \preceq delno urejen, saj nekatere rešitve medsebojno niso primerljive. Za slikovno razjasnitev pojma dominantnosti glej sliko 2.1. Kot je videti iz slike, je rešitev A boljša od B in D ter neprimerljiva z E .



Slika 2.1: Pojem dominantnosti na primeru večkriterijske funkcije $\mathbf{f}(\mathbf{x})$.

Pareto optimalna rešitev je takšna rešitev \mathbf{x} , ki je ne dominira nobena druga dopustna rešitev \mathbf{z} in bi veljalo $\mathbf{f}(\mathbf{z}) \preceq \mathbf{f}(\mathbf{x})$.

Množica nedominiranih rešitev v množici rešitev P je množica vseh tistih rešitev, ki jih ne dominira nobena rešitev iz množice P . Kriterijski vektorji rešitev iz množice nedominiranih rešitev tvorijo **nedominirano fronto**.

Množico nedominiranih rešitev celotnega prostora dopustnih rešitev imenujemo **Pareto optimalna množica**, njene elemente **Pareto optimalne rešitve** in pripadajočo nedominirano fronto v prostoru kriterijev imenujemo **Pareto optimalna**



fronta. Na sliki 2.1 tvorijo množico nedominiranih rešitev rešitve G , C in F , saj niso dominirane od nobene druge rešitve, tj. njihovi kriterijski vektorji $\mathbf{f}(G)$, $\mathbf{f}(C)$ in $\mathbf{f}(F)$ imajo manjše vrednosti od ostalih po vsaj enem kriteriju.

Pristope večkriterijskega optimiziranja lahko, predvsem iz zgodovinskih razlogov, ločimo na dve vrsti. Prvi, t.i. **prednostni pristop**, večkriterijsko funkcijo pretvori v enokriterijsko (npr. s postavitvijo uteži w_i , $f(\mathbf{x}) = \sum_i w_i f_i(\mathbf{x})$). Reši enokriterijski pristop in postopek ponovi, z drugačno postavitvijo uteži (w_i). Drugi, t.i. **idealni pristop**, najprej poišče množico optimalnih rešitev. Nato iz nje sami izberemo rešitev, ki nam najbolj ustreza. Izbiro najustreznejše rešitve naredimo na podlagi dodatne informacije o pomembnosti kriterijev, ki navadno ni znana vnaprej, zato je idealni pristop bolj pregleden in manj subjektiven od prednostnega pristopa. Če poznamo informacijo o kriterijih, ki nam omogoča ciljno uporabiti prednostni pristop, ni nobenega razloga, da bi namesto tega uporabljali idealni pristop.

Pri reševanju z idealnim pristopom želimo, da večkriterijska optimizacijska metoda najde čim več Pareto optimalnih rešitev. Ker bomo izmed teh rešitev (s pomočjo informacije o pomembnosti kriterijev) izbirali najboljšo, si želimo, da so dobljene rešitve kar se da enakomerno razporejene po prostoru kriterijev. Nalogu večkriterijskega optimiziranja lahko tako pretvorimo v nalogu iskanja množice nedominiranih rešitev, ki jo tu imenujemo **aproksimacijska množica** (ang. *approximation set*). Za elemente aproksimacijske množice si želimo, da velja:

- da so čim bliže Pareto optimalni fronti in
- da so enakomerno razporejeni vzdolž Pareto optimalne fronte.

Ta dva cilja večkriterijske optimizacije pa si pogosto nasprotujeta. Zato je že sam proces večkriterijske optimizacije lahko večkriterijski.

Pogosto zanemarjen, tretji cilj algoritmov večkriterijske optimizacije je čim krajši čas računanja za doseg prvih dveh ciljev večkriterijske optimizacije [4]. Z algoritmom večkriterijske optimizacije želimo doseči tudi čim večjo robustnost ob različnih zagonih algoritma [5, 6], tj. da algoritom s čim večjo verjetnostjo uspešno reši optimizacijski problem.

Uporabnost večkriterijskih iskalnih algoritmov je bila prikazana že v različnih domenah [7]. Primeri večkriterijskih optimizacijskih problemov iz dejanskega sveta vključujejo optimizacijo stacionarne plinske gorilne turbine [8], oblikovanje drobilca kamnov [9], problem distribucije naftnih derivatov [10], upravljanje z jedrskim gorivom [11], planiranje [12], načrtovanje telekomunikacijskih omrežij [13], obrambne aplikacije [14] in načrtovanje elektromotorjev [15].

Za ocenitev učinkovitosti iskalnih algoritmov so bile predlagane številne metrike [16, 17], zato je večina obstoječih študij in primerjav algoritmov zasnovanih

na specifičnih testnih meritvah. Te študije so največkrat statistične, izvedene nad izbranimi testnimi problemi, ki jih algoritem skuša rešiti.

2.1.1 Klasične optimizacijske metode

Klasične optimizacijske metode uporabljajo prednostni pristop in so prve metode, s katerimi so v preteklosti reševali večkriterijske optimizacijske probleme. Oglejmo si dve najbolj priljubljeni klasični metodi: metodo utežene vsote in metodo ϵ -omejitev. Več o klasičnih metodah najdemo v [7].

Metoda utežene vsote

Metoda utežene vsote (*weighted sum method*) je najbolj razširjena klasična metoda za večkriterijsko optimiziranje. Večkriterijski problem pretvorimo v enokriterijskega tako, da izberemo uteži w_i , ki določajo pomembnost kriterijev. Na ta način iz naloge

$$\text{optimiziraj } \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_M(\mathbf{x}))$$

dobimo naloge

$$\text{optimiziraj } \mathbf{g}(\mathbf{x}) = (w_1 f_1(\mathbf{x}) + \dots + w_M f_M(\mathbf{x})),$$

ki jo lahko rešujemo z ustrezno metodo za enokriterijsko optimiziranje. Ker se optimum te naloge ne spremeni, če uteži pomnožimo s konstanto, navadno predpostavljamo, da za uteži velja

$$w_i \in [0, 1] \text{ in } \sum_{i=1}^M w_i = 1.$$

To je enostavna metoda, ki pa ima več slabosti. Prva očitna težava metode je, da zahteva vektor uteži. Če ne poznamo prednosti kriterijev, je določitev takšnega vektorja zahtevna naloga. Poleg tega moramo, da bi dobili več Pareto optimalnih rešitev, metodo uporabiti večkrat – vsakič z drugačno nastavitvijo uteži. Vendar je pri tem treba paziti, saj enakomerno porazdeljeni vektorji uteži ne dajo nujno enakomerno porazdeljenih točk na Pareto optimalni fronti.

Ker preslikava med vektorjem uteži in točkami na Pareto optimalni fronti navadno ni poznana, težko najdemo nastavitev uteži, ki bo dala želeno točko Pareto optimalne fronte. Poleg tega različni vektorji uteži ne dajo nujno različnih točk na Pareto optimalni fronti. Podobno tudi en vektor uteži lahko določa različne točke Pareto optimalne fronte. Če je večkriterijski optimizacijski problem konveksen, po-

tem lahko vsako točko Pareto optimalne fronte izračunamo z metodo utežene vsote. Vendar je metoda omejena na konveksne Pareto optimalne fronte, za nekonveksne fronte pa odpove. Metoda je tudi občutljiva na razmerja med vrednostmi kriterijev, zato je dobro kriterije predhodno normirati. Največja prednost te metode je njena enostavnost. Če rešujemo konveksen večkriterijski optimizacijski problem, pri katerem so prednosti kriterijev poznane, je metoda uteženih vsot prava izbira.

Metoda ϵ -omejitev

Če se želimo izogniti težavam, ki so značilne za metodo utežene vsote pri nekonveksnih Pareto optimalnih frontah, lahko uporabimo metodo ϵ -omejitev (*ϵ -constraint method*). Haimes in sodelavci [18] so predlagali reševanje večkriterijskega optimizacijskega problema tako, da optimiziramo samo en kriterij, ostale pa dodamo med omejitve. Preoblikovana optimizacijska naloga se tako glasi:

$$\text{optimiziraj } f_j(\mathbf{x}) \text{ pri pogojih } f_i(\mathbf{x}) \leq \epsilon_i, \quad i = 1, \dots, M \text{ in } i \neq j$$

V tej formulaciji parameter ϵ_i označuje zgornjo mejo za vrednost f_i in ne pomeni nujno majhne vrednosti blizu nič.

Kot zanimivost lahko povemo, da se metoda ϵ -omejitev uporablja tudi na področju enokriterijske optimizacije [19], za podoben namen, kako pretvoriti enokriterijski problem s podanimi omejitvami v enokriterijskega brez omejitev.

Podobno kot metoda utežene vsote tudi metoda ϵ -omejitev zahteva informacijo o kriterijih. Tokrat je ta podana v obliki vektorja omejitev. Če izberemo drugačen vektor omejitev ali če za optimiziranje izberemo drug kriterij, dobimo drugačne rešitve. Ker omejujemo kriterije (in ne spremenljivk), nam ta metoda omogoča lažje lokaliziranje rešitev v želenih območjih. Poleg tega nima težav z obliko Pareto optimalne fronte, saj deluje na enak način za konveksne in nekonveksne ter zvezne in diskretne fronte. Metoda tudi ne zahteva normiranja kriterijev, saj razlike v vrednosti kriterijev upoštevamo v vektorju omejitev.

2.2 Evolucijski algoritmi

Evolucijski algoritmi (EA) so se pokazali kot zelo uporabni pri reševanju kompleksnih večkriterijskih optimizacijskih problemov, vključno s številnimi problemi iz dejanskega sveta. Evolucijski algoritmi temeljijo na konceptu neodarvinizma [20], ki je nastal po združitvi Darwinovega koncepta naravne selekcije [21], Weismannovega koncepta dedne plazme [22] in Mendlovih zakonov dedovanja [23]. Neodarvinizem je biološka podlaga evolucijskim algoritmom, tako da nanje gledamo kot posnema-



nje narave. Koncept neodarvinizma vse življenje na Zemlji modelira oz. razloži z naslednjimi koncepti [24]:

- **reprodukcia** (ang. *reproduction*): način, kako iz obstoječega nabora primerkov narediti nove,
- **mutacija** (ang. *mutation*): manjša sprememba kodirane dedne zasnove, ki povzroči večje spremembe v obnašanju primerka,
- **tekmovanje** (ang. *competition*): mehanizem za oceno ustreznosti primerka v danem okolju in
- **selekcija** (ang. *selection*): vzdržuje ali povečuje ustreznost populacije, ustreznost definira kot sposobnost preživetja in reprodukcije v danem okolju.

Na začetku raziskav na področju večkriterijske optimizacije so bile v uporabi klasične enokriterijske optimizacijske metode (npr. Newtonova metoda), pri tem pa se je večkriterijski problem z utežno funkcijo pretvoril v enokriterijski problem. Prvi poskus reševanja večkriterijskega problema z iskanjem vseh možnih rešitev je bil predstavljen v [25]. Od takrat naprej [26, 27], še najbolj pa v zadnjem desetletju, se je nato pri večkriterijski optimizaciji precej povečal interes za uporabo populacijskih naključnih iskalnih algoritmov, od [7, 28, 29, 30, 31, 32, 33, 34] vse do [35], kot so evolucijski algoritmi, simulirano ohlajanje in iskanje s tabuji. S temi želimo najti čim več in čim ugodnejše porazdeljenih Pareto optimalnih rešitev.

Poznamo več vrst evolucijskih algoritmov:

- **genetski algoritem** (ang. *Genetic Algorithm – GA*) [36] rešitve kreira tako, da realno rešitev binarno kodira, nato pa z operatorji **mutacije** (*mutation*) in **križanja** (*crossover*) spreminja posamezne bite v binarni predstavitev. Operator mutacije navadno deluje uniformno naključno nad vsakim bitom binarno kodiranega iskalnega parametra v posamezni rešitvi. Operator križanja pa združi posamezne iskalne parametre iz več posameznikov (staršev) v novega posameznika, tako da jemlje nekaj bitov iz vsakega od staršev,
- **genetsko programiranje** (ang. *Genetic Programming* [37]) gradi drevesno strukturo, program; uporabljeni genetski operatorji so posebej prirejeni za operacije nad drevesom,
- **evolucijsko programiranje** (ang. *Evolutionary Programming – EP*) je bilo predstavljeno kot pristop v umetni inteligenci [38], enega od sodobnejših algoritmov pa predstavlja delo [39], ki daje tudi primerjalno osnovo za številne medsebojne primerjave ostalih algoritmov nad standardnim izbranim naborom testnih funkcij za enokriterijsko optimizacijo,

- **optimizacija z rojem delcev** (ang. *Particle Swarm Optimization* – PSO) poleg vsakega vektorja iskalnih parametrov (delca) hrani še njegov diferenčni vektor, ki modelira hitrost premikanja delca v geografskem (iskalnem) prostoru [40, 41],
- **optimizacija s kolonijo mravelj** (ang. *Ant Colony Optimization* – ACO) modelira pomnenje zgodovine dobrih iskalnih parametrov po vzoru mravelj, ki v primeru najdene hrane za seboj puščajo feromonske sledi [42],
- **evolucijske strategije** (ang. *Evolution Strategies* – ES), ki poleg iskalnih parametrov v posamezni vektor kodirajo tudi krmilne parametre iskalnega algoritma (za pregled glej [43, 44, 45], močnejši predstavniki skupine so [46, 47]) in
- **diferencialna evolucija** (ang. *Differential Evolution* – DE), ki je opisana v naslednjem razdelku.

2.2.1 Diferencialna evolucija

Diferencialna evolucija (DE) [1] je novejši algoritem, ki se uspešno uporablja za globalno optimizacijo realno kodiranih numeričnih funkcij. Algoritem je preprost in ima malo parametrov (F – skalirni faktor diferenčnega vektorja za mutacijo, CR – stopnja križanja, NP – velikost populacije ter včasih tudi s – izbrana strategija mutacije in križanja), vendar zaradi svoje narave prilagajanja problemu (pri tvorbi potomcev se upoštevajo difference med starši) in stabilnosti iskanja (ang. *steady-state search*) zaradi uporabljenega elitističnega selekcijskega mehanizma, daje boljše rezultate [48, 49, 50, 51] od ostalih evolucijskih algoritmov [52, 53].

Natančneje razložimo osnovni algoritem diferencialne evolucije, ki deluje za enokriterijsko optimizacijo [1]. Ta sestoji iz glavne evolucijske zanke, v kateri algoritem DE z evolucijskimi operatorji mutacije, križanja in selekcije postopno in vzporedno izboljšuje približek iskane rešitve. Algoritem DE ima globalno in lokalno povezano velikost diferenčnega koraka, ki se samoprilagaja skozi čas glede na položaj posameznikov populacije v iskalnem prostoru [54]. Evolucijski operatorji vplivajo na vsak primerek \mathbf{x}_i , $\forall i \in [0, NP]$ v populaciji rešitev, iz katerih se zgradi nova populacija za naslednjo **generacijo** (ang. *generation*). Eno kreiranje novega osebka poimenujemo **iteracija** (ang. *iteration*). V vsaki iteraciji operator mutacije izračuna **mutiran vektor** (ang. *donor vector*) $\mathbf{v}_{i,G+1}$:

$$\mathbf{v}_{i,G+1} = \mathbf{x}_{r_1,G} + F \times (\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G}),$$

kjer so $r_1, r_2, r_3 \in 1, 2, \dots, NP$ paroma in od i različni indeksi primerkov iz populacije v generaciji G , $i \in 1, 2, \dots, NP$ in $F \in [0, 2]$. Vektor r_1 imenujemo **osnovni vektor** (ang. *base vector*). Izraz $\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G}$ imenujemo **diferenčni vektor** (ang. *difference vector*) in po množenju s faktorjem ojačanja F , **utežen diferenčni vektor** (ang. *weighted difference vector*).

Pravkar opisano strategijo mutacije imenujemo 'rand/1', obstajajo še druge [48], kot je strategija 'best/1':

$$\mathbf{v}_{i,G+1} = \mathbf{x}_{best,G} + F \times (\mathbf{x}_{r_1,G} - \mathbf{x}_{r_2,G}),$$

kjer $\mathbf{x}_{best,G}$ označuje najbolje ocenjen primerek iz generacije G . V mutacijo lahko vključimo tudi starševski primerek $\mathbf{x}_{i,G}$, kot v strategiji 'rand-to-best/1':

$$\mathbf{v}_{i,G+1} = \mathbf{x}_{i,G} + F \times (\mathbf{x}_{best,G} - \mathbf{x}_{i,G}) + F \times (\mathbf{x}_{r_1,G} - \mathbf{x}_{r_2,G}).$$

V kolikor v mutacijo vključimo več naključnih posameznikov, poznamo še strategijo 'rand/2':

$$\mathbf{v}_{i,G+1} = \mathbf{x}_{r_1,G} + F \times (\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G}) + F \times (\mathbf{x}_{r_4,G} - \mathbf{x}_{r_5,G})$$

in strategijo mutacije z najboljšim vektorjem, 'best/2':

$$\mathbf{v}_{i,G+1} = \mathbf{x}_{best,G} + F \times (\mathbf{x}_{r_1,G} - \mathbf{x}_{r_2,G}) + F \times (\mathbf{x}_{r_3,G} - \mathbf{x}_{r_4,G}).$$

Po mutaciji dobljeni mutiran vektor $\mathbf{v}_{i,G+1}$ križamo s **ciljnimi vektorjem** (ang. *target vector*) $\mathbf{x}_{i,G}$ in tako dobimo **poskusni vektor** (ang. *trial vector*) $\mathbf{u}_{i,G+1}$. Operator križanja v algoritmu DE prevzema dve obliki, imenovani binarno križanje ('/bin') ali eksponentno križanje ('/exp'). Prvo zapišemo kot:

$$u_{i,j,G+1} = \begin{cases} v_{i,j,G+1} & \text{if } \text{rand}(0, 1) \leq CR \text{ ali } j = j_{rand} \\ x_{i,j,G} & \text{sicer} \end{cases},$$

kjer $j \in [1, D]$ označuje j -ti iskalni parameter v prostoru z D dimenzijami, funkcija $\text{rand}(0, 1) \in [0, 1]$ označuje vzorčenje uniformno (psevdo) naključno porazdeljenega števila in j_{rand} izbira uniformno naključen indeks iskalnega parametra, ki ga vedno izmenjamo (da bi s tem preprečili izdelavo enakih posameznikov). CR označuje že omenjen krmilni parameter stopnje križanja.

Eksponentno križanje lahko izrazimo kot prepis le določenih zaporednih iskalnih

parametrov

$$u_{i,j,G+1} = \begin{cases} v_{i,j,G+1} & \text{if } j = \langle n \rangle_D, \langle n + 1 \rangle_D, \dots, \langle n + L - 1 \rangle_D, \\ x_{i,j,G} & \text{sicer} \end{cases},$$

kjer $\langle \cdot \rangle_D$ označuje operacijo ostanka pri deljenju z D , n označuje indeks začetka iz $v_{i,j,G+1}$ prepisovanega zaporedja in $L \in [0, D - 1]$ njeno eksponentno porazdeljeno dolžino zaporedja $P(L = v) = (CR)^v$.

Operator selekcije v algoritmu DE je dokaj preprost, saj za vsak novo generiran primerek zgolj preveri, ali je ocena $f(\mathbf{u}_{i,G+1})$ poskusnega vektorja boljša (boljše pomeni manjšo vrednost) od ocene ciljnega vektorja $f(\mathbf{x}_{i,G})$:

$$\mathbf{x}_{i,G+1} = \begin{cases} \mathbf{u}_{i,G+1} & \text{if } f(\mathbf{u}_{i,G+1}) < f(\mathbf{x}_{i,G}), \\ \mathbf{x}_{i,G} & \text{sicer} \end{cases},$$

kjer algoritem DE privzema elitističen princip preživetja najboljših [21].

S kombinacijo naštetih operatorjev mutacije in križanja se je uveljavilo oštevilčenje imen teh kombinacij v algoritmu DE, ki jih vidimo v tabeli 2.1. V kolikor

Tabela 2.1: Diferencialna evolucija – kombinacije operatorjev križanja in mutacije ter njihovo poimenovanje v [1].

1	'DE/best/1/exp'
2	'DE/rand/1/exp'
3	'DE/rand-to-best/1/exp'
4	'DE/best/2/exp'
5	'DE/rand/2/exp'
6	'DE/best/1/bin'
7	'DE/rand/1/bin'
8	'DE/rand-to-best/1/bin'
9	'DE/best/2/bin'
10	'DE/rand/2/bin'

imamo iskalni problem, kjer potrebujemo hitro konvergenco, lahko izberemo strategijo 1 ali strategije 4, 6 in 9. Najpogosteje uporabljena je strategija 'DE/rand/1/bin'. Strategije 2, 5 in 10 lahko uporabimo, če ostale odpovejo, obstaja pa tudi veliko število izboljšav in dopolnitev osnovnega algoritma diferencialne evolucije, glej [55].

Obstaja tudi nekaj izboljšav osnovnih mutacijskih strategij [56, 57, 58, 59, 60]. Izboljšave diferencialne evolucije zajemajo tudi robustnost pri velikih dimenzijah iskalnega prostora [61, 62]. Križanje algoritma DE in evolucijskega programiranja [39] zasledimo v delu [63]. Bošković in sodelavci so diferencialno evolucijo uporabljali za ugleševanje parametrov iger s popolno informacijo in ničelno vsoto [64].

Algoritem DE je prav tako rotacijsko invarianten v prostoru spremenljivk [65]. Posledica tega je, da lahko dobro rešuje probleme z neločljivimi iskalnimi parametri. Takšno lastnost izkazuje precej realnih oz. industrijskih problemov [65, 66]. Zaradi svoje uspešnosti je algoritem DE bil že večkrat uporabljen tudi za večkriterijsko optimizacijo. V teh primerih uporabe so najbolj znana dela [30, 67, 68, 32, 33].

2.2.2 Samoprilagajanje krmilnih parametrov

Samoprilagajanje krmilnih parametrov je bilo predstavljeno z evolucijskimi strategijami [43, 44, 45], kjer so krmilni parametri vključeni v osebke skupaj z iskalnimi parametri. Notacija splošne evolucijske strategije (ES) je $\mu/\rho, \lambda$ -ES, kjer je μ velikost starševske populacije, ρ število staršev za vsakega novega posameznika in λ velikost nove populacije. Posameznik je definiran kot $\mathbf{a} = (\mathbf{x}, \mathbf{s}, F(\mathbf{x}))$, kjer so \mathbf{x} iskalni parametri, \mathbf{s} krmilni parametri in $F(\mathbf{x})$ ocenitev posameznika.

Čeprav obstajajo druge notacije in variante evolucijskih strategij [46, 69, 47], omenimo zaenkrat zgolj osnovno varianto. Ta uporablja logaritmično normalno porazdeljeno [70] prilagajanje krmilnih parametrov, ki se skozi selekcijo samoprilagajajo. Preživijo le osebki, ki se prilagajajo dobrim krmilnim parametrom in dobrim iskalnim parametrom. Z bolj primernimi vrednostmi krmilnih parametrov proces iskanja namreč doseže boljše rešitve in zaradi tega se iskanje hitreje približuje boljšim rešitvam, ki preživijo z večjo verjetnostjo in ustvarijo več potomcev ter s tem razširijo svoje krmilne parametre [6, 71].

Tudi za algoritem diferencialne evolucije (DE) [1, 58], je že bila potrjena potreba po samoprilagajanju krmilnih parametrov [72, 73, 74] in uporabljena v [75, 76, 48, 49] ter [50, 77, 78]. Pristop izdelave kandidatnih rešitev z diferencialno evolucijo so izbrali že algoritmi za večkriterijsko optimizacijo, izmed katerih so najbolj znani algoritmi [30, 67, 32].

2.3 Evolucijski algoritmi za večkriterijsko optimizacijo

Vsi evolucijski algoritmi so populacijsko naravnani, saj ta pristop pride prav pri večkriterijski optimizaciji, kjer želimo odkriti množico rešitev, ki ponujajo številne kompromise med kriteriji. Omenimo sorodne pristope k selekciji (*environmental selection*) oz. sekanju (*prunning*) za umestitev našega algoritma:

NSGA-II (*Elitist Non-Dominated Sorting Genetic Algorithm*) je algoritem Deba in sodelavcev iz leta 2000 [79] in je zelo uspešen genetski algoritem (algori-



tem je nadgradnja NSGA [80]), ki se je v svojem času odrezal najbolje od vseh (takrat obstoječih) evolucijskih algoritmov za večkriterijsko optimizacijo. Algoritom uporablja **nedominirano urejanje** in **metriko nakopičenosti**, da tako izloča manj obetavne rešitve. Pri prvem gre za urejanje rešitev po frontah, pri čemer po vrsti za vsako velja, da nobena rešitev iz boljše fronte ni dominirana od rešitve iz naslednje slabše fronte. Metrika nakopičenosti pomaga razrešiti neodločene izide po prvem pristopu, tj. čim bolj oddaljen od rešitev v isti fronti.

POGA (*Preference Ordering Genetic Algorithm*) [35] je novejši primer nadgradnje večkriterijske selekcije algoritma NSGA-II, ki ima spremenjeno urejanje po frontah tako, da vpelje delno dominantnost, to je dominantnost na podmnožici kriterijev. V primeru, ko sta dve rešitvi neprimerljivi oz. nedominantni, se ohrani rešitev, ki po čim več kriterijih dominira druge.

SPEA-II (*Strength Pareto Evolutionary Algorithm*) je drugačen algoritem Zitzlerja in sodelavcev iz leta 2001 [81, 29] in je prav tako popularen genetski algoritmom za večkriterijsko optimizacijo, ki glede na NSGA-II najde enakomernejše porazdeljene (manj nakopičene) rešitve. Vrednotenje osebkov ne poteka po frontah, pač pa enovito z merjenjem **moči** (*strength*), **grobe uspešnosti** (*raw fitness*) in **gostote** (*density*) osebkov. Moč osebka je enaka številu osebkov iz populacije, ki jih izbrani primerek dominira. Kot zanimivost povejmo, da Zitzler in sod. označujejo relacijo dominiranosti v obratni smeri, s \succ . Grobo uspešnost osebka izračunamo kot vsoto moči vseh osebkov iz populacije, ki ta primerek dominirajo. Ker z evolucijo številni osebki postanejo nedominirani (in imajo zato grobo uspešnost enako nič), potrebujemo dodatno informacijo o nakopičenosti rešitev. Zato za vse osebke z enako grobo uspešnostjo izračunamo še gostoto. Algoritom za njen izračun je prilagojena različica algoritma k -tega najbližjega soseda [82]. Metriko nakopičenosti osebka dobimo tako, da seštejemo grobo uspešnost in gostoto.

Nekateri novejši evolucijski algoritmi za večkriterijsko optimizacijo rešujejo tudi probleme šuma v vhodnih podatkih [83]. Laumanns in sodelavci so enokriterijsko optimizacijo z omejitvami reševali z večkriterijsko optimizacijo [84]. Tan in sodelavci [85] so v večkriterijskem optimizacijskem algoritmu uporabili koevolucijo [86] in optimizacijo izvajali paralelno. Primer uporabe diferencialne evolucije za večkriterijsko optimizacijo realnih problemov vključuje tudi podatkovno rudarjenje [87].

Poglavlje 3

Diferencialna evolucija za večkriterijsko optimizacijo s samoprilagajanjem

Preden podamo celoten načrt algoritma diferencialne evolucije za večkriterijsko optimizacijo s samoprilagajanjem (DEMOwSA) [88] v obliki psevdokoda, osvetlimo njegove bistvene elemente. Podali bomo krajši pregled ožje sorodnih tehnik in predstavili postopek preiskovanja iskalnega prostora in samoprilagajanja krmilnih parametrov tega preiskovanja.

Pristop izdelave kandidatnih rešitev z diferencialno evolucijo so izbrali že obstoječi algoritmi za večkriterijsko optimizacijo, izmed katerih so najbolj znani algoritmi [30, 67, 32, 33].

Izvorni kod in idejna zasnova algoritma sloni na izvornem kodu osnovnega algoritma DEMO [33, 89]. DEMO Tušarjeve in Filipiča je algoritem diferencialne evolucije [1, 90] za večkriterijsko optimizacijo, ki uporablja fiksne krmilne parametre diferencialne evolucije ($F = 0,5$, $CR = 0,3$). Omogoča izbiro več selekcijskih mehanizmov NSGA-II [28], SPEA-II [29] ali IBEA [31], ki jih izberemo pred optimacijskim zagonom. Izboljšane rešitve shranjuje v arhiv aproksimacijske množice rešitev, ki predstavlja ob enem tudi trenutno populacijo rešitev.

Algoritmu DEMO [33, 91, 89] smo dodali zmožnost samoprilagajanja krmilnih parametrov za izdelavo kandidatnih rešitev po vzoru evolucijskih strategij, z logaritmično normalno porazdeljenim prilagajanjem. Izbrali smo SPEA-II večkriterijsko selekcijo in ohranili ostale nastavitve parametrov v algoritmu DEMO. Tako smo pripravili nov algoritem za večkriterijsko optimizacijo, ki ga poimenujemo DEMOwSA.

3.1 Predstavitev samoprilagodljivih krmilnih parametrov

Vsek posameznik (glej sliko 3.1) algoritma DEMOwSA je razširjen s samoprilagodljivimi krmilnimi parametri F in CR . Vrednosti teh parametrov se samoprilagajajo skozi proces iskanja, saj so krmilni parametri predstavljeni v vsakem posamezniku poleg ostalih parametrov.

$x_{I,1,G}$	$x_{I,2,G}$	$x_{I,3,G}$...	$x_{I,D,G}$	$F_{I,G}$	$CR_{I,G}$	$f_1(\mathbf{x}_{I,G})$	$f_2(\mathbf{x}_{I,G})$	$f_3(\mathbf{x}_{I,G})$...	$f_M(\mathbf{x}_{I,G})$
$x_{2,1,G}$	$x_{2,2,G}$	$x_{2,3,G}$...	$x_{2,D,G}$	$F_{2,G}$	$CR_{2,G}$	$f_1(\mathbf{x}_{2,G})$	$f_2(\mathbf{x}_{2,G})$	$f_3(\mathbf{x}_{2,G})$...	$f_M(\mathbf{x}_{2,G})$
$x_{3,1,G}$	$x_{3,2,G}$	$x_{3,3,G}$...	$x_{3,D,G}$	$F_{3,G}$	$CR_{3,G}$	$f_1(\mathbf{x}_{3,G})$	$f_2(\mathbf{x}_{3,G})$	$f_3(\mathbf{x}_{3,G})$...	$f_M(\mathbf{x}_{3,G})$
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\ddots	\vdots
$x_{NP,1,G}$	$x_{NP,2,G}$	$x_{NP,2,G}$...	$x_{NP,D,G}$	$F_{NP,G}$	$CR_{NP,G}$	$f_1(\mathbf{x}_{NP,G})$	$f_2(\mathbf{x}_{NP,G})$	$f_3(\mathbf{x}_{NP,G})$...	$f_M(\mathbf{x}_{NP,G})$

Slika 3.1: Predstavitev samoprilagodljivih krmilnih parametrov in ovrednotenja v posamezniku.

3.2 Spreminjanje samoprilagodljivih krmilnih parametrov

V procesu izdelave primerkov izdelamo nov primerek iz starševskih primerkov s pomočjo operatorjev mutacije in križanja. Glavna razlika v algoritmu DEMOwSA od algoritma DEMO je ravno v uporabi operatorja mutacije. Tu se uporablja prilagodljivi faktor ojačanja diferenčnega vektorja F_i za vsakega od posameznikov i iz generacije G , da bi ustvarili nov primerek za generacijo $G + 1$:

$$F_{i,G+1} = \langle F_G \rangle_i \times e^{\tau N(0,1)},$$

kjer τ označuje stopnjo učenja in je ponavadi sorazmeren s $\tau \sim 1/(\sqrt{2D})$ in D označuje dimenzijo problema. $N(0, 1)$ je po Gaussu porazdeljena naključna spremenljivka. Izraz $\langle F_G \rangle_i$ označuje povprečenje parametra F i -tega obravnavanega posameznika in r_1 , r_2 ter r_3 označujejo uniformno naključno izbrane posameznike iz generacije G :

$$\langle F_G \rangle_i = \frac{F_{i,G} + F_{r_1,G} + F_{r_2,G} + F_{r_3,G}}{4},$$

kjer so indeksi r_1 , r_2 in r_3 paroma ter od i različni indeksi posameznikov, ki so uporabljeni za izračun diference v mutacijskem procesu algoritma DE.

Mutacijski proces nad i -tim posameznikom $\mathbf{v}_{i,G+1}$ za generacijo $G + 1$ deluje na naslednji način:

$$\mathbf{v}_{i,G+1} = \mathbf{x}_{r_1,G} + F_{i,G+1} \times (\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G}),$$

kjer so $\mathbf{x}_{r_1,G}$, $\mathbf{x}_{r_2,G}$ in $\mathbf{x}_{r_3,G}$ vrednosti iskalnih parametrov uniformno naključno izbranih staršev iz generacije G .

Za krmilni parameter CR uporabimo podobno formulo:

$$CR_{i,G+1} = \langle CR_G \rangle_i \times e^{\tau N(0,1)},$$

kjer je τ podoben kot v prilagajanju parametra F in izraz $\langle CR_G \rangle_i$ ponovno označuje povprečenje:

$$\langle CR_G \rangle_i = \frac{CR_{i,G} + CR_{r_1,G} + CR_{r_2,G} + CR_{r_3,G}}{4}.$$

Proces križanja v algoritmu DEMOwSA je vzet iz strategije 'rand/1/bin', ki jo uporablja originalni algoritem DEMO [33] oz. originalni algoritem DE [1]. Krmilni parameter tega procesa, CR , smo prilagajali in dobili $CR_{i,G+1}$. Slednji je uporabljen pri izdelavi kandidata $u_{i,j,G+1}$ z binarnim operatorjem križanja:

$$u_{i,j,G+1} = \begin{cases} v_{i,j,G+1} & \text{if } \text{rand}(0, 1) \leq CR_{i,G+1} \text{ ali } j = j_{rand} \\ x_{i,j,G} & \text{sicer,} \end{cases}$$

kjer $j \in [1, D]$ označuje j -ti iskalni parameter, $\text{rand}(0, 1) \in [0, 1]$ pomeni vzorčenje uniformno (psevdo) naključno porazdeljenega naključnega števila in j_{rand} pomeni uniformno naključen indeks iskalnega parametra, ki ga vedno izmenjamo (da bi s tem preprečili izdelavo enakih posameznikov).

Operator selekcije pomaga tudi pri nadzoru in ustreznom prilagajanju parametrov F in CR , saj samoprilagaja njune vrednosti.

3.3 Zapis v psevdokodu

Da bi razumeli celoten algoritmom DEMOwSA, nam bo lažje, če ga osvojimo po delih. Zato bomo najprej predstavili nekaj enostavnijih netrivialnih podalgoritmov, nato pa podali celovito sliko algoritma v psevdokodu, ki združuje zapisane manjše kose celega algoritma.

Postopek za ovrednotenje posameznih primerkov je lahko dokaj preprost. Ponavadi izvede numerično funkcijo, takšen primer podaja tudi algoritem Ovrednoti_{OKA2} (glej algoritmom 1).

En od preprostih manjših postopkov je tudi primerjava rešitev, ki ga ponazarja

Algoritem 1 Ovrednoti_{OKA2}: primer testne funkcije OKA2.**Vhod:** \mathbf{p} – primerek rešitve večkriterijske funkcije, $x_1 \in [-\pi, \pi]$, $x_2, x_3 \in [-5, 5]$.**Izhod:** Izračunana $\mathbf{f}(\mathbf{x})$, ki se shrani v \mathbf{p} (za ponazoritev predstavitev primerka glej sliko 3.1).

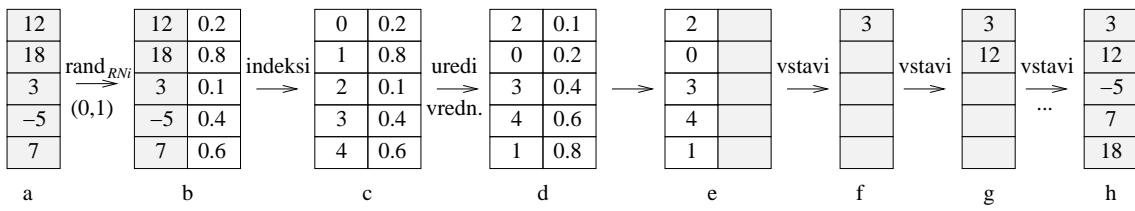
- 1: $x_1, x_2, x_3 \leftarrow \mathbf{p}$ {Iz \mathbf{p} izlušči iskalne parametre.}
 - 2: $f_1(\mathbf{x}) = x_1$
 - 3: $f_2(\mathbf{x}) = 1 - \frac{1}{4\pi^2} (x_1 + \pi)^2 + |x_2 - 5 \cos(x_1)|^{\frac{1}{3}} + |x_3 - 5 \sin(x_1)|^{\frac{1}{3}}$
 - 4: $\mathbf{f}(\mathbf{x}) = (f_1, f_2)$
 - 5: $\mathbf{p} \leftarrow \mathbf{f}(\mathbf{x})$ {V \mathbf{p} shrani vrednosti kriterijev.}
-

algoritom Primerjaj rešitvi (glej algoritom 2). Ta medsebojno primerja dve rešitvi in vrne, katera je boljša ali pa razsodi, da sta rešitvi neprimerljivi, tj. nobena ni boljša. Pri primerjavi se upošteva pojem dominiranja, definiran v uvodu.

Algoritem 2 Primerjaj rešitvi.**Vhod:** \mathbf{p}_i – primerek starševske rešitve iz trenutne populacije.**Vhod:** \mathbf{u} – nov, kandidatni primerek rešitve.**Vhod:** M – dimenzija prostora kriterijev izbrane funkcije.**Izhod:** ND – nedominiranost (1 = nov je dominanten, -1 = starš je dominanten, 0 = neprimerljiva).

- 1: $d_{new} = 0$ {Zastavica beleži, ali je nov kandidatni primerek že dominiran po enem od kriterijev.}
 - 2: $d_{parent} = 0$ {Zastavica beleži, ali je starševski primerek že dominiran po enem od kriterijev.}
 - 3: **for** $j = 0$ to M **do**
 - 4: **if** $\mathbf{u}_j < \mathbf{p}_{i,j}$ **then**
 - 5: $d_{parent} = 1$
 - 6: **else if** $\mathbf{p}_{i,j} < \mathbf{u}_j$ **then**
 - 7: $d_{new} = 1$
 - 8: **end if**
 - 9: **if** $d_{parent} = 1$ **and** $d_{new} = 1$ **then**
 - 10: $ND := 0$ {Rešitvi sta neprimerljivi, nobena ne dominira druge.}
 - 11: **return** {Takoj prenehamo s primerjavo — izboljšava iz DEMO na DEMOwSA.}
 - 12: **end if**
 - 13: **end for**
 - 14: $ND := d_{new} - d_{parent}$ {1, če je dominanten nov, -1 , če je dominanten starševski primerek.}
-

Komentirajmo algoritmom Premešaj populacijo (glej algoritom 3, ter sliko 3.2 za primer takšnega naključnega premešanja). Ta deluje tako, da seznam $\mathbf{a} = (12, 18, 3, -5, 7)$ naključno premeša (a). Zato za vsak element seznama zgenerira enakomerno porazdeljeno naključno število med 0 in 1 (b) in si ob tem zabeleži še pripadajočo številko indeksa, h kateremu elementu število spada, (c). Seznam uredi po generiranih naključnih vrednostih (d), nato začeten seznam \mathbf{a} sprazni (e) in ga zaporedoma znova napolni (f–h), tako da po vrsti vstavlja elemente iz dobljenih indeksov v koraku (e).



Slika 3.2: Prikaz algoritma za premešanje seznama na preprostem primeru.

Algoritem 3 Premešaj populacijo.

Vhod: \mathbf{P}_g – vhodna populacija primerkov v generaciji g , pred pričetkom izdelave novih primerkov.

Vhod: NP – velikost populacije in aproksimacijske množice \bar{N} .

Vhod: RNi – trenutno stanje generatorja enakomerno porazdeljenih naključnih števil.

Izhod: \mathbf{P}'_g – delovna populacija primerkov za generacijo g , med tvorjenjem novih primerkov.

Izhod: Spremenjen RNi .

- 1: $\mathbf{r} := \emptyset$ {Seznam naključnih števil.}
 - 2: $\mathbf{r}' := \emptyset$ {Seznam indeksov, glej sliko 3.2.}
 - 3: **for** $i = 0$ to NP **do**
 - 4: $\mathbf{r} := \mathbf{r} \cup \text{rand}_{RNi}(0, 1)$ {Dodaj enakomerno porazdeljena naključna števila – slika 3.2b.}
 - 5: $\mathbf{r}' := \mathbf{r} \cup i$ {Pred vsak element seznama \mathbf{r} shrani indeks i – slika 3.2c.}
 - 6: **end for**
 - 7: Seznam \mathbf{r}, \mathbf{r}' uredi po \mathbf{r} . {Hitro urejanje: $\Theta(NP \log NP)$ – slika 3.2d.}
 - 8: $\mathbf{P}'_g := \emptyset$ {Nov seznam primerkov v populaciji.}
 - 9: **for** $i = 1$ to NP **do**
 - 10: $\mathbf{P}'_g := \mathbf{P}'_g \cup \mathbf{p}_{g,r_i}$ {Primerke na indeksih r_i iz populacije \mathbf{P}_g dodaj v \mathbf{P}'_g – slika 3.2e–h.}
 - 11: **end for**
-

Primerke rešitev optimizacijske funkcije algoritom DEMOwSA inicializira, kot je prikazano v algoritmu Inicializiraj nov primerek (glej algoritom 4).

Ustvarjanje novih kandidatnih rešitev in s tem preiskovanje iskalnega prostora poteka v algoritmu Nov primerek (glej algoritom 5). Ta iz starša \mathbf{p}_i in diferenc med

Algoritem 4 Inicializiraj nov primerek.

Vhod: $func_type$ ($\mathbf{g}(\mathbf{x}) \geq 0$) – zaporedna številka funkcije, ki jo optimiziramo (njene omejitve).

Vhod: D – dimenzija prostora spremenljivk.

Vhod: M – dimenzija prostora kriterijev.

Vhod: F_{init} – začetna vrednost parametrov F .

Vhod: CR_{init} – začetna vrednost parametrov CR .

Vhod: RNi – seme generatorja enakomerno porazdeljenih naključnih števil.

Vhod: RNj – seme generatorja normalno porazdeljenih naključnih števil.

Izhod: \mathbf{p} – nov primerek rešitve večkriterijske funkcije.

Izhod: Spremenjen RNi .

- 1: $\mathbf{a} = \emptyset$
 - 2: **for** $j = 0$ to D **do**
 - 3: $a_j := \text{rand}_{RNi}(x_{min,i}, x_{max,i})$ { $x_{min,i}$ in $x_{max,i}$ sta vsebovana v omejitvah $\mathbf{g}(\mathbf{x}) \geq 0$.}
 - 4: **end for**
 - 5: $\mathbf{s} = (F_{init}, CR_{init})$
 - 6: $\mathbf{G} = \mathbf{0}_{M+1}$ { M kriterijev in nedominantnost.}
 - 7: $\mathbf{p} = (\mathbf{a}, \mathbf{s}, \mathbf{G})$ {Sestavi primerek iz iskalnih parametrov in krmilnih parametrov.}
-

primerki v populaciji \mathbf{P}_g iz generacije g ustvari novo kandidatno rešitev \mathbf{u} .

Ker je primerkov pred koncem vsake generacije lahko v populaciji preveč, je nekatere potrebno izločiti, oz. populacijo t.i. porezati. Takšen postopek prikazuje algoritem Poreži_{SPEA-II} (glej algoritem 6), ki je natančneje opisan še z Nakopičenost_{SPEA-II} (glej algoritem 8) in Odstrani prenakopičene rešitve_{SPEA-II} (glej algoritem 7).

Sedaj imamo pripravljene vse sestavne elemente algoritma DEMOwSA. Zato lahko sedaj načrt celotnega algoritma v obliki psevdokoda podamo v algoritmu 9. Kot vidimo, algoritem 9 vsebuje prej omenjene manjše algoritme. Omenimo lahko še, da je algoritem DEMOwSA izboljšan v algoritmu 2 glede na algoritem DEMO. Spremenjeni so tudi ostali, razen algoritmov 3 in 6, 8 ter 7, ki so enaki kot v algoritmu DEMO oz. so iz algoritma SPEA2 [29]. Glavna prednost algoritma DEMOwSA glede na algoritem DEMO je v spremenjenem algoritmu Nov primerek (glej algoritem 5).

Algoritem 5 Nov primerek_{DEMOwSA/rand/1/bin}.

Vhod: \mathbf{p}_i – starševski primerek rešitve.

Vhod: D – dimenzija prostora spremenljivk.

Vhod: τ – stopnja učenja za prilagajanje samoprilagodljivih parametrov F in CR .

Vhod: F_{min} (F_{max}) – najmanjša (največja) možna vrednost parametra F .

Vhod: CR_{min} (CR_{max}) – najmanjša (največja) možna vrednost parametra CR .

Vhod: $\mathbf{g}(\mathbf{x}) \geq 0$ – omejitve funkcije, ki jo optimiziramo.

Vhod: RNi – seme generatorja enakomerno porazdeljenih naključnih števil.

Vhod: RNj – seme generatorja normalno porazdeljenih naključnih števil.

Vhod: \mathbf{P}_g – populacija primerkov v generaciji g .

Izhod: Kandidatna rešitev \mathbf{u} za generacijo $G + 1$.

Izhod: Spremenjen RNi in RNj .

```

1: repeat
2:    $i_1 := \text{rand}_{RNi}(0, NP); i_2 := \text{rand}_{RNi}(0, NP); i_3 := \text{rand}_{RNi}(0, NP)$ 
3: until not ( $i = i_3$  or  $i_1 = i_3$  or  $i_2 = i_3$ ) {Izberi paroma različne indekse  $\mathbf{p}_{g,i_1}$ ,  $\mathbf{p}_{g,i_2}$ ,  $\mathbf{p}_{g,i_3} \in \mathcal{P}_g$ .}
4:  $\langle F_G \rangle_i = \frac{F_{i,G} + F_{r_1,G} + F_{r_2,G} + F_{r_3,G}}{4}$  {Globalno povprečenje skozi primerke.}
5:  $F_{i,G+1} = \langle F_G \rangle_i \times e^{\tau N(0,1)}$  {Logaritmično normalna porazdelitev samoprilagajanja.}
6: if  $F_{i,G+1} < F_{min}$  then
7:    $F_{i,G+1} := F_{min}$  {Popravimo  $F$  v meje.}
8: else if  $F_{i,G+1} > F_{max}$  then
9:    $F_{i,G+1} := F_{max}$ 
10: end if
11:  $\mathbf{v}_{i,G+1} = \mathbf{x}_{r_1,G} + F_{i,G+1} \times (\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G})$ 
12:  $\langle CR_G \rangle_i = \frac{CR_{i,G} + CR_{r_1,G} + CR_{r_2,G} + CR_{r_3,G}}{4}$  {Globalno povprečenje primerkov.}
13:  $CR_{i,G+1} = \langle CR_G \rangle_i \times e^{\tau N(0,1)}$  {Logaritmično normalna porazdelitev samoprilagajanja.}
14: if  $CR_{i,G+1} < CR_{min}$  then
15:    $CR_{i,G+1} := CR_{min}$  {Popravimo  $CR$  v meje.}
16: else if  $CR_{i,G+1} > CR_{max}$  then
17:    $CR_{i,G+1} := CR_{max}$ 
18: end if
19:  $j_{rand;g,i} := \text{rand}_{RNi}(0, D)$  {Vsaj enega parametra pri križanju ne bomo zamenjali.}
20: for  $j = 0$  to  $D$  do
21:   if  $\text{rand}_{RNi}(0, 1) \leq CR_{i,G+1}$  ali  $j = j_{rand;g,i}$  then
22:      $u_{i,j,G+1} = v_{i,j,G+1}$  {Križamo.}
23:   else
24:      $u_{i,j,G+1} = x_{i,j,G}$  {Ne križamo.}
25:   end if
26:   if  $u_{i,j,G+1} < x_{min,i}$  then
27:      $u_{i,j,G+1} := x_{min,i}$  { $x_{min,i}$  in  $x_{max,i}$  sta vsebovana v omejitvah  $\mathbf{g}(\mathbf{x}) \geq 0$ .}
28:   else if  $u_{i,j,G+1} > x_{max,i}$  then
29:      $u_{i,j,G+1} := x_{max,i}$ 
30:   end if
31: end for
```

Algoritem 6 Poreži_{SPEA-II}.

Vhod: NP – velikost populacije in aproksimacijske množice \bar{N} .

Vhod: \mathbf{P}_g – populacija primerkov v generaciji g .

Izhod: \mathbf{P}_{g+1} – na podlagi nakopičenosti do velikosti NP porezana populacija \mathbf{P}_g .

- 1: $F(\mathbf{P}_g), \sigma :=$ Nakopičenost_{SPEA-II}(\mathbf{P}_g) {Najprej izmerimo kakovost vseh rešitev: tj. nakopičenost.}
 - 2: $\mathbf{P}_{g+1} :=$ Odstrani prenakopičene rešitve_{SPEA-II}($NP, \mathbf{P}_g, F(\mathbf{P}_g), \sigma$)
-

Algoritem 7 Odstrani prenakopičene rešitve_{SPEA-II}.

Vhod: NP – velikost populacije in aproksimacijske množice \bar{N} .

Vhod: \mathbf{P}_g – populacija primerkov v generaciji g .

Vhod: $F(\mathbf{P}_g)$ in σ – matrika nakopičenosti in matrika razdalj do k -tega primerka.

Izhod: \mathbf{P}_{g+1} – na podlagi nakopičenosti do velikosti NP porezana populacija \mathbf{P}_g .

- 1: $\mathbf{P}_{g+1} := \emptyset$ {Tu se prične kopiranje primerkov v novo generacijo.}
 - 2: **for all** $\mathbf{p}_{g,i}$ **with** $F(\mathbf{p}_{g,i}) < 1$ **do**
 - 3: $\mathbf{P}_{g+1} := \mathbf{P}_{g+1} \cup \mathbf{p}_{g,i}$ {Prekopiraj prvo fronto: vse nedominirane posameznike iz \mathbf{P}_g v \mathbf{P}_{g+1} .}
 - 4: $\mathbf{P}_g := \mathbf{P}_g \setminus \mathbf{p}_{g,i}$ {V stari generaciji primerke prve fronte odstranjujemo, da bo $F(\mathbf{p}_{g,i}) \geq 1$.}
 - 5: **end for**
 - 6: **if** $|\mathbf{P}_{g+1}| < NP$ **then**
 - 7: Uredi \mathbf{P}_{g+1} naraščajoče po $F(\mathbf{p}_{g,i})$ {Hitro urejanje: $\Theta(\mathbf{P}_{g+1} \log |\mathbf{P}_{g+1}|)$.}
 - 8: $i := 0$ {Boljši (tj. s čim manjšo nakopičenostjo $F(\mathbf{p}_{g,i})$) primerki so po urejanju spredaj.}
 - 9: **repeat**
 - 10: $\mathbf{P}_{g+1} := \mathbf{P}_{g+1} \cup \mathbf{p}_{g,i}$
 - 11: $i := i + 1$ {Kopiramo v smeri od boljših primerkov k slabšim, do zapolnitve arhiva.}
 - 12: **until** $|\mathbf{P}_{g+1}| = NP$
 - 13: **else if** $|\mathbf{P}_{g+1}| > NP$ **then**
 - 14: **repeat**
 - 15: $k := 0$ {Poiščemo najmanj oddaljen primerek $\mathbf{p}_{g+1,i}$:}
 - 16: **for all** $\mathbf{p}_{g+1,i} \in \mathbf{P}_{g+1}$ **do**
 - 17: **for all** $\mathbf{p}_{g+1,j} \in \mathbf{P}_{g+1}$ **do**
 - 18: **if** $\forall 0 < k < |\mathbf{P}_{g+1}| : \sigma_i^k = \sigma_j^k \vee \exists 0 < k < |\mathbf{P}_{g+1}| :$
 $[(\forall 0 < l < k : \sigma_i^l = \sigma_j^l)] \wedge \sigma_i^k < \sigma_j^k$ **then**
 - 19: $\mathbf{P}_{g+1} := \mathbf{P}_{g+1} \setminus \mathbf{p}_{g+1,k}$ {Odstranimo najbolj nakopičen primerek, pogoj je $\Theta(M)$.}
 - 20: **break for** i {Dokler velikost presega NP , še izvajaj operator sekanj.}
 - 21: **end if**
 - 22: **end for**
 - 23: **end for**
 - 24: **until** $|\mathbf{P}_{g+1}| = NP$
 - 25: **end if**
-

Algoritem 8 Nakopičenost_{SPEA-II}.

Vhod: NP – velikost populacije in aproksimacijske množice \bar{N} .

Vhod: \mathbf{P}_g – populacija primerkov v generaciji g .

Izhod: $F(\mathbf{P}_g)$ in σ – matrika nakopičenosti in matrika razdalj do k -tega primerka.

```

1:  $\mathbf{S}_g := \mathbf{0}$  {Primerkom iz  $\mathbf{P}_g$  inicializiraj moč na 0 — moč je predstavljena z dominan-
tnostjo.}
2: for all  $\mathbf{p}_{g+1,i} \in \mathbf{P}_{g+1}$  do
3:   for all  $\mathbf{p}_{g+1,j} \in \mathbf{P}_{g+1}$  do
4:     if Primerjaj rešitvi( $\mathbf{p}_{g,i}, \mathbf{p}_{g,j}, M$ ) = 1 then
5:        $S_{g,i} := S_{g,i} + 1$  {Prešteje dominirane primerke  $S_{\mathbf{p}_{g,i}} = |\{\mathbf{p}_{g,j} | \mathbf{p}_{g,j} \in \mathbf{P}_g \wedge \mathbf{p}_{g,i} \prec \mathbf{p}_{g,j}\}|$ }
6:     end if
7:   end for
8: end for
9:  $\mathbf{R}_g := \mathbf{0}$  {Iz  $\mathbf{S}_g$  izračunamo grobo uspešnost  $R_g$ , ki je vsota moči od  $\mathbf{p}_{g,i}$  dominiranih
primerkov.}
10: for all  $\mathbf{p}_{g+1,i} \in \mathbf{P}_{g+1}$  do
11:   for all  $\mathbf{p}_{g+1,j} \in \mathbf{P}_{g+1}$  do
12:     if Primerjaj rešitvi( $\mathbf{p}_{g,i}, \mathbf{p}_{g,j}, M$ ) = 1 then
13:        $R_{g,i} := R_{g,i} + R_{g,j}$  {Čim manjša vrednost je boljša.}
14:     end if
15:   end for
16: end for
17:  $\mathbf{n}_n = -\mathbf{1}_{|\mathbf{P}_{g+1}| \times |\mathbf{P}_{g+1}|}$  {Kvadratna matrika velikosti  $|\mathbf{P}_{g+1}|$ , za kreiranje potrebuje
 $\Theta((|\mathbf{P}_{g+1}|)^2)$ .}
18:  $\mathbf{c} = \mathbf{1}_{|\mathbf{P}_{g+1}|}$  {Vektor enic velikosti  $|\mathbf{P}_{g+1}|$ .}
19: for all  $\mathbf{p}_{g+1,i} \in \mathbf{P}_{g+1}$  do
20:    $n_{n;i,0} := i$  {Zgradimo vektor razdalj za vsak primerek.}
21:   for  $j := 0$  to  $i$  do
22:      $\sigma_{i,j} := \sigma_{j,i}$  {Že izračunane dolžine povezav samo prekopiramo.}
23:   end for
24:    $\sigma_{i,i} := 0$  {Razdalja do samega sebe je nič.}
25:   for  $j := i + 1$  to  $|\mathbf{P}_{g+1}|$  do
26:      $\sigma_{i,j} := \sum_{i=0}^M (f_i(\mathbf{x}_{g+1,i}) - f_i(\mathbf{x}_{g+1,j}))^2$  {Geometrijska razdalja,  $\Theta(M)$ .}
27:     if  $\sigma_{i,j} = 0$  then
28:        $n_{n;i,c_i} := j$  {Beležimo indekse točk, ki so podvojene: na indeksu  $i$ , zadnja
podvojena ( $c_i$ ).}
29:        $c_i := c_i + 1$  {Štejemo število kopij enake točke.}
30:        $c_j := c_j + 1$  {Štejemo število kopij enake točke — tudi v nasprotni smeri.}
31:     end if
32:   end for
33: end for
34: for  $i := 0$  to  $|\mathbf{P}_{g+1}|$  do
35:    $F(\mathbf{p}_{g,i}) := \frac{R_{g,i} + 1}{\sigma_{i,k} + 2}$  {Metriko nakopičenosti izračunamo iz moči in inverza razdalje
do  $k$ -tega najbližjega soseda.}
36: end for
```

Algoritem 9 Diferencialna evolucija za večkriterijsko optimizacijo s samoprilaganjem.

Vhod: $func_type (\mathbf{g}(\mathbf{x}) \geq 0)$ — zaporedna številka testirane funkcije in njene omejitve.

Vhod: D – dimenzija prostora spremenljivk.

Vhod: M – dimenzija prostora kriterijev.

Vhod: $MAXFEs$ – maksimalno število ovrednotenj (ustavitevni pogoj).

Vhod: NP – velikost populacije in aproksimacijske množice \bar{N} .

Vhod: s_{DE} – številka strategije mutacije DE (strategija preiskovanja).

Vhod: sel_type_{MO} – strategija večkriterijske selekcije (strategija sekanja).

Vhod: F_{init} – začetna vrednost parametrov F .

Vhod: CR_{init} – začetna vrednost parametrov CR .

Vhod: F_{min} (F_{max}) – najmanjša (največja) možna vrednost parametra F .

Vhod: CR_{min} (CR_{max}) – najmanjša (največja) možna vrednost parametra CR .

Vhod: τ – stopnja učenja za prilagajanje samoprilagodljivih parametrov F in CR .

Vhod: RNi_{seed} – seme generatorja enakomerno porazdeljenih naključnih števil.

Vhod: $front_gen$ – množica intervalov vzorčenj populacije \mathbf{P}_g po FE.

Izhod: Seznam aproksimacijskih množic iskalnih parametrov \mathcal{P} (množica matrik).

```

1:  $RNj_{seed} := \text{rand}(RNi)$  {Seme generatorja normalno porazdeljenih števil  $RNj_{seed}$ .}
2:  $g = 0$  {Z g štej generacije populacij od 0 naprej.}
3: for  $i = 0$  to  $NP - 1$  do
4:    $\mathbf{p}_{0,i} :=$  Inicializiraj nov primerek( $func\_type, D, M, F_{init}, CR_{init}, RNi, RNj$ )
5:   Ovrednoti $func\_type(\mathbf{p}_{0,i})$  {Ovrednoti primerek v začetni populaciji.}
6:    $\mathbf{P}_0 := \mathbf{P}_0 \cup \mathbf{p}_{0,i}$  {Dopolnjuj začetno populacijo  $\mathbf{P}_0$ .}
7: end for
8:  $FEs := NP$  {Ovrednotenih je  $NP$  primerkov.}
9: while  $FEs < MAXFEs$  do
10:   if  $FEs \geq front\_gen_0$  then
11:      $\mathcal{P} := \mathcal{P} \cup \mathbf{P}_g^{NP,D}$  {V  $\mathcal{P}$  dodaj  $D$  iskalnih parametrov za vseh  $NP$  primerkov.}
12:      $front\_gen := front\_gen \setminus front\_gen_0$  {Odstrani prvi element množice.}
13:   end if
14:    $\mathbf{P}_g :=$  Premešaj populacijo( $\mathbf{P}_g, NP, RNi$ ) {Naključno oštevilči osebke iz  $\mathbf{P}_g$ .}
15:   for  $i = 0$  to  $NP - 1$  do
16:      $\mathbf{u} :=$  Nov primerek $s_{DE}(\mathbf{p}_i, D, \tau, F_{min}, F_{max}, CR_{min}, CR_{max}, RNi, RNj, \mathbf{P}_g)$ 
17:     Ovrednoti $func\_type(\mathbf{u})$  {Ovrednoti primerek glede na funkcijo  $func\_type$ .}
18:      $FEs := FEs + 1$  {Beleži število ovrednotenj.}
19:     Primerjaj rešitvi( $\mathbf{p}_i, \mathbf{u}, M$ ) {Ugotovi dominiranost v prostoru kriterijev.}
20:     if  $ND_u = 1$  then
21:        $\mathbf{P}_g := (\mathbf{P}_g \setminus \mathbf{P}_i) \cup \mathbf{u}$  {Primerek  $\mathbf{u}$  dominanten, prepiši starša s primerkom.}
22:     else
23:       if  $ND_u = -1$  then
24:          $\mathbf{P}_g := \mathbf{P}_g \setminus \mathbf{p}_i$  {Starš  $\mathbf{p}_i$  dominanten, primerek  $\mathbf{u}$  izbriši.}
25:       else
26:          $\mathbf{P}_g := \mathbf{P}_g \cup \mathbf{u}$  {Dodaj primerek  $\mathbf{u}$  v aproksimacijsko množico rešitev.}
27:       end if
28:     end if
29:   end for
30:    $\mathbf{P}_{g+1} =$  Poreži $sel\_type_{MO}(\mathbf{P}_g, NP)$  {Poreži populacijo  $\mathbf{P}_g$  do velikosti  $NP$ .}
31:    $g := g + 1$  {Povečaj števec generacij.}
32: end while
```

Poglavlje 4

Rezultati

V tem poglavju predstavljamo rezultate opravljenega dela. Najprej podamo primer zagona algoritma z navedbo vhodnih in dobljenih numeričnih podatkov. Sledi časovna analiza algoritma, v kateri poleg analize časovne zahtevnosti zanjo podamo tudi empirične meritve. Kakovost algoritma izmerimo s testnimi funkcijami in dobljene rezultate ocenimo z indikatorji kakovosti. Dobljene ocene indikatorjev primerjamo še z dobljenimi ocenami indikatorjev pri sorodnih algoritmih ter označimo statistične signifikantnosti razlik med našim pristopom in sorodnimi. Po primerjavi kakovosti algoritma prikažemo še obnašanje mehanizma samoprilagajanja krmilnih parametrov, tako da njihove dobljene vrednosti prikažemo skozi zagone algoritma za različne funkcije.

4.1 Primer zagona algoritma

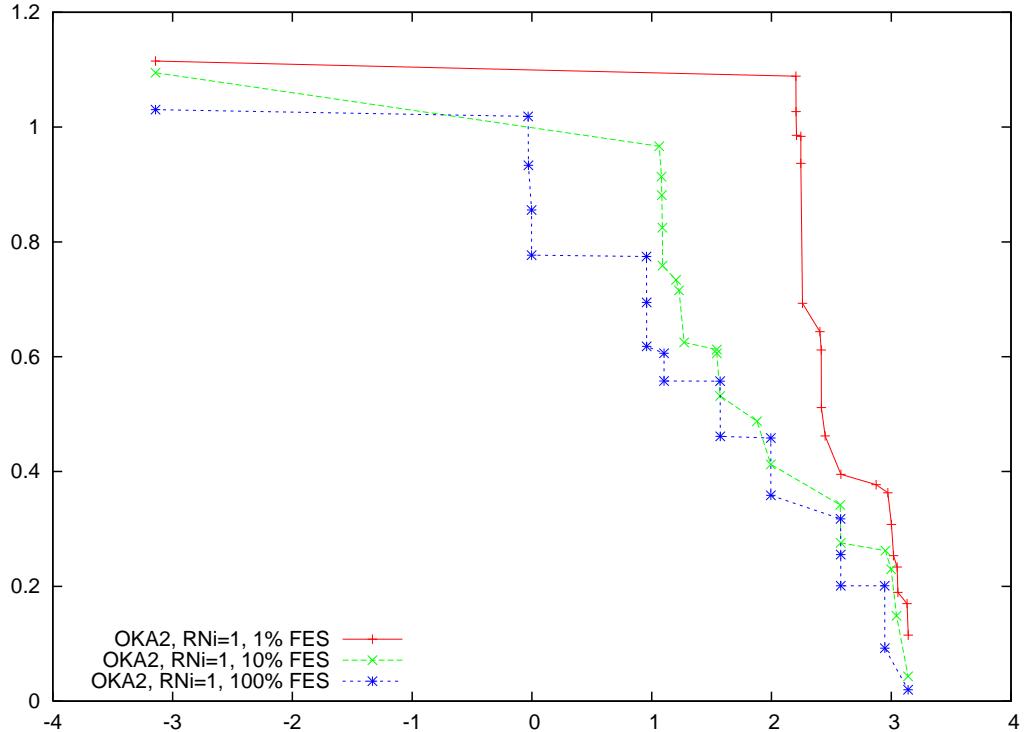
Primer zagona algoritma 9 za funkcijo OKA2:

Izhod: $\mathcal{P} = \text{DEMOwSA}($

$\text{func_type} = 1$ {Testna funkcija OKA2.},
 $D = 3$ {Prostor spremenljivk v treh dimenzijah.},
 $M = 2$ {Prostor kriterijev v dveh dimenzijah.},
 $MAXFEs = 5e+5$ {500.000 ovrednotenj.},
 $NP = 20$ {Velikost populacije in aproksimacijske množice sta enaki.},
 $s_{DE} = 13$ {'DE/rand/1/bin' strategija s samoprilagodljivimi krmilnimi parametri.},
 $sel_type_{MO} = 2$ {SPEA-II.},
 $F_{init} = 0, 5$ {Na sredi intervala $(0, 1)$.},
 $CR_{init} = 0, 3$ {Primerna vrednost za zahtevne testne funkcije (večmodalne in $M > 1$).},
 $F_{min} = 0, 1$ {Spodnja meja za F .},
 $F_{max} = 0, 9$ {Zgornja meja za F .},
 $CR_{min} = 1/24$ {Spodnja meja za CR .},

$CR_{max} = 0,9$ {Zgornja meja za $CR.$ },
 $\tau = 2/\sqrt{2D}$ {Stopnja samoprilagajanja.},
 $RNi_{seed} = 1$ {Prvi zagon, seme je 1.},
front_gen = {5e+3, 5e+4, 5e+5} {Populacijo vzorčimo po presegu teh ovrednotenj.}
)

Primer izhoda algoritma 9 za funkcijo OKA2, $\mathcal{P} = \{ \{ (-3,1416, 1,1151), (3,1416, 0,1151), (3,1323, 0,1699), (3,0552, 0,1893), (3,0500, 0,2335), (3,0189, 0,2534), (3,0018, 0,3079), (2,9717, 0,3629), (2,8739, 0,3772), (2,5794, 0,3951), (2,4483, 0,4620), (2,4170, 0,5115), (2,4162, 0,6117), (2,4044, 0,6437), (2,2598, 0,6931), (2,2045, 1,0887), (2,2049, 1,0270), (2,2454, 0,9368), (2,2089, 0,9858), (2,2449, 0,9839) \}, \{ (-3,1416, 1,0946), (1,0632, 0,9666), (1,0822, 0,9135), (1,0849, 0,8814), (1,0901, 0,8247), (1,0910, 0,7587), (1,2038, 0,7337), (1,2282, 0,7155), (1,2712, 0,6248), (1,5440, 0,6126), (1,5448, 0,6058), (1,5719, 0,5314), (1,8794, 0,4875), (1,9951, 0,4123), (2,5746, 0,3417), (2,5787, 0,2757), (3,1412, 0,0431), (3,0441, 0,1487), (2,9988, 0,2297), (2,9516, 0,2621) \}, \{ (-3,1416, 1,0303), (-0,0318, 1,0187), (-0,0290, 0,9334), (-0,0020, 0,8557), (-0,0019, 0,7768), (3,1412, 0,0197), (0,9570, 0,7746), (0,9579, 0,6945), (0,9582, 0,6181), (2,9464, 0,0921), (2,9461, 0,2009), (1,1039, 0,6060), (1,1039, 0,5576), (2,5788, 0,2010), (2,5786, 0,2553), (2,5772, 0,3174), (1,5717, 0,5575), (1,5727, 0,4612), (1,9948, 0,4583), (1,9952, 0,3585) \} \} \}.$



Slika 4.1: Aproksimacijske množice DEMOwSA po različnem številu ovrednotenj funkcije OKA2, z $RNi_{seed} = 1$ in $NP = 20$.

Kot vidimo iz slike 4.1, se aproksimacijske množice s povečevanjem števila ovrednotenj izboljšujejo, tj. dobljene rešitve so si vedno bližje in bližje optimumom.



4.2 Časovna analiza algoritma

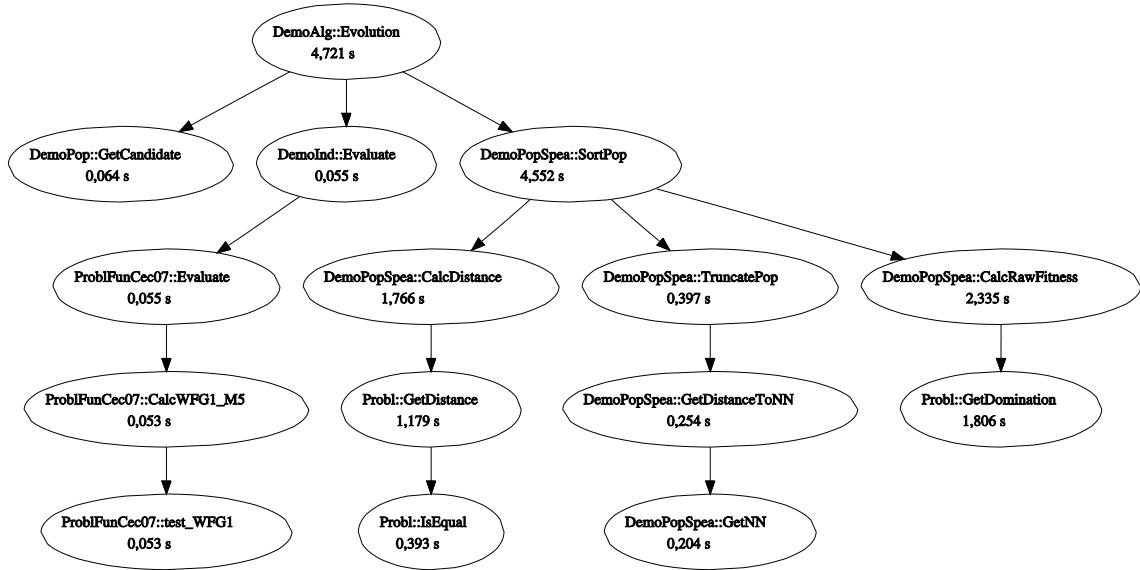
Ocena zahtevnosti večkriterijskih evolucijskih optimizacijskih algoritmov je najpogo-
steje predstavljena z oceno potrebnega števila ovrednotenj za pokritje Pareto fronte.
Obstaja malo študij, kjer so to oceno zapisali v obliki reda kompleksnosti [92], ostale
študije so bolj empirične.

Analitično oceno zahtevnosti algoritma DE podaja delo Zielinskijeve in sodelav-
cev [41], ki poudarja tudi, da za izračun zahtevnosti moramo poznati ustavitev
pogoj optimizacije. Ta je lahko odvisen od števila ovrednotenj kriterijse funkcije
(ang. *number of function evaluations – FE*).

Ustavitev pogoj optimizacije, ki ga bomo izbrali v tem delu, je omejitev naj-
večjega števila ovrednotenj (*MAXFEs*). Opomnimo, da ne bomo podajali analize,
koliko *MAXFEs* algoritom potrebuje za ustavitev. Ob tem je namreč vredno spo-
mniti na izrek *no free lunch* [93, 94], ki v tem primeru lahko pomeni, da lahko
podamo zgolj analizo večkriterijskega evolucijskega algoritma, ki temelji na točno
določenem večkriterijskem problemu. Vsakršna splošna analiza takšnega algoritma
(tj. na naboru vseh obstoječih funkcij) je podvržena omenjenemu izreku in zato ne-
smiselna. Tukaj zato podajamo meritve profilirnika, ki podaja časovno zahtevnost
algoritma, študijo odvisnosti od števila ovrednotenj pa podajamo v razdelku 4.3.3.

Na sliki 4.2 vidimo drevo klicev glavnih funkcij programa za 5000 ovrednotenj
kriterijske funkcije WFG1_M5. Čas je merjen v sekundah, skupni čas teka pro-
grama je približno 5 sekund. Vidimo, da program precej časa porabi za ureja-
nje rešitev po njihovi moči. Urejanje se izvede v metodi **SortPop**, ki moč rešitve
izračuna nad njenim kriterijskim vektorjem s pomočjo grobe uspešnosti v metodi
CalcRawFitness in razdalje do najbližjega sosednega kriterijskega vektorja v me-
todi **CalcDistance**. Metoda **SortPop** nato izvede selekcijo in iz aproksimacijske
množice odstrani slabe rešitve. Program veliko manj časa porabi za izdelavo novih
rešitev (metoda **GetCandidate**). Čeprav se zdi, da čas ovrednotenja ni tako velik
(metoda **Evaluate**, iz katere se posredno izvede metoda **test_WFG1** z ustreznimi
parametri), je pomembno poudariti, da čas ovrednotenja rešitev realnih problemov
pogosto presegajo čas ovrednotenja rešitev uporabljenih testnih funkcij. Čas
ovrednotenja je namreč v realnih problemih še nekaj redov večji, tako da časovna
zahtevnost optimizatorja postane zanemarljiva in za oceno kakovosti algoritma po-
stane važno le število ovrednotenj.

V tabeli 4.1 podajamo čas računanja 10.000 ovrednotenj vseh funkcij v ogrodju
CEC 2007 [2] skupaj (T_1) in enako, skupaj s časom algoritma (T_2).



Slika 4.2: Drevo klicev glavnih funkcij programa za 5000 ovrednotenj kriterijske funkcije WFG1_M5.

Tabela 4.1: Časovna zahtevnost algoritma.

T_1	T_2	$(T_2 - T_1)/T_1$
1,06 s	27,693 s	25,13

4.2.1 Analitična ocena računske zahtevnosti algoritma

Čeprav smo že navedli, da je analiza računske zahtevnosti evolucijskega algoritma težka naloga, se je bomo tukaj vseeno lotili. Tukaj se omejimo zgolj na uporabljene algoritme in splošne primerjave ostalih sorodnih algoritmov zaradi potrebne preobsežne analize ne bomo podali. Napovejmo torej računsko zahtevnost zapisanih algoritmov v odvisnosti od njihovih vhodnih podatkov ob znanem ustavitevem pogoju. Za test smo uporabili računalniški sistem, ki je zapisan v poglavju 4.3.3. Prvi test računske zahtevnosti kaže tabela 4.2. Pri tem D in M označujeta dimenzijo prostora spremenljivk oz. prostora kriterijev optimizacijskega problema, NP velikost populacije in aproksimacijske množice \bar{N} ter G število generacij:

$$G = \left\lceil \frac{\text{MAXFEs}}{NP} \right\rceil,$$

kjer MAXFEs predstavlja maksimalno število ovrednotenj (ustavitevni pogoj). V algoritmu 7 pričakovana zahtevnost je kvadratna glede na NP zaradi urejenosti seznama primerkov po nakopičenosti.

Model časovne zahtevnosti programa v odvisnosti od števila ovrednotenj (FEs) za polinomsko odvisnost a -te stopnje je $O(FEs) = c \cdot (FEs)^a$, kjer za izračun pro-

Tabela 4.2: Ocene računske zahtevnosti algoritma DEMOwSA.

Alg.	Ime	Ocenjena računska zahtevnost
1	Ovrednoti _{OKA2}	$\Theta(MD)$
2	Primerjaj rešitvi	$\Theta(M)$
3	Premešaj populacijo _{SPEA-II}	$\Theta(NP \log NP + 2NP)$
4	InicIALIZIRAJ nov primerek	$\Theta(D)$
5	Nov primerek _{DEMOwSA/rand/1/bin}	$\Theta(D)$
6	Poreži _{SPEA-II}	$\Theta(((NP)^2 M) + (3(NP)^2 M + (NP)^2 \log NP))$
7	Odstrani prenakičene rešitve _{SPEA-II}	$\Theta((NP)^2 M)$
8	Nakopičenost _{SPEA-II}	$\Theta(3(NP)^2 M + (NP)^2 \log NP)$
9	DEMOwSA	$\Theta(G(5M(NP)^2))$

Tabela 4.3: Čas izvajanja algoritma DEMOwSA pri različnih MAXFEs.

Alg.	Tip funkcije	NP	M	D	5e+3 FE	5e+4 FE	5e+5 FE	$\bar{c} (\sigma(\bar{c}))$	$\bar{a} (\sigma(\bar{a}))$	$\Theta(FE)$
9	OKA2	100	2	3	116 ms	988 ms	8285 ms	0,04 (0)	0,93 (0)	DA
9	S_ZDT1	100	2	30	92 ms	992 ms	12153 ms	0,01 (0)	1,06 (0,04)	DA
9	S_DTLZ2_M3	150	3	30	168 ms	2052 ms	23689 ms	0,02 (0)	1,07 (0,02)	DA
9	WFG1_M3	150	3	24	328 ms	3644 ms	35262 ms	0,06 (0,03)	1,02 (0,04)	DA
9	S_DTLZ2_M5	800	5	28	1276 ms	12809 ms	157202 ms	0,17 (0,11)	1,05 (0,06)	DA
9	WFG1_M5	800	5	28	1708 ms	23069 ms	241367 ms	0,24 (0,18)	1,08 (0,08)	DA
9	S_DTLZ2_M5	3000	5	28	4060 ms	51711 ms	703380 ms	0,29 (0,06)	1,12 (0,02)	DA
9	WFG1_M5	3000	5	28	4860 ms	108415 ms	1539784 ms	0,23 (0,26)	1,25 (0,14)	DA

blema velikosti FEs potrebujemo c -krat po $(FEs)^a$ operacij. Pri empiričnih meritvah na računalniškem sistemu izmerimo čas izvajanja programa, ki predstavlja časovno zahtevnost $T(FEs) = O(FEs)$. Pri dveh različnih številih ovrednotej $FEs_1, FEs_2 \in [1, MAXFEs]$ opravljenima z dvema časovnima meritvama $T(FEs_1) = c \cdot (FEs_1)^a$ in $T(FEs_2) = c \cdot (FEs_2)^a$ izrazimo konstanti a in c , ki ju nato izračunamo kot:

$$a = \frac{\ln \frac{T_1}{T_2}}{\ln \frac{FEs_1}{FEs_2}} \text{ in } c = \frac{T_1}{FEs_1^a}.$$

Označimo z \bar{c} in \bar{a} povprečni izmerjeni vrednosti za konstanto c in eksponent a , ki ju dobimo z več meritvami za posamezno FEs . Dejanski čas izvajanja algoritma DEMOwSA in potrditev analitičnega modela zahtevnosti podaja tabela 4.3.

Tabela 4.3 za algoritem 9 potrjuje v zadnji vrstici tabele 4.2 napovedano polinomsko (linearno) odvisnost algoritma od števila ovrednotenj. Ker sta število ovrednotenj in število generacij linearno soodvisni, je zahtevnost algoritma tudi od števila generacij odvisna linearno. Do manjšega odstopanja pri odvisnosti od števila generacij pride zaradi zaokrožitve navzdol, saj zaradi postavljenega ustavitevnega pogoja za določeno generacijo naredimo več ovrednotenj kriterijske funkcije, kot je potrebno. Tako je dejansko število ovrednotenj FEs pri 5e+3 enako 5600 in pri 5e+4 enako 50400, pri 5e+5 pa ostane nespremenjeno, tj. natančno 500000. Konstanta c se za različne funkcije zaradi stopnjevanja zahtevnosti kriterijskih funkcij povečuje, kar je pričakovano.

Tabelo 4.3 lahko analiziramo še z drugega vidika. V odvisnosti od velikosti populacije NP lahko potrdimo tudi $\Theta((NP)^2)$ zahtevnost algoritma. Vendar v ta namen raje poglejmo tabelo 4.4. Ta v indeksih imen funkcij vsebuje informacijo o velikosti populacije NP . Tukaj smo kot G izbrali fiksno vrednost in zato $MAXFEs \in \{10000, 15000, 80000, 300000\}$ pri $G = 100$ ter sorazmerno manj za $G = 50$ (dvakrat manj) in $G = 10$ (desetkrat manj).

Tabela 4.4: Čas izvajanja algoritma DEMOWSA pri različnih NP .

Alg.	G	$WFG1_M3_{100}$	$WFG1_M3_{150}$	$WFG1_M3_{800}$	$WFG1_M3_{3000}$	\bar{c}	\bar{a}	$\Theta((NP)^2)$
9	10	32 ms	0080 ms	1872 ms	27562 ms	0,03 (0,03)	1,97 (0,18)	DA
9	50	221 ms	492 ms	10822 ms	173231 ms	0 (0)	2,06 (0,11)	DA
9	100	456 ms	1060 ms	22581 ms	366496 ms	0,05 (0,07)	2,01 (0,2)	DA

4.3 Analiza kakovosti algoritma

Razvoj raznolikih algoritmov za večkriterijsko optimizacijo je botroval številnim primerjavam teh algoritmov. Cilj primerjav je bil v splošnem prikazati izboljšavo enega algoritma glede na svoje vrstnike [95]. Čeprav v splošnem velja t.i. izrek *no free lunch* [93, 94], še vedno ostaja želja po primerjavi in klasifikaciji teh algoritmov.

4.3.1 Testne funkcije

Do sedaj je nastalo že veliko funkcij za testiranje učinkovitosti in primerjavo med evolucijskimi algoritmi. Ustaljen proces primerjave teh algoritmov je navadno povezan z veliko množico umetno ustvarjenih večkriterijskih testnih problemov (testov) in široko paleto meritvenih tehnik (metrik) za primerjavo rezultatov nad temi testi. Predstavimo tipičen scenarij za primerjavo evolucijskih algoritmov [95].

1. Določimo množico algoritmov, ki jih bomo primerjali.
2. Izberemo množico obstoječih (najbolje merljivih) testnih primerov ali pripravimo nove.
3. Izberemo množico meritov, s katerimi bomo preverili množico rezultatov evolucijskih algoritmov.
4. Priskrbimo rezultate za vsakega od evolucijskih algoritmov, s spleta ali z implementacijo.
5. Izvedemo meritve in primerjamo rezultate.
6. Podamo ugotovitve.

Mnoge od uporabljenih večkriterijskih *testnih funkcij* EA iz literature niso bile rigorozno analizirane, ampak zgolj de facto uporabljane. Posledica tega je, da je težko povleči točne sklepe o prednostih in slabostih algoritmov, ki jih preskusimo s temi testi. V prispevku [95] najdemo sistematičen pregled in analizo več testnih funkcij iz literature EA. Te funkcije so vse iz razreda večkriterijskih funkcij nad realnimi števili brez omejitev. Ta analiza testnih problemov izpostavlja številna področja, ki zahtevajo pozornost. Ne samo, da so številni preostali testni problemi slabo pripravljeni, ampak je pomembna skupina *neločljivih večmodalnih* problemov slabo zasedana, prav tako obstaja le malo parametričnih testnih ogrodij z nastavljivo težavnostno stopnjo. Ogrodje testnih funkcij WFG te pomanjkljivosti odpravlja.

Testne funkcije v testnem ogrodju CEC 2007

Testno ogrodje na posebni sekciji za oceno učinkovitosti večkriterijskih optimizacijskih algoritmov na CEC 2007 vsebuje 13 testnih funkcij [2], izmed katerih je 6 funkcij podvojenih z manjšimi spremembami, kar skupno nanese 19 različnih večkriterijskih funkcij za optimizacijo. Vse funkcije sestojijo iz f_1, \dots, f_M enokriterijskih funkcij, ki jih je potrebno minimizirati. Posamezne testne funkcije so natančneje opisane v dodatku B.

4.3.2 Metrike učinkovitosti

Metrike učinkovitosti oz. indikatorji kakovosti (za razliko med pojnama glej razlago z utemeljitvijo v [16], str. 6) numerično izmerijo uspeh aproksimacijske množice (approximation set) nedominiranih rešitev. Metrike kažejo zgolj videnje ocenjevalca, kar pomeni, da sta lahko dve rešitvi z različnimi metrikami popolnoma različno ocenjeni oz. relativno razvrščeni glede na preostale rešitve.

Metrike učinkovitosti se ponavadi računajo glede na podano referenčno aproksimacijsko množico, potem se izračuna razlika med našo aproksimacijsko množico in referenčno, boljši rezultat je tisti, ki je manjši. Poleg referenčne aproksimacijske množice pogosto potrebujemo tudi referenčno točko, ki navzgor omejuje vse rešitve, tj. je slabša po vsakem od kriterijev od vseh rešitev v vsaki naši aproksimacijski množici. To je najlažje izvesti tako, da dobljene vrednosti kriterijev normaliziramo med [1, 2], referenčno točko pa postavimo na vrednost 2, 1 za vsak kriterij.

Najbolj znan indikator kakovosti je hipervolumen $I_{\overline{H}}$ [96, 97, 98, 99], ki meri dominiran prostor rešitev, ki ga je algoritem že našel. Metrika deluje nad diskretnim prostorom kriterijev, tako da zvezen prostor Z diskretizira v diskretni prostor Ω . Ta indikator meri učinkovitost obeh ciljev večkriterijske optimizacije, tako porazdelitev rešitev po Pareto fronti kot tudi bližino k Pareto fronti.



Indikator I_R [100] pokaže zgolj oddaljenost od Pareto fronte, pri tem pa vsak kriterijski vektor pretvori v realno vrednost. Pri pretvorbi upošteva pomožno funkcijo za utežitev kriterijev (npr. funkcijo Čebiševa).

Poleg numeričnih metrik učinkovitosti je dostikrat uporabna tudi vizualizacija dobljenih aproksimacijskih množic. Ena od takšnih metod je izris ploskev (npr. 50%) dosega. To ploskev dobimo tako, da na diskretnem prostoru kriterijev ugotovimo, katere rešitve je iskalni algoritem dosegel (npr. v 50% zagonov).

Metrike učinkovitosti [16] za oceno dobljenih aproksimacijskih množic smo dobili iz orodja PISA [101]. Indikatorja kakovosti, indikator $I_{\bar{H}}$ [96] in indikator I_R [100], sta izračunana za vse funkcije. Za funkcijo SYM-PART je izračunan še indikator CS, da bi ugotovili pokritost množic iskalnih parametrov (v dodatku glej sliko B.1).

Ploskve dosega [102] so prav tako izračunane s pomočjo [101]. Na koncu podajamo še časovno zahtevnost algoritma.

4.3.3 Rezultati simulacij

Predstavljen algoritem 9 smo 25-krat zagnali na 19 testnih funkcijah [2] posebne sekcije za oceno učinkovitosti večkriterijskih algoritmov za optimizacijo na IEEE Congress on Evolutionary Computation (CEC 2007). Med zagoni smo spreminali zgolj parameter RNi_{seed} iz algoritma 9, tj. $RNi_{seed} = 1, 2, \dots, 25$. Na podlagi dobljenih aproksimacijskih množic smo po $5e+3$, $5e+4$ in $5e+5$ ovrednotenjih vektorske kriterijske funkcije pognali metrike učinkovitosti, ki so izmerile učinkovitost algoritma. Predstavljamo rezultate z velikostjo zunanjega arhiva do $\bar{N} = 800$.

Vsi predstavljeni testi so bili opravljeni na sistemu labraj2.uni-mb.si, z jedrom Linux, 2.6.17-1.2142_FC4smp. Tip centralnega procesorja v računalniku je 64-bit Dual Core AMD Opteron(tm) Processor 280 (2390,651 MHz, 4787,74 bogomips) in je opremljen z 8 GB RAM. Algoritem je objektno implementiran v programskem jeziku C++.

Algoritem DEMOWSA ima več parametrov, izmed katerih sta najpomembnejša izbira selekcij in strategije za izdelavo novih kandidatnih rešitev. Kot selekcijo smo izbrali algoritem po vzoru SPEA-II. Za strategijo izdelave kandidatov smo izbrali samoprilagodljivi algoritem diferencialne evolucije. Prilagajali smo parametra F in CR . Maksimalni dinamični razpon teh parametrov je $F \in [0, 2]$ in $CR \in [0, 1]$. Ta parametrov smo za zagon omejili na $0,1 \leq F \leq 0,9$ in $1/24 \leq CR \leq 0,9$. Parameter τ smo nastavili na $2/\sqrt{2D}$, kar sovpada s priporočili iz [43]. Ocenjen strošek prilaganja parametrov je majhen (in ničen glede na število ovrednotenj) v primerjavi s celotnim algoritmom, saj gre za preprost, kanoničen izračun prilagodljivih parametrov. Uporabljeni testni ogrodje daje naslednje rezultate:

Tabela 4.5: Vrednosti indikatorja pokritosti množic za testno funkcijo SYM-PART.

<i>FES</i>	5e+3	5e+4	5e+5
Najboljši	1,0000e+00	1,0000e+00	1,0000e+00
Srednji	1,0000e+00	1,0000e+00	1,0000e+00
Najslabši	1,0000e+00	1,0000e+00	1,0000e+00
Poprečje	1,0000e+00	1,0000e+00	1,0000e+00
Std	0,0000e+00	0,0000e+00	0,0000e+00

- najboljša, medialna, najslabša in povprečna vrednost indikatorjev I_R in $I_{\bar{H}}$ ter njena standardna deviacija za vseh 25 zagonov za testne funkcije 1–7 in 8–13 (z $M = 3$ in $M = 5$) po 5e+3, 5e+4 in 5e+5 ovrednotenjih testne funkcije: tri tabele za vsakega od indikatorjev so priložene iz teh rezultatov (glej tabele 4.6, 4.7, 4.8, 4.9, 4.10 in 4.11),
- najboljša, medialna, najslabša in povprečna vrednost indikatorja CS ter njena standardna deviacija za vseh 25 zagonov za testno funkcijo SYM-PART po 5e+3, 5e+4 in 5e+5 ovrednotenjih testne funkcije (glej tabelo 4.5) – algoritem venomer najde le po eno Pareto optimalno množico rešitev v iskalnem prostoru,
- ploskve 0%, 50% in 100% dosega vseh 25 zagonov na testnih funkcijah 1–7 po 5e+5 ovrednotenjih (glej slike 4.3.3–4.3.3),
- ploskve 50% dosega vseh 25 zagonov na testnih funkcijah 8–13 z $M = 3$ po 5e+5 ovrednotenjih (glej sliko 4.10); zgornji grafi so za WFG8, spodnji za WFG9. Obseg in oznako osi je možno razbrati iz slik na diagonali. Za WFG8 oznaka v stolpcu označuje absciso in oznaka v vrstici označuje ordinato, ter obratno za WFG9. Iz slik vidimo, da algoritem prostor kriterijev dobro pokrije in
- aproksimacijske množice pri zagonu z medialno vrednostjo indikatorja I_R po 5e+5 ovrednotenjih testnih funkcij 12 in 13 (glej sliko 4.11).

Iz vseh dobljenih tabel za uporabljene indikatorje je moč opaziti, da je algoritem v veliki meri učinkovit glede na te indikatorje do bližine 0,05 referenčni množici za vse dane testne funkcije, z manjšo izjemo za WFG1 (manjše vrednosti indikatorjev so boljše). Referenčne množice imajo vrednosti indikatorjev enako 0. V primeru, ko je dobljena vrednost indikatorja negativna, aproksimacijska množica presega kakovost podane referenčne množice z ozirom na podan indikator. Podajmo krajšo diskusijo dobljenih povprečnih vrednosti indikatorjev za podane testne funkcije.

Iz tabele 4.6 je videti, da je naš algoritem učinkovit v primerjavi z referenčno množico na funkcijah OKA2, SYM-PART, S_ZDT1, S_ZDT4, R_ZDT4 in S_ZDT6.

Funkcija S_ZDT2 je najtežja med danimi dvokriterijskimi funkcijami. Indikator I_R za funkcijo OKA2 pokaže negativno vrednost, v povprečju že pri $5e+4$ ovrednotenjih (FEs). Pri funkcijah OKA2, SYM-PART, S_ZDT1 in S_ZDT2 vidimo, da tukaj algoritom že pri $5e+4$ ovrednotenjih doseže podobno kakovostne rešitve, kot pri $5e+5$ ovrednotenjih.

Iz tabele 4.7 vidimo, da algoritom dobro rešuje tudi funkcije S_DTLZ2, R_DTLZ2 in S_DTLZ3, kot tudi funkciji WFG8 in WFG9. V tej tabeli je najbolje dosežena funkcija WFG8 in najslabše funkcija WFG1. V tabeli 4.8 vidimo, da tudi za testne probleme z $M = 5$ algoritom doseže dobre rezultate, ocenjeno z indikatorjem I_R . Tukaj je najbolje ocenjena funkcija S_DTLZ3, največ težav pa algoritmu povzroči funkcija WFG1. Funkcijo WFG8 algoritom v najboljšem zagonu reši bolje, kot je zastavljena referenčna aproksimacijska množica.

Iz tabele 4.9 opazimo, da algoritom, glede na indikator hipervolumna, funkcije OKA2 ne najde boljših rešitev od podane referenčne množice, kar je lep primer razloga, zakaj uporabiti več indikatorjev za oceno evolucijskega algoritma – indikator namreč daje le model videnja potencialnega (človeškega) odločevalca na porazdelitev dobljenih kriterijskih vektorjev po Pareto fronti. Z uporabo več indikatorjev se lahko o kakovosti algoritma tako lažje prepričamo oz. dobimo boljši vpogled. V primeru funkcije OKA2 indikator hipervolumna tako daje dokaj slabe rezultate, kar hitro opazimo tudi iz priložene slike za ploskev dosega (slika 4.3.3), vendar je uspešen pri iskanju enakomerno porazdeljenih kriterijskih vektorjev blizu njene Pareto fronte. Pri ostalih funkcijah indikator hipervolumna pokaže boljše rezultate, predvsem pri funkciji SYM-PART in funkcijah S_ZDT1, S_ZDT4, R_ZDT4 in S_ZDT6. Funkcija S_ZDT2 ima največjo vrednost tega indikatorja. Tabela 4.10 kaže podobno sliko kot tabela 4.8, kjer je funkcija S_DTLZ3 v povprečju še bolje ocenjena, prav tako velja za funkciji WFG8 in WFG89.

V tabeli 4.11 znova opazimo razhajanje med indikatorjema, tokrat pri funkcijah WFG8 in WFG9, ki nakazuje, da je algoritom Pareto fronto teh dveh funkcij glede na ta indikator vseeno dokaj dobro dosegel, saj je celo presegel podano referenčno aproksimacijsko množico. Najtežja funkcija iz tega nabora ostaja WFG1, tudi za R_DTLZ2 je ocena nekoliko slabša, vendar je za S_DTLZ2 in S_DTLZ3 vrednost indikatorja po $5e+5$ glede na indikator I_R celo boljša.

Sedaj tudi na empiričnih rezultatih vidimo, da za naš algoritom velja izrek *no free lunch* [93, 94], saj algoritom ne reši vseh funkcij z enako kakovostjo. Zato si poglejmo, kako drugi algoritmi rešujejo enake testne funkcije in podajmo statistično oceno razlik. V ta namen bomo v naslednjem razdelku kakovost algoritma izrazili v relaciji s konkurenčnimi algoritmi in ocene rangirali.

Tabela 4.6: Vrednosti indikatorja I_R na testnih funkcijah 1–7.

FEs		1. OKA2	2. SYMPART	3. S_ZDT1	4. S_ZDT2	5. S_ZDT4	6. R_ZDT4	7. S_ZDT6
$5e+3$	Najboljši	-9,3342e-05	1,8065e-02	3,2227e-02	9,0227e-02	7,7621e-02	1,8463e-02	1,3063e-01
	Srednji	1,4285e-03	3,2901e-02	4,2786e-02	1,0564e-01	8,7825e-02	3,2837e-02	1,3880e-01
	Najslabši	4,1012e-03	5,1199e-02	5,5653e-02	1,3739e-01	9,6298e-02	3,8180e-02	1,4313e-01
	Poprečje	1,6949e-03	3,2040e-02	4,3756e-02	1,0598e-01	8,7295e-02	3,0508e-02	1,3841e-01
	Std	1,0779e-03	7,7663e-03	5,5966e-03	9,6085e-03	4,4246e-03	5,5133e-03	2,8771e-03
$5e+4$	Najboljši	-8,8465e-04	7,8014e-06	1,8116e-05	3,1987e-05	4,4734e-03	6,5665e-04	2,6095e-03
	Srednji	-5,9176e-04	1,1593e-05	3,3985e-05	4,8270e-04	1,1250e-02	2,3156e-03	1,9932e-02
	Najslabši	1,1057e-03	1,6756e-05	3,5685e-04	4,0447e-02	1,8088e-02	7,6808e-03	5,3947e-02
	Poprečje	-4,9518e-04	1,2194e-05	4,9882e-05	1,7773e-02	1,1658e-02	2,9961e-03	2,0990e-02
	Std	4,3799e-04	2,2467e-06	6,6400e-05	2,0265e-02	3,3856e-03	2,0438e-03	1,5540e-02
$5e+5$	Najboljši	-9,9912e-04	1,2418e-06	3,5832e-09	8,8060e-08	-5,1506e-09	1,1625e-04	-1,0216e-06
	Srednji	-9,1005e-04	1,5838e-06	4,9212e-07	1,2702e-06	3,0912e-04	5,2437e-04	-1,0216e-06
	Najslabši	-6,5077e-05	2,0972e-06	5,4084e-06	4,0190e-02	6,8877e-04	1,4756e-03	4,5350e-02
	Poprečje	-8,4180e-04	1,5972e-06	9,3739e-07	1,7634e-02	2,6829e-04	5,7458e-04	3,6224e-03
	Std	2,2104e-04	1,9206e-07	1,2040e-06	2,0304e-02	1,8110e-04	3,1949e-04	1,2536e-02

Tabela 4.7: Vrednosti indikatorja I_R na testnih funkcijah 8–13 z $M = 3$.

FEs		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
$5e+3$	Najboljši	8,7235e-05	3,3085e-04	2,7835e-04	5,5997e-02	-1,3746e-02	-8,6648e-03
	Srednji	1,4036e-04	1,8578e-03	3,6502e-04	5,6298e-02	-1,1561e-02	-6,8257e-03
	Najslabši	2,6565e-04	4,3200e-03	4,8230e-04	5,6504e-02	-8,1226e-03	-3,8443e-03
	Poprečje	1,6191e-04	1,9075e-03	3,7112e-04	5,6270e-02	-1,1157e-02	-6,7864e-03
	Std	5,3717e-05	9,5922e-04	4,4777e-05	1,3757e-04	1,3721e-03	1,0520e-03
$5e+4$	Najboljši	4,9802e-06	1,9487e-04	3,4630e-05	5,3504e-02	-2,4940e-02	-8,9855e-03
	Srednji	1,8532e-05	7,3927e-04	4,7379e-05	5,4632e-02	-2,2843e-02	-7,8004e-03
	Najslabši	4,8125e-05	1,9698e-03	6,3080e-05	5,5076e-02	-2,0654e-02	-5,8255e-03
	Poprečje	2,1649e-05	8,3688e-04	4,7184e-05	5,4462e-02	-2,2823e-02	-7,8577e-03
	Std	1,2187e-05	5,2766e-04	6,2961e-06	4,1577e-04	1,1082e-03	6,8295e-04
$5e+5$	Najboljši	3,1669e-06	1,9488e-04	1,7962e-07	3,7746e-02	-2,6671e-02	-8,9602e-03
	Srednji	1,9806e-05	2,7225e-04	1,3207e-06	4,1318e-02	-2,5825e-02	-7,8284e-03
	Najslabši	3,7352e-05	1,8596e-03	5,9104e-06	4,3197e-02	-2,4263e-02	-6,3708e-03
	Poprečje	1,9844e-05	4,4428e-04	1,6725e-06	4,0943e-02	-2,5818e-02	-7,8520e-03
	Std	8,9721e-06	3,9119e-04	1,4308e-06	1,6153e-03	6,0301e-04	6,4689e-04

Tabela 4.8: Vrednosti indikatorja I_R na testnih funkcijah 8–13 z $M = 5$.

FEs		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
$5e+3$	Najboljši	2,1630e-04	1,9067e-04	2,0344e-04	4,7001e-02	6,0313e-03	7,1558e-03
	Srednji	3,6520e-04	4,6050e-04	2,5665e-04	4,7203e-02	9,7476e-03	1,0726e-02
	Najslabši	6,3009e-04	1,1045e-03	3,0923e-04	4,7368e-02	1,2483e-02	1,5007e-02
	Poprečje	3,9405e-04	5,5613e-04	2,5820e-04	4,7190e-02	9,7101e-03	1,0799e-02
	Std	1,1339e-04	2,9624e-04	2,8835e-05	8,8086e-05	1,2003e-03	1,9437e-03
$5e+4$	Najboljši	1,5920e-05	8,6488e-05	1,8228e-05	4,6874e-02	2,7767e-03	1,6274e-03
	Srednji	2,4497e-05	1,3545e-04	2,2680e-05	4,7030e-02	5,2470e-03	2,6175e-03
	Najslabši	3,7371e-05	6,0184e-04	3,2126e-05	4,7247e-02	7,6459e-03	4,0949e-03
	Poprečje	2,5081e-05	1,8857e-04	2,2831e-05	4,7046e-02	5,3152e-03	2,7226e-03
	Std	4,5533e-06	1,2360e-04	3,1608e-06	9,9580e-05	9,2150e-04	5,9080e-04
$5e+5$	Najboljši	7,0462e-06	7,3698e-05	1,3937e-06	4,5506e-02	-1,3866e-03	1,3504e-03
	Srednji	1,0496e-05	9,3354e-05	1,9857e-06	4,6004e-02	9,1131e-04	2,4433e-03
	Najslabši	1,8683e-05	2,2076e-04	3,0883e-06	4,6334e-02	2,7408e-03	3,2385e-03
	Poprečje	1,0820e-05	9,8614e-05	2,0119e-06	4,5998e-02	6,5218e-04	2,3653e-03
	Std	2,7352e-06	3,0240e-05	4,3690e-07	2,2415e-04	1,0519e-03	4,9068e-04

Tabela 4.9: Vrednosti indikatorja $I_{\bar{H}}$ na testnih funkcijah 1–7.

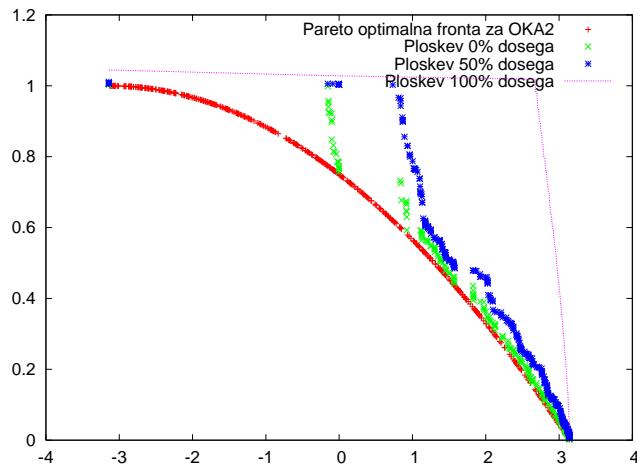
FEs		1. OKA2	2. SYMPART	3. S_ZDT1	4. S_ZDT2	5. S_ZDT4	6. R_ZDT4	7. S_ZDT6
$5e+3$	Najboljši	2,2842e-02	5,1690e-02	1,2873e-01	2,2216e-01	2,3436e-01	5,4872e-02	3,3331e-01
	Srednji	2,9697e-02	9,3333e-02	1,5892e-01	2,5407e-01	2,6935e-01	9,8669e-02	3,5110e-01
	Najslabši	3,7584e-02	1,4359e-01	1,9403e-01	2,9029e-01	2,9729e-01	1,1361e-01	3,6248e-01
	Poprečje	2,9818e-02	9,0834e-02	1,5962e-01	2,5562e-01	2,6579e-01	9,3577e-02	3,5019e-01
	Std	3,8085e-03	2,1603e-02	1,6371e-02	2,0018e-02	1,4416e-02	1,4859e-02	7,5076e-03
$5e+4$	Najboljši	1,1653e-02	2,3635e-05	2,2544e-04	2,8737e-04	1,3558e-02	2,4399e-03	5,4948e-03
	Srednji	1,6851e-02	3,4058e-05	2,9960e-04	1,2374e-03	3,3302e-02	7,1319e-03	3,9101e-02
	Najslabši	3,0766e-02	4,8472e-05	1,5224e-03	4,8998e-02	5,3351e-02	2,3019e-02	1,2358e-01
	Poprečje	1,7700e-02	3,5867e-05	3,5656e-04	2,1533e-02	3,4509e-02	9,3785e-03	4,7118e-02
	Std	4,6298e-03	6,3677e-06	2,5186e-04	2,4101e-02	9,9004e-03	6,0704e-03	3,5745e-02
$5e+5$	Najboljši	5,2590e-03	3,6018e-06	1,3953e-04	1,9139e-04	8,4608e-07	3,2074e-04	-2,3094e-04
	Srednji	7,6196e-03	4,7420e-06	1,4616e-04	2,1250e-04	1,4826e-03	1,7670e-03	-2,2958e-04
	Najslabši	2,4920e-02	6,1155e-06	1,6669e-04	4,8226e-02	2,5823e-03	4,4835e-03	1,0198e-01
	Poprečje	9,3329e-03	4,7780e-06	1,4747e-04	2,1184e-02	1,2154e-03	1,7640e-03	7,9347e-03
	Std	5,8079e-03	5,4239e-07	5,1448e-06	2,4157e-02	7,3065e-04	9,5558e-04	2,8247e-02

Tabela 4.10: Vrednosti indikatorja $I_{\bar{H}}$ na testnih funkcijah 8–13 z $M = 3$.

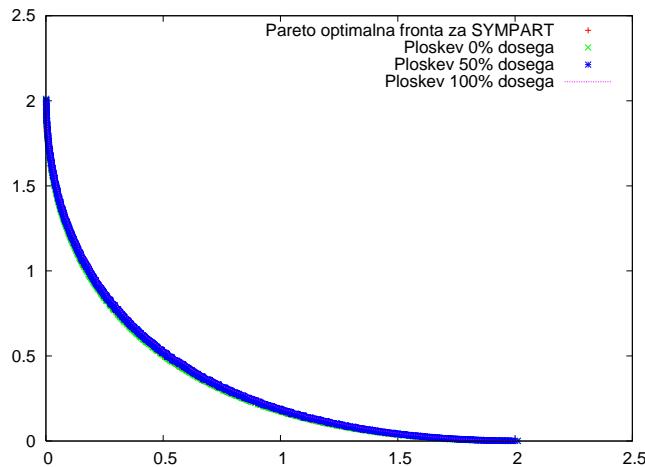
FEs		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
$5e+3$	Najboljši	1,7329e-03	1,5205e-02	6,9113e-03	2,8689e-01	-9,6992e-02	-4,7792e-02
	Srednji	2,3973e-03	2,2871e-02	1,0502e-02	2,8825e-01	-8,4202e-02	-4,1525e-02
	Najslabši	3,4646e-03	3,5751e-02	1,2542e-02	2,8918e-01	-7,1539e-02	-3,1877e-02
	Poprečje	2,5855e-03	2,3041e-02	1,0277e-02	2,8810e-01	-8,3876e-02	-4,1596e-02
	Std	5,3366e-04	5,0958e-03	1,5192e-03	5,9392e-04	5,9293e-03	3,6534e-03
$5e+4$	Najboljši	4,2147e-05	1,1258e-02	1,1726e-05	2,7333e-01	-1,5768e-01	-5,7822e-02
	Srednji	8,8024e-05	1,5108e-02	5,7007e-05	2,7862e-01	-1,4560e-01	-5,4130e-02
	Najslabši	2,0721e-04	2,1197e-02	9,2743e-05	2,8095e-01	-1,3856e-01	-4,5707e-02
	Poprečje	1,0270e-04	1,5995e-02	5,1449e-05	2,7797e-01	-1,4697e-01	-5,3719e-02
	Std	4,9459e-05	2,9375e-03	1,8915e-05	1,9901e-03	4,8168e-03	2,5260e-03
$5e+5$	Najboljši	3,7128e-05	9,4432e-03	1,0378e-10	1,9567e-01	-1,6428e-01	-5,7671e-02
	Srednji	8,1118e-05	1,1602e-02	1,2057e-08	2,1337e-01	-1,6045e-01	-5,4269e-02
	Najslabši	1,3746e-04	1,9873e-02	2,3965e-07	2,2280e-01	-1,5359e-01	-5,0538e-02
	Poprečje	8,2895e-05	1,2067e-02	3,7420e-08	2,1158e-01	-1,6011e-01	-5,4009e-02
	Std	2,8280e-05	2,4892e-03	6,2481e-08	8,1245e-03	2,6295e-03	2,1130e-03

Tabela 4.11: Vrednosti indikatorja $I_{\bar{H}}$ na testnih funkcijah 8–13 z $M = 5$.

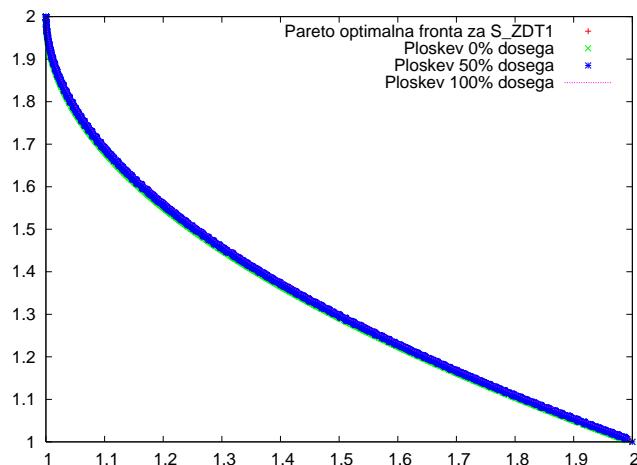
FEs		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
$5e+3$	Najboljši	9,2041e-03	1,2920e-02	6,3531e-03	5,3023e-01	-4,7606e-02	9,5664e-02
	Srednji	1,5070e-02	1,7508e-02	8,7657e-03	5,3136e-01	-2,8500e-02	1,2968e-01
	Najslabši	2,0044e-02	2,6271e-02	1,0835e-02	5,3284e-01	-1,5278e-02	1,5643e-01
	Poprečje	1,5271e-02	1,8495e-02	8,7578e-03	5,3128e-01	-2,9789e-02	1,2825e-01
	Std	2,6715e-03	4,2258e-03	1,1973e-03	7,0675e-04	7,9259e-03	1,4046e-02
$5e+4$	Najboljši	1,8416e-05	9,6066e-03	3,6800e-05	5,2868e-01	-1,5403e-01	-1,0401e-01
	Srednji	2,4435e-05	1,1007e-02	4,8566e-05	5,3036e-01	-1,3930e-01	-9,6581e-02
	Najslabši	7,4098e-05	1,7884e-02	9,0375e-05	5,3308e-01	-1,1671e-01	-8,4310e-02
	Poprečje	2,9494e-05	1,1537e-02	5,1869e-05	5,3046e-01	-1,4020e-01	-9,6181e-02
	Std	1,2713e-05	1,9708e-03	1,1843e-05	9,9277e-04	7,8765e-03	5,2835e-03
$5e+5$	Najboljši	4,4943e-07	7,5310e-03	3,6717e-10	5,1646e-01	-2,1199e-01	-1,2990e-01
	Srednji	1,8934e-06	8,7524e-03	1,6341e-09	5,2159e-01	-1,9635e-01	-1,2384e-01
	Najslabši	6,8299e-06	1,1796e-02	6,9752e-09	5,2409e-01	-1,7477e-01	-1,1667e-01
	Poprečje	2,6355e-06	8,8118e-03	2,0630e-09	5,2126e-01	-1,9555e-01	-1,2346e-01
	Std	1,8277e-06	8,8397e-04	1,6132e-09	2,1147e-03	1,0860e-02	3,3397e-03



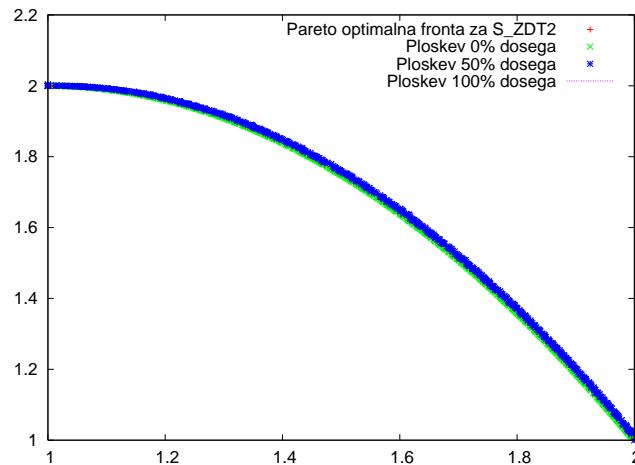
Slika 4.3: Ploskve 0%, 50% in 100% dosega vseh 25 zagonov na testni funkciji OKA2 po 5e+5 ovrednotenjih.



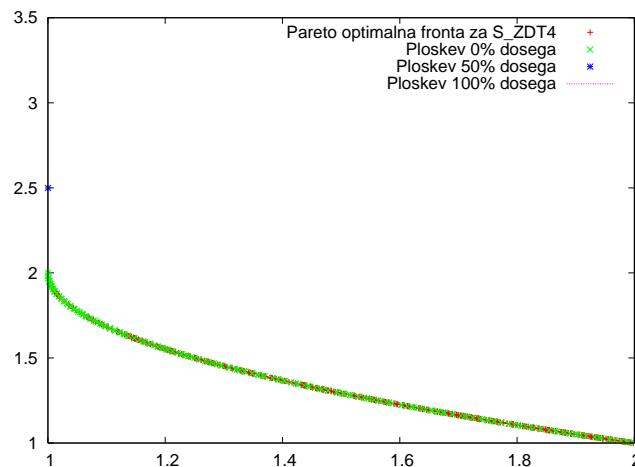
Slika 4.4: Ploskve 0%, 50% in 100% dosega vseh 25 zagonov na testni funkciji SYMPART po 5e+5 ovrednotenjih.



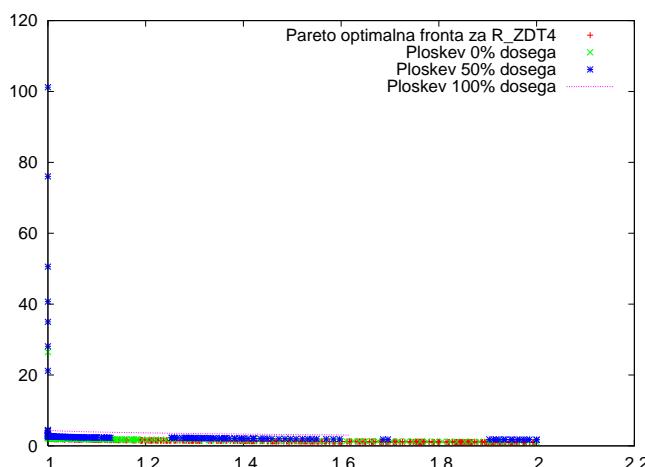
Slika 4.5: Ploskve 0%, 50% in 100% dosega vseh 25 zagonov na testni funkciji S_ZDT1 po 5e+5 ovrednotenjih.



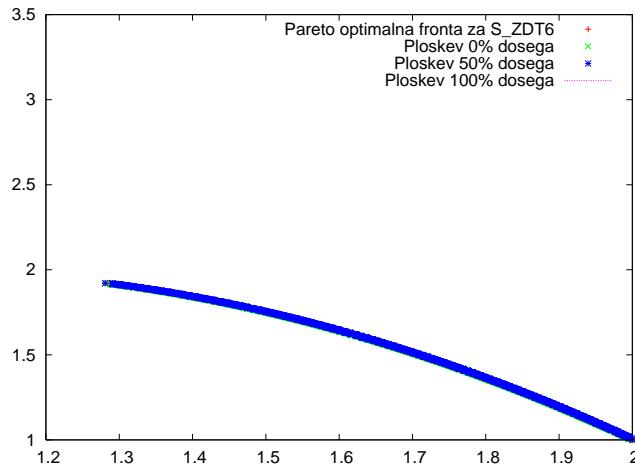
Slika 4.6: Ploskve 0%, 50% in 100% dosega vseh 25 zagonov na testni funkciji S_ZDT2 po 5e+5 ovrednotenjih.



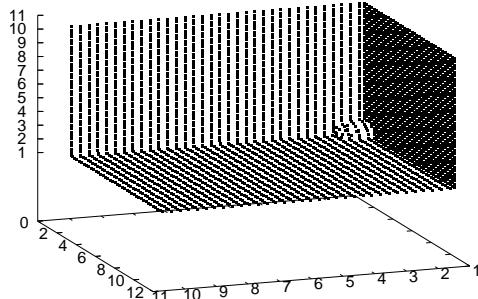
Slika 4.7: Ploskve 0%, 50% in 100% dosega vseh 25 zagonov na testni funkciji S_ZDT4 po 5e+5 ovrednotenjih.



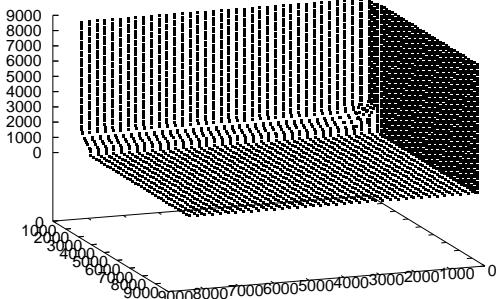
Slika 4.8: Ploskve 0%, 50% in 100% dosega vseh 25 zagonov na testni funkciji R_ZDT4 po 5e+5 ovrednotenjih.



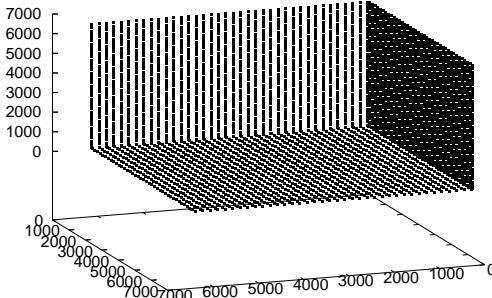
Slika 4.9: Ploskve 0%, 50% in 100% dosega vseh 25 zagonov na testni funkciji S_ZDT6 po $5e+5$ ovrednotenjih.



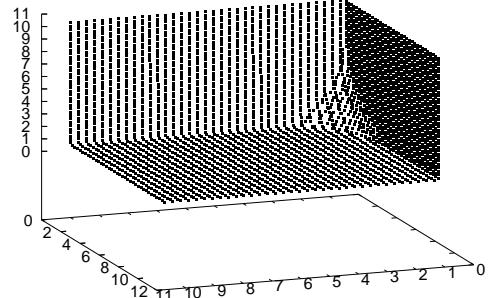
(a) Testna funkcija 8 (S_DTLZ2 z $M = 3$).



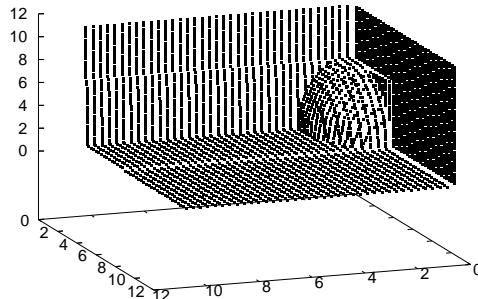
(b) Testna funkcija 9 (R_DTLZ2 z $M = 3$).



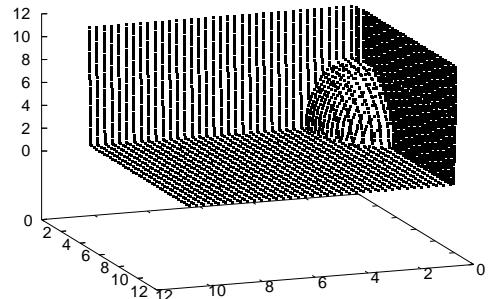
(c) Testna funkcija 10 (S_DTLZ3 z $M = 3$).



(d) Testna funkcija 11 (WFG1 z $M = 3$).

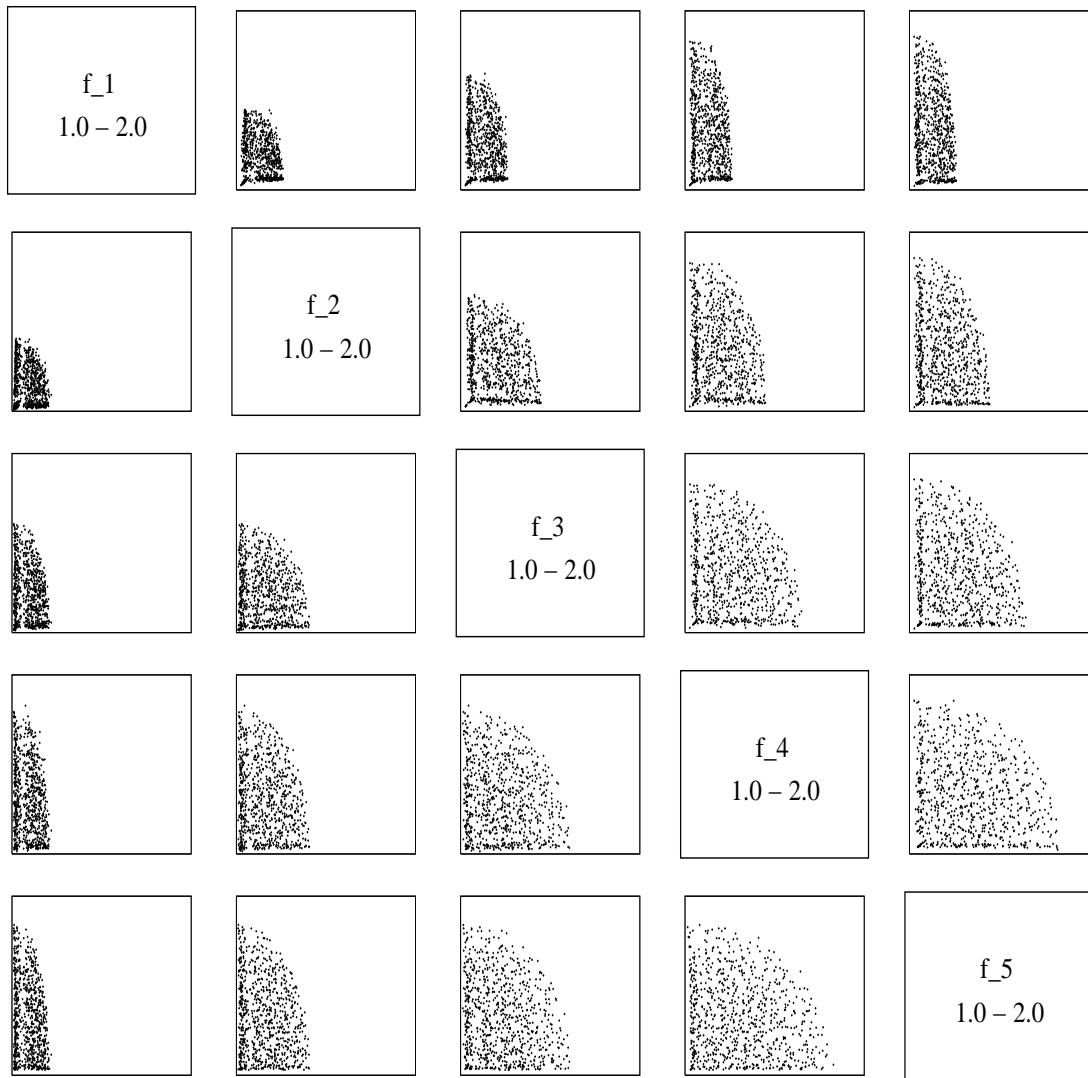


(e) Testna funkcija 12 (WFG8 z $M = 3$).



(f) Testna funkcija 13 (WFG9 z $M = 3$).

Slika 4.10: Ploskve 50% dosega vseh 25 zagonov na testnih funkcijah 8–13 z $M = 3$ po $5e+5$ ovrednotenjih.



Slika 4.11: Po kriterijih projicirane aproksimacijske množice pri zagonu z medialno vrednostjo indikatorja I_R po $5e+5$ ovrednotenjih funkcij 12 in 13.

4.4 Statistična analiza rezultatov algoritmov

V tem razdelku statistično primerjamo algoritom DEMOwSA z algoritmom DEMO (fiksni krmilni parametri) in z algoritmi, ki so bili predstavljeni na posebni sekciiji za oceno učinkovitosti večkriterijskih algoritmov za optimizacijo na IEEE Congress on Evolutionary Computation (CEC 2007). Način ocenjevanja smo opisali v prejšnjem razdelku, sedaj pa podajamo razlike med ocenami, primerjane na t-testu z 99,9% zanesljivostjo. Negativna vrednost je boljša za DEMOwSA. Oznaka \dagger in poudarjeno besedilo označuje statistično signifikantno boljše ocene pri DEMOwSA. Oznaka \ddagger in poševno besedilo označuje nasprotno, slabše ocene.

4.4.1 Algoritem DEMO

Primerjavo algoritmov DEMOwSA in DEMO smo prvič predstavili že na spremenljajoči se prvi različici tehničnega poročila o funkcijskem naboru testnih funkcij CEC 2007 [103]. Rezultati so pokazali, da algoritom DEMOwSA daje večkrat boljše rezultate v primerjavi z algoritmom DEMO. Sedaj podajamo še primerjavo nad končno različico testnih funkcij [2] in ponovno potrjujemo, da algoritom DEMOwSA večkrat daje boljše rezultate od algoritma DEMO.

V tabelah 4.12–4.14 poglejmo najprej končne dobljene aproksimacijske množice po 5e+5 ovrednotenjih. Tukaj vidimo, da je algoritom DEMOwSA na funkcijah SYM-PART, S_ZDT1, S_ZDT4, S_DTLZ2 z $M = 3$, R_DTLZ2 z $M = 3$ ali $M = 5$, S_DTLZ3 z $M = 3$ ali $M = 5$, WFG1 z $M = 3$ ter WFG9 z $M = 3$ ali $M = 5$ signifikantno boljši od algoritma DEMO. Obratno, da je osnovni algoritrom DEMO boljši od algoritma DEMOwSA, za te aproksimacijske množice velja za funkcije OKA2, R_ZDT4, S_ZDT6, WFG8 z $M = 3$ ali $M = 5$, S_DTLZ2 z $M = 5$ ter WFG1 z $M = 5$. Primerjajmo rezultate v tabelah 4.15–4.17. Iz teh vidimo, da algoritom DEMOwSA daje signifikantno boljše rezultate nad funkcijami OKA2, SYM-PART, S_ZDT4, R_DTLZ2 z $M = 3$ ali $M = 5$, D_DTLZ3 z $M = 3$ ali $M = 5$, WFG1 z $M = 3$ in WFG9 z $M = 3$ ali $M = 5$. Pozitivne ocene signifikantnega izboljšanja obeh indikatorjev se skladajo na funkcijah SYM-PART, S_ZDT4, R_DTLZ2 in S_DTLZ3 z $M = 3$ ali $M = 5$, WFG1 z $M = 3$ in WFG9 z $M = 3$ ali $M = 5$. Kot vidimo, indikatorja za aproksimacijske množice na funkciji OKA2 izmenično pokažeta različne signifikantne razlike, na indikatorju I_R v prid za algoritmom DEMO, na indikatorju $I_{\overline{H}}$ za algoritmom DEMOwSA. Po 5e+4 ovrednotenjih je algoritom DEMOwSA dvanajstkrat boljši na obeh indikatorjih I_R oz. $I_{\overline{H}}$ in štiri oz. šestkrat slabši na obeh indikatorjih I_R oz. $I_{\overline{H}}$. Po 5e+3 ovrednotenjih je algoritom DEMOwSA od algoritma DEMO sedem oz. devetkrat boljši na obeh indikatorjih I_R oz. $I_{\overline{H}}$ in sedemkrat slabši na obeh indikatorjih.

Tabela 4.12: Razlika vrednosti indikatorja I_R za algoritma DEMO in DEMOWSA na testnih funkcijah 1–7.

<i>FEs</i>		1. OKA2	2. SYMPART	3. S_ZDT1	4. S_ZDT2	5. S_ZDT4	6. R_ZDT4	7. S_ZDT6
5e+3	Najboljši	9,7136e–04	–3,6300e–03	–1,6876e–02	–2,3033e–02	2,6190e–03	–1,5730e–03	–7,5200e–03
	Srednji	7,2939e–04	–2,5950e–03	–1,6578e–02	–1,6390e–02	–4,0200e–04	6,0730e–03	–3,8300e–03
	Najslabši	7,4200e–04	–3,8940e–03	–1,0627e–02	7,1300e–03	–3,2100e–03	–2,6500e–03	–4,4200e–03
	Poprečje	$7,1056e-04^\dagger$	$-4,5560e-03^\dagger$	$-1,5020e-02^\dagger$	$-1,5790e-02^\dagger$	–6,1500e–04	$3,7620e-03^\dagger$	$-4,4800e-03^\dagger$
	Std	9,7790e–05	–3,7910e–04	1,8844e–03	4,6073e–03	–1,1268e–03	6,6830e–04	9,7250e–04
5e+4	Najboljši	1,8005e–04	–1,3908e–05	–1,9732e–04	–4,6935e–04	–2,2927e–02	–1,6369e–03	–2,7657e–02
	Srednji	1,0084e–04	–1,5078e–05	–2,5009e–04	–4,3243e–04	–2,0052e–02	–1,6089e–03	–1,4693e–02
	Najslabši	–1,3750e–03	–1,9755e–05	–7,1390e–05	–3,0900e–04	–1,8794e–02	4,2390e–04	1,3981e–02
	Poprečje	–3,4700e–06	$-1,4800e-05^\dagger$	$-2,4976e-04^\dagger$	–4,9000e–04	$-2,0099e-02^\dagger$	$-9,0920e-04^\dagger$	$-1,3747e-02^\dagger$
	Std	–3,0817e–04	–2,0043e–06	2,0429e–05	4,1000e–05	4,2150e–04	8,9320e–04	1,3179e–02
5e+5	Najboljši	6,5580e–05	–1,8446e–06	–1,1709e–07	8,8060e–08	–2,4653e–03	–1,6000e–07	0,0000e+00
	Srednji	1,0655e–04	–2,8263e–06	–1,6635e–06	1,1237e–06	–3,5788e–03	2,8520e–04	0,0000e+00
	Najslabši	1,9666e–04	–3,2821e–06	–1,1859e–05	1,3700e–04	–5,1171e–03	3,0600e–04	4,5334e–02
	Poprečje	$7,0130e-05^\dagger$	$-2,7977e-06^\dagger$	$-2,0493e-06^\dagger$	1,1000e–05	$-3,6777e-03^\dagger$	$2,6089e-04^\dagger$	$3,6228e-03^\dagger$
	Std	4,0600e–06	–3,2571e–07	–2,4488e–06	1,2000e–05	–7,0671e–04	6,6930e–05	1,2533e–02

Tabela 4.13: Razlika vrednosti indikatorja I_R za algoritma DEMO in DEMOWSA na testnih funkcijah 8–13 z $M = 3$.

<i>FEs</i>		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Najboljši	–4,8895e–05	–3,9933e–04	8,5000e–06	6,6100e–04	1,7240e–03	–5,0080e–04
	Srednji	–6,2910e–05	–1,4490e–04	1,5510e–05	3,5700e–04	7,9700e–04	–3,0900e–04
	Najslabši	–4,7640e–05	–4,2700e–05	3,0430e–05	1,4400e–04	4,8630e–04	6,4430e–04
	Poprečje	$-4,7270e-05^\dagger$	–1,5230e–04	$2,5330e-05^\dagger$	$3,3800e-04^\dagger$	$1,1490e-03^\dagger$	–1,8320e–04
	Std	8,3300e–06	3,8390e–05	2,8430e–06	–9,6750e–05	–6,9200e–05	–2,7500e–05
5e+4	Najboljši	4,1653e–06	–5,6380e–05	–3,4648e–05	1,0070e–03	4,5800e–04	–2,7410e–04
	Srednji	–2,9390e–06	–4,8563e–04	–3,1762e–05	1,2720e–03	1,5250e–03	–9,1700e–05
	Najslabši	–4,2580e–06	–6,2000e–04	–2,6208e–05	8,9300e–04	1,1190e–03	4,6600e–04
	Poprečje	–4,2600e–07	$-4,6762e-04^\dagger$	$-3,2078e-05^\dagger$	$1,0650e-03^\dagger$	$1,3000e-03^\dagger$	$-1,6810e-04^\dagger$
	Std	–5,1000e–07	–1,0030e–04	1,6328e–06	1,0726e–04	2,8267e–04	9,9190e–05
5e+5	Najboljši	–5,7127e–06	–5,0100e–05	–1,1329e–05	–1,6470e–03	1,0900e–03	–2,5210e–04
	Srednji	–6,5050e–06	–8,3160e–05	–1,4365e–05	–7,8000e–04	1,6320e–03	–2,0320e–04
	Najslabši	–9,4470e–06	–3,4700e–05	–1,3831e–05	–1,5100e–04	2,5580e–03	2,6800e–04
	Poprečje	$-6,5970e-06^\dagger$	$-2,0201e-04^\dagger$	$-1,3896e-05^\dagger$	$-9,4800e-04^\dagger$	$1,6290e-03^\dagger$	$-1,9040e-04^\dagger$
	Std	–1,8679e–06	–7,6410e–05	–7,3020e–07	6,5705e–04	3,8397e–04	2,4200e–06

Tabela 4.14: Razlika vrednosti indikatorja I_R za algoritma DEMO in DEMOWSA na testnih funkcijah 8–13 z $M = 5$.

<i>FEs</i>		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Najboljši	–4,5070e–05	2,5220e–05	7,4700e–06	1,4400e–04	–6,4640e–04	–1,1191e–03
	Srednji	6,3300e–06	–1,7378e–04	2,6200e–06	2,4000e–05	7,5140e–04	–1,0900e–03
	Najslabši	4,5460e–05	–4,9200e–05	–3,7750e–05	5,3000e–05	5,8300e–04	–1,9920e–03
	Poprečje	1,8680e–05	$-1,0342e-04^\dagger$	1,5800e–06	$2,4000e-05^\dagger$	$4,4180e-04^\dagger$	$-9,0200e-04^\dagger$
	Std	1,2770e–05	4,4970e–05	–9,5640e–06	–3,4330e–06	–1,5730e–04	–2,8430e–04
5e+4	Najboljši	–2,5540e–06	–1,1814e–05	1,6370e–06	4,3700e–04	1,2130e–03	2,0200e–05
	Srednji	–3,7340e–06	–1,8671e–04	–1,3410e–06	1,6000e–04	1,5553e–03	–1,9920e–04
	Najslabši	–5,6670e–06	8,6000e–06	9,5400e–07	1,3300e–04	1,4089e–03	–7,1500e–05
	Poprečje	$-4,5510e-06^\dagger$	$-1,0428e-04^\dagger$	$-1,0140e-06^\dagger$	$2,1000e-04^\dagger$	$1,7641e-03^\dagger$	$-1,4160e-04^\dagger$
	Std	–1,0586e–06	–2,4830e–05	–4,0450e–07	–6,5240e–05	–1,9470e–04	–9,1050e–05
5e+5	Najboljši	3,6725e–06	–9,8610e–06	–3,6507e–06	–1,2100e–04	3,3368e–03	–4,8700e–04
	Srednji	2,7008e–06	–9,5160e–06	–4,7585e–06	6,0000e–05	4,7597e–03	–1,1230e–04
	Najslabši	6,5750e–06	–1,3314e–04	–5,7430e–06	9,2000e–05	4,6783e–03	1,8500e–05
	Poprečje	$2,8165e-06^\dagger$	$-2,0536e-05^\dagger$	$-4,8919e-06^\dagger$	$6,0000e-05^\dagger$	$4,3552e-03^\dagger$	$-1,7190e-04^\dagger$
	Std	1,6250e–07	–2,5105e–05	–4,4258e–07	6,5200e–05	3,5106e–04	4,3690e–05

Tabela 4.15: Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma DEMO in DEMOWSA na testnih funkcijah 1–7.

<i>FEs</i>		1. OKA2	2. SYMPART	3. S_ZDT1	4. S_ZDT2	5. S_ZDT4	6. R_ZDT4	7. S_ZDT6
5e+3	Najboljši	5,2230e-03	-1,0283e-02	-6,7220e-02	-5,7880e-02	5,7900e-03	-6,3710e-03	-1,9320e-02
	Srednji	2,0790e-03	-7,1970e-03	-5,4650e-02	-5,9300e-02	-8,0000e-05	1,8168e-02	-1,1700e-02
	Najslabši	1,3110e-03	-1,0600e-02	-3,9530e-02	-4,7130e-02	-7,1300e-03	-1,1670e-02	-1,2010e-02
	Poprečje	<i>1,2120e-03</i> [‡]	-1,2626e-02 [†]	-5,5010e-02 [†]	-5,7200e-02 [†]	-2,6500e-03	<i>1,2210e-02</i> [‡]	-1,3050e-02 [†]
	Std	8,2100e-05	-9,0100e-04	5,8780e-03	4,6860e-03	-2,3860e-03	7,5600e-04	2,7027e-03
5e+4	Najboljši	-5,5800e-03	-4,0578e-05	-1,0310e-03	-1,2624e-03	-6,7283e-02	-4,4775e-03	-6,4610e-02
	Srednji	-5,1870e-03	-4,5375e-05	-1,1574e-03	-1,2210e-03	-5,9173e-02	-4,6091e-03	-4,0820e-02
	Najslabši	1,1090e-03	-5,8768e-05	-4,0220e-04	-9,3000e-04	-5,5809e-02	1,3140e-03	3,1593e-02
	Poprečje	-4,8290e-03 [†]	-4,4153e-05 [†]	-1,1410e-03 [†]	-1,3450e-03	-5,9319e-02 [†]	-2,5885e-03 [†]	-3,2890e-02 [†]
	Std	1,8859e-03	-6,2103e-06	8,2080e-05	5,0000e-06	1,0628e-03	2,6823e-03	3,0316e-02
5e+5	Najboljši	-7,7780e-03	-6,2203e-06	5,7500e-06	7,0200e-06	-7,7217e-03	-3,7000e-07	5,0000e-07
	Srednji	-7,5934e-03	-8,5240e-06	1,2100e-06	2,1980e-05	-1,0375e-02	1,0988e-03	9,1000e-07
	Najslabši	-4,0000e-05	-9,8585e-06	-2,0100e-05	4,1400e-04	-1,4846e-02	1,1069e-03	1,0217e-01
	Poprečje	-7,1461e-03 [†]	-8,3930e-06 [†]	7,0000e-07	4,1000e-05	-1,0811e-02 [†]	<i>8,7797e-04</i> [‡]	<i>8,1637e-03</i> [‡]
	Std	2,1342e-03	-9,4861e-07	-5,8122e-06	3,0000e-05	-1,8477e-03	2,2503e-04	2,8240e-02

Tabela 4.16: Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma DEMO in DEMOWSA na testnih funkcijah 8–13 z $M = 3$.

<i>FEs</i>		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Najboljši	-1,0948e-03	-2,4600e-03	1,9380e-04	3,0100e-03	5,5980e-03	1,1200e-03
	Srednji	-9,2140e-04	2,3900e-04	2,0097e-03	1,7800e-03	5,1990e-03	3,7500e-04
	Najslabši	-1,1188e-03	-9,5000e-04	1,8500e-03	7,6000e-04	4,2260e-03	-3,6840e-03
	Poprečje	-8,3660e-04 [†]	-6,9800e-04	<i>1,4644e-03</i> [‡]	<i>1,6300e-03</i> [‡]	<i>5,7900e-03</i> [‡]	-8,8600e-04 [†]
	Std	1,1576e-04	-1,7360e-04	2,7720e-04	-4,8238e-04	5,0400e-05	-1,8038e-03
5e+4	Najboljši	2,0546e-05	-6,2300e-04	-2,0402e-04	4,6400e-03	2,5400e-03	-1,6850e-03
	Srednji	2,3009e-05	-3,0000e-03	-2,5082e-04	5,7500e-03	9,2400e-03	-1,2520e-03
	Najslabši	-1,2280e-05	-4,8470e-03	-2,3905e-04	4,3600e-03	6,6600e-03	1,7360e-03
	Poprečje	<i>1,8288e-05</i> [‡]	-2,6510e-03 [†]	-2,4131e-04 [†]	<i>5,1300e-03</i> [‡]	<i>7,1100e-03</i> [‡]	-1,0480e-03 [†]
	Std	-1,9160e-06	-5,5780e-04	-1,6688e-05	5,2040e-04	1,4091e-03	1,5350e-04
5e+5	Najboljši	2,0898e-05	-4,1840e-04	-8,7974e-07	-7,9000e-03	7,0700e-03	-1,1700e-04
	Srednji	4,7910e-06	-4,9100e-04	-2,1858e-06	-3,7900e-03	9,1600e-03	-1,3640e-03
	Najslabši	-1,3539e-04	-7,5500e-04	-3,5608e-06	-7,9000e-04	1,4200e-02	-1,0800e-04
	Poprečje	-3,4440e-06	-1,3460e-03 [†]	-2,1885e-06 [†]	-4,7500e-03 [†]	<i>9,7100e-03</i> [‡]	-6,3600e-04 [†]
	Std	-2,9680e-05	-3,4260e-04	-8,7009e-07	3,3184e-03	1,6821e-03	6,5700e-05

Tabela 4.17: Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma DEMO in DEMOWSA na testnih funkcijah 8–13 z $M = 5$.

<i>FEs</i>		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Najboljši	-1,5079e-03	4,9000e-05	-3,3510e-04	1,8200e-03	3,0410e-03	-4,8260e-03
	Srednji	4,8700e-04	-2,5020e-03	-2,9920e-04	3,2000e-04	-3,9300e-03	-3,8000e-03
	Najslabši	1,3180e-03	-1,3340e-03	-1,0670e-03	3,0000e-04	-5,1840e-03	-1,9210e-02
	Poprečje	<i>6,6300e-04</i> [‡]	-1,9140e-03 [†]	-1,2600e-04	<i>3,2000e-04</i> [‡]	-3,0610e-03 [†]	-6,5600e-03 [†]
	Std	4,2450e-04	5,0300e-04	-1,6250e-04	-1,5939e-04	-3,2911e-03	-5,8300e-03
5e+4	Najboljši	-1,1185e-05	-8,1340e-04	5,6920e-06	4,0000e-03	2,9780e-02	5,1600e-03
	Srednji	-2,1672e-05	-2,8230e-03	1,0800e-06	1,7500e-03	2,9320e-02	-2,1110e-03
	Najslabši	1,5890e-06	-1,9900e-04	2,0653e-05	2,1100e-03	3,0510e-02	-7,3670e-03
	Poprečje	-1,8171e-05 [†]	-2,0630e-03 [†]	<i>2,1340e-06</i> [‡]	<i>2,0400e-03</i> [‡]	<i>2,7800e-02</i> [‡]	-2,6200e-03 [†]
	Std	1,7870e-06	-2,6100e-04	2,1929e-06	-7,1893e-04	-5,3590e-04	-3,1014e-03
5e+5	Najboljši	1,3879e-07	-8,7320e-04	-2,4870e-07	-1,7800e-03	6,1030e-02	-4,3300e-03
	Srednji	-4,0960e-07	-8,3460e-04	-6,6175e-07	2,6000e-04	5,9480e-02	-3,2400e-03
	Najslabši	2,5240e-06	-1,2940e-03	-1,4143e-06	-2,2000e-04	5,3760e-02	-9,6000e-04
	Poprečje	<i>4,4520e-07</i> [‡]	-7,1650e-04 [†]	-6,7060e-07 [†]	2,7000e-04	<i>5,9320e-02</i> [‡]	-2,8500e-03 [†]
	Std	8,4167e-07	-7,2570e-05	-2,8304e-07	4,0530e-04	2,5929e-03	1,0227e-03

4.4.2 Algoritem MOSaDE

MOSaDE [104] Huangove in sodelavcev je nadgrajena različica algoritma SaDE [105], ki uporablja diferencialno evolucijo. Dodana je metrika nakopičenosti [79], ki omogoča večkriterijsko selekcijo za ohranjanje velikosti arhiva in aproksimacijske množice. Uporablja $NP = 50$ in dve mutacijski strategiji, ki ju enakomerno porazdeljeno naključno izbira. Algoritem uporablja tudi princip lokalnega iskanja, tj. metodo kvazi-Newtonovega lokalnega iskanja.

V tabelah 4.18–4.23 vidimo, da algoritem DEMOwSA daje boljše rezultate pri vseh treh vzorčenjih aproksimacijske množice algoritem DEMOwSA daje boljše rezultate za oba indikatorja pri funkcijah OKA2, S_DTLZ3 z $M = 3$ ali $M = 5$, WFG8 z $M = 3$ ali $M = 5$, WFG9 z $M = 3$ ali $M = 5$, S_DTLZ2 z $M = 5$ in R_DTLZ2 z $M = 5$.

Pri funkcijah S_ZDT4, R_ZDT4, S_DTLZ2 z $M = 3$, R_DTLZ2 z $M = 3$ in WFG1 z $M = 3$ ali $M = 5$ oba indikatorja algoritem DEMOwSA ocenita slabše. Kot vidimo, je takih funkcij s hkrati boljšima ocenama indikatorjev več od teh, na katerih oba indikatorja za algoritem DEMOwSA dajeta slabše ocene.

Pri funkcijah S_ZDT1 in S_ZDT2 indikator I_R bolje oceni algoritem MOSaDE, a indikator $I_{\overline{H}}$ za $5e+4$ ovrednotenjih in $5e+5$ ovrednotenjih pokaže nasprotno. Tudi pri funkciji SYM-PART je sprva boljši primerjan algoritem MOSaDE, nato po $5e+4$ ovrednotenjih in $5e+5$ ovrednotenjih algoritem DEMOwSA doseže boljše rezultate.

Med funkcijami z $M = 3$ ali $M = 5$ opazimo znatnejšo razliko med dobljenimi rezultati, kar bi lahko pripisali kakovostnejši večkriterijski selekciji SPEA-II napram sicer računsko hitrejši NSGA-II.

Algoritem DEMOwSA ne uporablja lokalnega iskanja. Opaziti je, da kljub temu, da MOSaDE uporablja lokalno iskanje, algoritem DEMOwSA v povprečju daje boljše rezultate od algoritma MOSaDE.

Tabela 4.18: Razlika vrednosti indikatorja I_R za algoritma MOSaDE in DEMOwSA na testnih funkcijah 1–7.

<i>FEs</i>		1. OKA2	2. SYMPART	3. S_ZDT1	4. S_ZDT2	5. S_ZDT4	6. R_ZDT4	7. S_ZDT6
5e+3	Najboljši	-1,2186e-01	1,6297e-02	1,2564e-02	4,5328e-02	3,4439e-02	3,5480e-03	2,1010e-02
	Srednji	-1,2163e-01	2,8954e-02	1,9016e-02	5,0376e-02	2,7982e-02	1,1655e-02	2,5210e-02
	Najslabši	-1,4346e-01	4,1629e-02	2,0495e-02	7,1984e-02	2,8432e-02	8,8220e-03	2,3760e-02
	Poprečje	-1,2644e-01†	2,7780e-02‡	1,8754e-02‡	5,0494e-02‡	2,8799e-02‡	8,9040e-03‡	2,4530e-02‡
	Std	-8,0046e-03	5,8483e-03	1,3242e-03	4,7873e-03	-2,3127e-03	1,4722e-03	4,5030e-04
5e+4	Najboljši	-1,2260e-01	-6,4970e-05	5,1977e-03	8,2387e-03	1,0953e-03	1,0376e-03	-1,7576e-02
	Srednji	-1,2280e-01	-1,0155e-04	5,1232e-03	8,1418e-03	5,7710e-03	1,6073e-03	-9,9420e-03
	Najslabši	-1,2285e-01	-3,5967e-04	5,1881e-03	4,7385e-02	1,0383e-02	5,2587e-03	1,3499e-02
	Poprečje	-1,2284e-01†	-1,2733e-04†	5,1289e-03‡	2,5436e-02‡	6,3470e-03‡	2,1046e-03‡	-9,3080e-03†
	Std	-9,4810e-05	-8,2608e-05	-2,6873e-05	1,9904e-02	2,3305e-03	1,2514e-03	9,9644e-03
5e+5	Najboljši	-1,2265e-01	-1,1922e-05	5,2415e-03	8,2831e-03	2,5686e-04	1,2210e-03	7,9505e-06
	Srednji	-1,2269e-01	-1,9579e-05	5,2007e-03	8,2749e-03	4,9284e-04	1,2890e-03	7,9505e-06
	Najslabši	-1,2213e-01	-4,5438e-05	5,0987e-03	4,8356e-02	7,9049e-04	7,6244e-04	4,5359e-02
	Poprečje	-1,2265e-01†	-2,3718e-05†	5,1923e-03‡	2,5899e-02‡	4,2603e-04‡	1,2886e-03‡	3,6314e-03‡
	Std	1,1359e-04	-9,4372e-06	-3,4802e-05	2,0275e-02	1,5922e-04	-6,3680e-05	1,2536e-02

Tabela 4.19: Razlika vrednosti indikatorja I_R za algoritma MOSaDE in DEMOwSA na testnih funkcijah 8–13 z $M = 3$.

<i>FEs</i>		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Najboljši	5,6466e-05	-8,8040e-05	-1,0826e-02	5,5864e-02	-1,3943e-02	-2,3325e-02
	Srednji	6,2133e-05	1,2790e-03	-1,6343e-02	5,6130e-02	-1,1819e-02	-2,8426e-02
	Najslabši	3,9300e-06	3,3762e-03	-2,3324e-02	5,6119e-02	-8,5505e-03	-3,2570e-02
	Poprečje	5,8750e-05‡	1,3040e-03‡	-1,7122e-02†	5,6073e-02‡	-1,1427e-02†	-2,8052e-02†
	Std	-3,7800e-06	8,3350e-04	-3,6471e-03	7,6583e-05	1,3165e-03	-2,8054e-03
5e+4	Najboljši	8,55665e-05	1,2742e-04	8,7128e-04	5,3486e-02	-2,4956e-02	-1,8159e-02
	Srednji	7,3746e-05	6,3221e-04	-4,9718e-03	5,4597e-02	-2,2878e-02	-2,7406e-02
	Najslabši	7,5658e-05	1,7949e-03	-7,7610e-03	5,5005e-02	-2,0750e-02	-3,3736e-02
	Poprečje	7,6164e-05‡	7,2957e-04‡	-4,2574e-03†	5,4425e-02‡	-2,2866e-02†	-2,6514e-02†
	Std	-1,6530e-06	5,0287e-04	-2,5265e-03	4,0317e-04	1,0866e-03	-3,5116e-03
5e+5	Najboljši	8,9209e-05	1,9477e-04	1,7982e-03	3,7740e-02	-2,6675e-02	-1,0730e-02
	Srednji	1,0482e-04	2,7180e-04	2,2611e-04	4,1303e-02	-2,5835e-02	-1,6335e-02
	Najslabši	1,1804e-04	1,8581e-03	-1,3224e-03	4,3174e-02	-2,4277e-02	-3,1253e-02
	Poprečje	1,0456e-04‡	4,4380e-04‡	2,6373e-04‡	4,0928e-02‡	-2,5827e-02†	-1,7338e-02†
	Std	7,6297e-06	3,9083e-04	-9,1342e-04	1,6107e-03	5,9963e-04	-5,5104e-03

Tabela 4.20: Razlika vrednosti indikatorja I_R za algoritma MOSaDE in DEMOwSA na testnih funkcijah 8–13 z $M = 5$.

<i>FEs</i>		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Najboljši	-3,9076e-04	-8,3200e-02	-1,0330e-02	4,6789e-02	-5,1354e-02	-1,3320e-03
	Srednji	-3,9152e-04	-8,5747e-02	-1,5191e-02	4,6968e-02	-4,9426e-02	-8,0800e-04
	Najslabši	-2,1130e-04	-8,6892e-02	-1,9116e-02	4,7062e-02	-4,8129e-02	1,3200e-03
	Poprečje	-3,5091e-04†	-8,5679e-02†	-1,4642e-02†	4,6953e-02‡	-4,9469e-02†	-6,2200e-04†
	Std	6,1748e-05	-9,4106e-04	-2,8857e-03	6,8381e-05	3,8260e-04	5,1600e-04
5e+4	Najboljši	-3,5060e-04	-7,2349e-02	-6,1573e-03	4,6773e-02	-4,7737e-02	-4,1269e-03
	Srednji	-3,9249e-04	-7,5816e-02	-8,0602e-03	4,6908e-02	-5,0109e-02	-4,7951e-03
	Najslabši	-4,3948e-04	-7,6525e-02	-1,1908e-02	4,7105e-02	-5,0039e-02	-5,3766e-03
	Poprečje	-3,9513e-04†	-7,5161e-02†	-8,6582e-03†	4,6925e-02‡	-4,9842e-02†	-4,9737e-03†
	Std	-2,4695e-05	-1,2582e-03	-1,8252e-03	8,8337e-05	-6,8690e-04	-4,3190e-04
5e+5	Najboljši	-9,9984e-05	-2,5978e-02	-8,8492e-04	4,5456e-02	-1,3267e-02	-4,4015e-03
	Srednji	-1,7950e-04	-2,8163e-02	-3,1332e-03	4,5926e-02	-1,7362e-02	-4,7839e-03
	Najslabši	-2,3285e-04	-3,2986e-02	-7,4134e-03	4,6239e-02	-4,1414e-02	-6,0405e-03
	Poprečje	-1,8248e-04†	-2,8594e-02†	-3,6158e-03†	4,5921e-02‡	-1,9018e-02†	-4,8729e-03†
	Std	-3,1843e-05	-1,8510e-03	-1,8333e-03	2,1402e-04	-5,6997e-03	-3,1507e-04

Tabela 4.21: Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma MOSaDE in DEMOwSA na testnih funkcijah 1–7.

<i>FEs</i>		1. OKA2	2. SYMPART	3. S_ZDT1	4. S_ZDT2	5. S_ZDT4	6. R_ZDT4	7. S_ZDT6
5e+3	Najboljši	-3,8146e-01	4,6589e-02	1,2170e-02	3,5440e-02	1,0000e-01	6,3680e-03	5,2890e-02
	Srednji	-3,8038e-01	8,1961e-02	1,7990e-02	5,0040e-02	8,1980e-02	3,0717e-02	5,6620e-02
	Najslabši	-4,2352e-01	1,1607e-01	2,3050e-02	6,2040e-02	9,0000e-02	1,8803e-02	5,0210e-02
	Poprečje	-3,8918e-01[†]	7,8556e-02[‡]	1,9100e-02[‡]	4,9600e-02[‡]	8,5560e-02[‡]	2,4131e-02[‡]	5,4850e-02[‡]
	Std	-1,3601e-02	1,6087e-02	4,2300e-03	9,2750e-03	-5,1330e-03	2,3540e-03	7,4680e-04
5e+4	Najboljši	-3,9057e-01	-1,6184e-04	-1,8859e-02	-4,7063e-02	3,8042e-03	3,8530e-03	-4,9769e-02
	Srednji	-3,8969e-01	-2,6608e-04	-1,9016e-02	-4,6551e-02	1,6493e-02	4,2005e-03	-3,8270e-02
	Najslabši	-3,8321e-01	-1,0160e-03	-1,8342e-02	6,5400e-04	3,0374e-02	1,5408e-02	1,9860e-02
	Poprečje	-3,8900e-01[†]	-3,4370e-04[†]	-1,8949e-02[†]	-2,6265e-02[†]	1,8046e-02[‡]	6,1494e-03[‡]	-3,2310e-02[†]
	Std	1,2377e-03	-2,3909e-04	6,4250e-05	2,3817e-02	6,7782e-03	3,6382e-03	2,2581e-02
5e+5	Najboljši	-3,9529e-01	-1,0296e-05	-1,8220e-02	-4,6424e-02	1,3334e-03	4,2234e-03	3,5439e-03
	Srednji	-3,9356e-01	-3,2428e-05	-1,8351e-02	-4,6413e-02	2,6437e-03	4,3792e-03	3,5443e-03
	Najslabši	-3,7754e-01	-1,0709e-04	-1,8621e-02	1,5720e-03	2,7909e-03	1,3381e-03	1,0575e-01
	Poprečje	-3,9186e-01[†]	-4,3185e-05[†]	-1,8363e-02[†]	-2,5443e-02[†]	2,3061e-03[‡]	4,1304e-03[‡]	1,1708e-02[‡]
	Std	5,3717e-03	-2,7239e-05	-9,2746e-05	2,4149e-02	4,3795e-04	-4,9742e-04	2,8246e-02

Tabela 4.22: Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma MOSaDE in DEMOwSA na testnih funkcijah 8–13 z $M = 3$.

<i>FEs</i>		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Najboljši	3,5820e-04	4,3570e-03	-1,2468e-01	2,8586e-01	-1,0129e-01	-4,0740e-01
	Srednji	3,9520e-04	4,6040e-03	-1,5657e-01	2,8640e-01	-9,1280e-02	-4,7252e-01
	Najslabši	5,0250e-04	1,3652e-02	-1,8680e-01	2,8262e-01	-8,1789e-02	-5,1370e-01
	Poprečje	5,1860e-04[‡]	5,3690e-03[‡]	-1,5673e-01[†]	2,8589e-01[‡]	-9,1077e-02[†]	-4,7459e-01[†]
	Std	9,7790e-05	2,0023e-03	-1,6988e-02	-5,3578e-04	4,4542e-03	-2,6349e-02
5e+4	Najboljši	4,8319e-05	1,1194e-02	-4,2350e-02	2,7388e-01	-1,5769e-01	-2,6011e-01
	Srednji	4,9896e-05	1,4973e-02	-6,3980e-02	2,7865e-01	-1,4566e-01	-3,7325e-01
	Najslabši	-6,1100e-06	2,0822e-02	-7,4407e-02	2,8088e-01	-1,3925e-01	-4,6342e-01
	Poprečje	4,7463e-05[‡]	1,5829e-02[‡]	-6,2672e-02[†]	2,7816e-01[‡]	-1,4710e-01[†]	-3,6776e-01[†]
	Std	-6,5930e-06	2,8426e-03	-8,6617e-03	1,7455e-03	4,6453e-03	-5,0380e-02
5e+5	Najboljši	6,1384e-05	9,4434e-03	-3,0677e-02	1,9624e-01	-1,6428e-01	-1,0483e-01
	Srednji	1,0277e-04	1,1602e-02	-3,8616e-02	2,1341e-01	-1,6045e-01	-1,8524e-01
	Najslabši	1,5229e-04	1,9873e-02	-4,3305e-02	2,2278e-01	-1,5359e-01	-4,2810e-01
	Poprečje	1,0385e-04[‡]	1,2067e-02[‡]	-3,7447e-02[†]	2,1184e-01[‡]	-1,6011e-01[†]	-2,1237e-01[†]
	Std	2,5873e-05	2,4892e-03	-3,3993e-03	7,8477e-03	2,6283e-03	-8,9164e-02

Tabela 4.23: Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma MOSaDE in DEMOwSA na testnih funkcijah 8–13 z $M = 5$.

<i>FEs</i>		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Najboljši	-3,5733e-02	-4,1718e-01	-1,0104e-01	5,2349e-01	-7,0238e-01	7,0811e-02
	Srednji	-3,9847e-02	-4,2798e-01	-1,3874e-01	5,2260e-01	-7,0214e-01	-1,9706e-01
	Najslabši	-4,6778e-02	-4,2806e-01	-1,5865e-01	5,2010e-01	-7,0434e-01	-2,4451e-01
	Poprečje	-3,9726e-02[†]	-4,2638e-01[†]	-1,3495e-01[†]	5,2231e-01[‡]	-7,0316e-01[†]	-6,7470e-02[†]
	Std	-2,0028e-03	-2,3327e-03	-1,7598e-02	-6,9845e-04	-7,6730e-04	-1,5168e-01
5e+4	Najboljši	-4,3707e-03	-3,7470e-01	-6,9520e-02	5,2776e-01	-7,4304e-01	-1,2473e-01
	Srednji	-8,4137e-03	-3,9001e-01	-8,7391e-02	5,2922e-01	-7,7713e-01	-2,7405e-01
	Najslabši	-1,3359e-02	-3,9030e-01	-1,1166e-01	5,3139e-01	-7,7654e-01	-3,5962e-01
	Poprečje	-8,8842e-03[†]	-3,8670e-01[†]	-8,9284e-02[†]	5,2928e-01[‡]	-7,7632e-01[†]	-2,2791e-01[†]
	Std	-2,7232e-03	-5,0044e-03	-1,0320e-02	7,9566e-04	-7,9705e-03	-1,0316e-01
5e+5	Najboljši	-5,1087e-04	-1,4590e-01	-3,5270e-02	5,1625e-01	-3,8737e-01	-1,5062e-01
	Srednji	-1,5005e-03	-1,5722e-01	-4,7204e-02	5,2117e-01	-4,5233e-01	-2,7106e-01
	Najslabši	-2,6200e-03	-1,8242e-01	-7,2178e-02	5,2347e-01	-7,1091e-01	-3,4806e-01
	Poprečje	-1,5233e-03[†]	-1,5945e-01[†]	-5,0744e-02[†]	5,2084e-01[‡]	-4,6773e-01[†]	-2,2971e-01[†]
	Std	-5,9771e-04	-9,4170e-03	-1,1031e-02	2,0157e-03	-6,4989e-02	-8,0882e-02

4.4.3 Algoritem GDE3

GDE3 [106] Kukkonena in Lampinena je algoritem diferencialne evolucije, ki uporablja fiksne krmilne parametre. Preiskovanje temelji na diferencialni evoluciji [55, 65]. Ima posebej ročno nastavljene krmilne parametre za vsako testirano funkcijo, čeprav testno ogrodje CEC 2007 priporoča enako nastavitev krmilnih parametrov algoritma pri vseh testnih funkcijah. Večje število generacij (tudi do 200) prispeva nedominirane rešitve v arhiv za končni izračun aproksimacije Pareto fronte. Uporabljena je večkriterijska selekcija za ohranjanje raznolikosti, ki se dobro obnese na več kriterijih $M > 2$ [107].

V tabelah 4.24–4.26 vidimo, da je algoritem DEMOwSA na testnih funkcijah WFG8 z $M = 3$ ali $M = 5$, WFG9 z $M = 3$, S_DTLZ2 in R_DTLZ2 vsakokrat boljši, na vseh zajetih aproksimacijskih množicah.

Za indikator I_R so rezultati slabši predvsem na funkcijah z manj parametri, OKA2, S_ZDT2, S_ZDT4 S_DTLZ3 z $M = 3$ ali $M = 5$ in WFG1 z $M = 3$ ali $M = 5$.

Pri funkcijah SYM-PART, S_ZDT1 in R_ZDT4 so rezultati na indikatorju I_R različni, pri funkcijah SYM-PART in R_ZDT4 boljši pri 5e+4 ovrednotenjih in pri S_ZDT1 boljši tudi pri 5e+5 ovrednotenjih.

V tabelah 4.27–4.29 lahko opazimo primerljive ocene, razlika je pri funkcijah S_DTLZ2 ob 5e+3 ovrednotenjih, WFG9 z $M = 3$ ob 5e+3 ovrednotenjih in 5e+4 ovrednotenjih ter tudi pri funkcijah R_DTLZ2 in WFG9 z $M = 5$ pri 5e+5 ovrednotenjih. Pri slednjih dveh je ocena boljša, pri prvih treh slabša.

Rezultati nakazujejo, da je algoritem GDE3 na danem testnem ogrodju in uporabljenou nastavljivo krmilnih parametrov v povprečju boljši od algoritma DEMOwSA. Razlike gre pripisati ročni prilagoditvi krmilnih parametrov algoritmu GDE3, za vsako funkcijo posebej, in uporabljeni večkriterijski selekciji, ki uporablja večji arhiv nedominiranih rešitev.

Tabela 4.24: Razlika vrednosti indikatorja I_R za algoritma GDE3 in DEMOwSA na testnih funkcijah 1–7.

FE_s		1. OKA2	2. SYMPART	3. S_ZDT1	4. S_ZDT2	5. S_ZDT4	6. R_ZDT4	7. S_ZDT6
$5e+3$	Najboljši	8,2333e-04	-1,5260e-03	5,3600e-04	5,1260e-03	1,8164e-02	9,0511e-03	4,7000e-04
	Srednji	1,7557e-03	5,6590e-03	-2,1680e-03	7,2140e-03	1,2102e-02	1,5045e-02	5,2700e-03
	Najslabši	-4,6518e-03	1,2949e-02	3,5820e-03	2,9070e-02	1,1918e-02	1,2894e-02	4,3300e-03
	Poprečje	$1,5321e-03^\ddagger$	$3,9260e-03^\ddagger$	-7,7500e-04	$7,8190e-03^\ddagger$	$1,1871e-02^\ddagger$	$1,2529e-02^\ddagger$	$4,1600e-03^\ddagger$
	Std	-8,4990e-04	2,6574e-03	3,6380e-04	3,5585e-03	-6,9710e-04	2,0009e-03	1,1350e-04
$5e+4$	Najboljši	1,7575e-04	-7,5526e-06	-4,3639e-05	-8,7123e-05	-1,2129e-03	3,5498e-04	-1,5384e-02
	Srednji	3,9250e-04	-1,0862e-05	-7,2055e-05	3,0002e-04	4,3866e-03	1,5414e-03	-3,0000e-06
	Najslabši	1,9687e-03	-3,1621e-05	1,6061e-04	3,2800e-04	8,9361e-03	5,4637e-03	3,2502e-02
	Poprečje	$4,9022e-04^\ddagger$	$-1,1944e-05^\dagger$	$-6,6228e-05^\dagger$	$1,4390e-02^\ddagger$	$4,7053e-03^\ddagger$	$2,1348e-03^\ddagger$	1,0840e-03
	Std	3,8569e-04	-4,7964e-06	3,1079e-05	9,2090e-03	2,5029e-03	1,5784e-03	1,4814e-02
$5e+5$	Najboljši	6,5480e-05	5,9400e-08	-2,6098e-08	-3,3312e-07	4,8950e-10	-5,7100e-05	5,7200e-08
	Srednji	1,4925e-04	-1,9900e-08	-3,2292e-06	-6,4194e-06	3,0912e-04	-5,4710e-05	5,7200e-08
	Najslabši	9,7502e-04	1,6230e-07	-9,6146e-06	1,3700e-04	6,1710e-04	-4,8630e-04	4,5351e-02
	Poprečje	$2,1650e-04^\ddagger$	3,4300e-08	$-3,5954e-06^\dagger$	$1,4421e-02^\ddagger$	$2,5683e-04^\ddagger$	$-9,7150e-05^\dagger$	$3,6235e-03^\ddagger$
	Std	2,1561e-04	-1,1280e-08	-2,4169e-06	9,2160e-03	1,5428e-04	-1,0094e-04	1,2536e-02

Tabela 4.25: Razlika vrednosti indikatorja I_R za algoritma GDE3 in DEMOwSA na testnih funkcijah 8–13 z $M = 3$.

FE_s		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
$5e+3$	Najboljši	-3,3785e-05	-1,9290e-05	1,2703e-02	5,5878e-02	-1,3927e-02	-1,1260e-02
	Srednji	-4,9950e-05	1,3692e-03	1,0701e-02	5,6097e-02	-1,1770e-02	-1,3202e-02
	Najslabši	-8,9000e-05	3,5760e-03	8,1643e-03	5,6195e-02	-8,4939e-03	-1,2308e-02
	Poprečje	$-4,7940e-05^\dagger$	$1,4195e-03^\ddagger$	$1,0363e-02^\ddagger$	$5,6065e-02^\ddagger$	$-1,1384e-02^\dagger$	$-1,2665e-02^\dagger$
	Std	-7,7080e-06	8,5450e-04	-1,3690e-03	8,3053e-05	1,3251e-03	-5,9880e-04
$5e+4$	Najboljši	-1,8975e-05	1,8482e-04	2,7354e-02	5,3479e-02	-2,4948e-02	-2,0760e-04
	Srednji	-2,8624e-05	7,2170e-04	2,5978e-02	5,4599e-02	-2,2857e-02	-2,8070e-04
	Najslabši	-4,6593e-05	1,9457e-03	2,5022e-02	5,5020e-02	-2,0676e-02	-2,0240e-04
	Poprečje	$-2,9228e-05^\dagger$	$8,1893e-04^\ddagger$	$2,6003e-02^\ddagger$	$5,4427e-02^\ddagger$	$-2,2837e-02^\dagger$	$-3,2060e-04^\dagger$
	Std	-7,9490e-06	5,2490e-04	-6,1822e-04	4,0853e-04	1,1049e-03	-1,7030e-04
$5e+5$	Najboljši	1,4143e-06	1,9488e-04	2,8890e-02	3,7734e-02	-2,6671e-02	3,0028e-03
	Srednji	1,4618e-05	2,7208e-04	2,8533e-02	4,1296e-02	-2,5825e-02	3,6376e-03
	Najslabši	1,8093e-05	1,8582e-03	2,8002e-02	4,3167e-02	-2,4264e-02	3,8832e-03
	Poprečje	$1,3790e-05^\ddagger$	$4,4393e-04^\ddagger$	$2,8506e-02^\ddagger$	$4,0921e-02^\ddagger$	$-2,5818e-02^\dagger$	$3,4930e-03^\ddagger$
	Std	5,0120e-06	3,9084e-04	-1,6402e-04	1,6108e-03	6,0274e-04	2,4767e-04

Tabela 4.26: Razlika vrednosti indikatorja I_R za algoritma GDE3 in DEMOwSA na testnih funkcijah 8–13 z $M = 5$.

FE_s		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
$5e+3$	Najboljši	-1,1744e-04	-6,5037e-02	6,6864e-03	4,6856e-02	-4,9470e-02	3,6599e-03
	Srednji	-7,3880e-05	-8,0199e-02	3,4550e-03	4,7010e-02	-4,6799e-02	4,1573e-03
	Najslabši	1,3731e-04	-8,1016e-02	5,1720e-05	4,7127e-02	-4,4900e-02	2,6920e-03
	Poprečje	$-3,5980e-05^\dagger$	$-7,9623e-02^\dagger$	$3,5680e-03^\ddagger$	$4,6998e-02^\ddagger$	$-4,6758e-02^\dagger$	$3,7739e-03^\ddagger$
	Std	6,8133e-05	-2,9236e-03	-2,2172e-03	6,5005e-05	7,4282e-04	-5,5560e-04
$5e+4$	Najboljši	-8,3981e-05	-4,5521e-02	1,1978e-02	4,6835e-02	-4,0561e-02	3,8900e-04
	Srednji	-9,7253e-05	-5,8804e-02	8,9747e-03	4,6976e-02	-3,9439e-02	1,0080e-04
	Najslabši	-1,3347e-04	-6,1498e-02	5,6566e-03	4,7175e-02	-4,1168e-02	1,6620e-04
	Poprečje	$-9,7649e-05^\dagger$	$-5,4928e-02^\dagger$	$8,9351e-03^\ddagger$	$4,6991e-02^\ddagger$	$-4,0221e-02^\dagger$	$2,5590e-04^\ddagger$
	Std	-1,1544e-05	-5,9247e-03	-1,2492e-03	9,0727e-05	-8,2850e-04	-1,5724e-04
$5e+5$	Najboljši	-5,8258e-06	3,7188e-04	1,5884e-02	4,5485e-02	-3,9562e-03	8,0938e-04
	Srednji	-9,8700e-06	-4,3132e-04	8,9051e-03	4,5971e-02	-3,1527e-03	2,2650e-04
	Najslabši	-2,0288e-05	-6,7705e-03	5,4499e-03	4,6286e-02	-5,6583e-03	-6,0200e-05
	Poprečje	$-1,1077e-05^\dagger$	$-1,5789e-03^\dagger$	$9,2242e-03^\ddagger$	$4,5965e-02^\ddagger$	$-3,9567e-03^\dagger$	$4,2550e-04^\ddagger$
	Std	-4,4277e-06	-2,1875e-03	-1,9790e-03	2,1904e-04	-5,1720e-04	-2,1198e-04

Tabela 4.27: Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma GDE3 in DEMOwSA na testnih funkcijah 1–7.

<i>FEs</i>		1. OKA2	2. SYMPART	3. S_ZDT1	4. S_ZDT2	5. S_ZDT4	6. R_ZDT4	7. S_ZDT6
5e+3	Najboljši	2,3660e-02	-4,3470e-03	-1,5390e-02	1,8260e-02	5,6300e-02	2,5015e-02	3,1800e-03
	Srednji	2,9818e-02	1,5781e-02	3,5100e-03	1,7530e-02	4,1130e-02	4,3471e-02	1,4370e-02
	Najslabši	2,6331e-02	3,5450e-02	2,3230e-02	2,7500e-02	4,2060e-02	3,5567e-02	1,1140e-02
	Poprečje	$2,9242e-02^{\ddagger}$	$1,0895e-02^{\ddagger}$	$3,9600e-03^{\ddagger}$	$1,8880e-02^{\ddagger}$	$3,6750e-02^{\ddagger}$	$3,8502e-02^{\ddagger}$	$1,0620e-02^{\ddagger}$
	Std	1,4206e-03	7,3360e-03	9,9228e-03	4,6070e-03	-1,3190e-03	4,3060e-03	1,1728e-03
5e+4	Najboljši	1,2880e-02	-2,2063e-05	-2,2643e-04	-2,4560e-04	-3,5230e-03	1,1235e-03	-3,6103e-02
	Srednji	1,7990e-02	-3,3455e-05	-2,4864e-04	6,0291e-04	1,2794e-02	4,2239e-03	-5,7610e-03
	Najslabši	3,1766e-02	-9,5738e-05	7,2425e-04	9,8600e-04	2,6176e-02	1,6171e-02	7,5432e-02
	Poprečje	$1,8843e-02^{\ddagger}$	$-3,6310e-05^{\dagger}$	$-2,1803e-04^{\dagger}$	$1,7104e-02^{\ddagger}$	$1,3761e-02^{\ddagger}$	$6,3448e-03^{\ddagger}$	$2,2180e-03$
	Std	4,5727e-03	-1,4743e-05	1,6020e-04	1,0986e-02	7,3091e-03	4,7410e-03	3,4226e-02
5e+5	Najboljši	6,4949e-03	-8,8500e-08	-1,7190e-05	-1,0500e-06	-4,0300e-09	-3,8360e-04	7,2000e-07
	Srednji	8,8476e-03	1,3410e-07	-2,5910e-05	2,3900e-06	1,4817e-03	-3,4950e-04	9,9000e-07
	Najslabši	2,6131e-02	1,8520e-07	-3,4710e-05	4,1400e-04	2,4163e-03	-1,6090e-03	1,0221e-01
	Poprečje	$1,0560e-02^{\ddagger}$	$1,3030e-07^{\ddagger}$	$-2,7370e-05^{\dagger}$	$1,7159e-02^{\ddagger}$	$1,1881e-03^{\ddagger}$	$-5,0500e-04^{\dagger}$	$8,1654e-03^{\ddagger}$
	Std	5,8022e-03	-7,6140e-08	-5,1102e-06	1,0979e-02	6,6889e-04	-2,9362e-04	2,8247e-02

Tabela 4.28: Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma GDE3 in DEMOwSA na testnih funkcijah 8–13 z $M = 3$.

<i>FEs</i>		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Najboljši	5,4260e-04	9,3582e-03	9,1194e-02	2,8607e-01	-9,8860e-02	2,3170e-02
	Srednji	6,6420e-04	1,5525e-02	8,1751e-02	2,8663e-01	-8,6674e-02	1,0980e-03
	Najslabši	7,4380e-04	2,5621e-02	7,0229e-02	2,8571e-01	-7,6084e-02	-1,5600e-02
	Poprečje	$7,5370e-04^{\dagger}$	$1,5613e-02^{\ddagger}$	$8,1527e-02^{\ddagger}$	$2,8636e-01^{\ddagger}$	$-8,6493e-02^{\dagger}$	1,0330e-03
	Std	1,4336e-04	4,0355e-03	-5,6777e-03	-7,5270e-05	5,2992e-03	-1,3136e-02
5e+4	Najboljši	2,2934e-05	1,1257e-02	1,6708e-01	2,7332e-01	-1,5768e-01	2,2953e-01
	Srednji	-1,5146e-05	1,5106e-02	1,6082e-01	2,7859e-01	-1,4561e-01	2,1500e-01
	Najslabši	-9,9220e-05	2,1189e-02	1,5571e-01	2,8084e-01	-1,3858e-01	1,9810e-01
	Poprečje	$-1,4930e-05^{\dagger}$	$1,5992e-02^{\ddagger}$	$1,6077e-01^{\ddagger}$	$2,7793e-01^{\ddagger}$	$-1,4698e-01^{\dagger}$	$2,1388e-01^{\ddagger}$
	Std	-3,1994e-05	2,9358e-03	-2,9879e-03	1,9686e-03	4,8130e-03	-1,0433e-02
5e+5	Najboljši	4,9054e-05	9,4432e-03	1,7596e-01	1,9567e-01	-1,6428e-01	2,9880e-01
	Srednji	8,8438e-05	1,1602e-02	1,7490e-01	2,1336e-01	-1,6045e-01	2,9617e-01
	Najslabši	1,3151e-04	1,9873e-02	1,7279e-01	2,2278e-01	-1,5359e-01	2,6614e-01
	Poprečje	$8,9724e-05^{\ddagger}$	$1,2067e-02^{\ddagger}$	$1,7461e-01^{\ddagger}$	$2,1157e-01^{\ddagger}$	$-1,6011e-01^{\dagger}$	$2,9305e-01^{\ddagger}$
	Std	2,4161e-05	2,4892e-03	-9,6861e-04	8,1197e-03	2,6295e-03	-7,2832e-03

Tabela 4.29: Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma GDE3 in DEMOwSA na testnih funkcijah 8–13 z $M = 5$.

<i>FEs</i>		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Najboljši	2,2032e-03	-3,2222e-01	4,6338e-02	5,2812e-01	-6,7210e-01	8,8512e-02
	Srednji	4,5800e-03	-3,9044e-01	2,8409e-02	5,2832e-01	-6,6403e-01	7,0486e-02
	Najslabši	3,7570e-03	-3,8871e-01	6,1792e-03	5,2825e-01	-6,5970e-01	3,6250e-02
	Poprečje	$4,4070e-03^{\ddagger}$	$-3,8690e-01^{\dagger}$	$2,8291e-02^{\ddagger}$	$5,2814e-01^{\ddagger}$	$-6,6475e-01^{\dagger}$	$7,2068e-02^{\ddagger}$
	Std	2,1000e-04	-1,1005e-02	-1,3692e-02	-1,0489e-04	3,0841e-03	-1,3909e-02
5e+4	Najboljši	-3,4178e-04	-2,2896e-01	7,0629e-02	5,2876e-01	-6,5042e-01	1,1640e-02
	Srednji	-5,2054e-04	-2,9221e-01	5,6641e-02	5,3035e-01	-6,5353e-01	5,4890e-03
	Najslabši	-7,9553e-04	-3,0038e-01	3,8720e-02	5,3300e-01	-6,7379e-01	-6,6810e-03
	Poprečje	$-5,2943e-04^{\dagger}$	$-2,7341e-01^{\dagger}$	$5,6515e-02^{\ddagger}$	$5,3046e-01^{\ddagger}$	$-6,6187e-01^{\dagger}$	$3,9290e-03^{\ddagger}$
	Std	-1,3300e-04	-2,7214e-02	-6,0298e-03	9,5213e-04	-1,1195e-02	-4,1911e-03
5e+5	Najboljši	-1,4326e-06	1,5296e-02	9,1893e-02	5,1661e-01	-2,3899e-01	-3,5000e-03
	Srednji	-5,2187e-06	9,4909e-03	5,7558e-02	5,2173e-01	-2,4175e-01	-1,0640e-02
	Najslabši	-2,0759e-05	-2,2811e-02	3,8588e-02	5,2419e-01	-2,7664e-01	-1,4630e-02
	Poprečje	$-5,8919e-06^{\dagger}$	$3,2941e-03^{\ddagger}$	$5,9748e-02^{\ddagger}$	$5,2139e-01^{\ddagger}$	$-2,4851e-01^{\dagger}$	$-1,0310e-02^{\dagger}$
	Std	-4,7432e-06	-1,1890e-02	-9,8129e-03	2,1013e-03	-9,6720e-03	-2,9142e-03

4.4.4 Algoritem NSGAII-SBX

NSGAII-SBX [108] Sharme in sodelavcev kombinira osnovni algoritem NSGA-II [79] in njegovo večkriterijsko selekcijo. Pri mutaciji v genetskem procesu uporablja operator SBX (ang. *Simulated Binary Crossover Operator* – SBX). V evolucijskem procesu uporablja tudi lokalno iskanje, sekvenčno kvadratično programiranje (ang. *Sequential Quadratic Programming* – SQP), ki precej izboljša dobljene rešitve eno generacijo pred vsakim izpisom aproksimacijske množice.

V tabelah 4.30–4.35 vidimo razlike rezultatov med algoritmoma NSGAII-SBX in DEMOwSA. Oba indikatorja ocenita, da je algoritem DEMOwSA boljši na vseh aproksimacijskih množicah pri funkcijah WFG8 z $M = 3$ ali $M = 5$ in funkciji R_DTLZ2 z $M = 5$. Ocenita tudi, da velja obratno, da je algoritem NSGAII-SBX boljši na vseh aproksimacijskih množicah, pri funkcijah OKA2, S_ZDT2, S_ZDT4, R_DTLZ2 z $M = 3$, WFG z $M = 3$ ali $M = 5$, S_DTLZ3 z $M = 5$ in WFG9 z $M = 5$. Kot vidimo, se oceni za enako funkcijo R_DTLZ2 z različnim parametrom $M = 3$ ali $M = 5$, povsem razlikujeta, kar gre pripisati selekcijskemu mehanizmu v DEMOwSA, ki pri $M = 5$ deluje bolje od NSGA-II. Ocene indikatorjev se skladajo tudi pri drugih funkcijah in se pri tem skozi optimizacijski postopek prevešajo v prid algoritma DEMOwSA, tj. pri funkcijah SYM-PART, R_ZDT4, S_ZDT6 in S_DTLZ2 z $M = 3$. Tudi pri funkciji S_ZDT1 se ocena za DEMOwSA glede na NSGAII-SBX izboljša na obeh indikatorjih po 5e+4 ovrednotenjih, pri indikatorju $I_{\bar{H}}$ tudi po 5e+5 ovrednotenjih. Pri funkciji S_DTLZ3 je algoritem DEMOwSA boljši od algoritma NSGAII-SBX na indikatorju I_R pri 5e+3 ovrednotenjih, a slabši drugod. Pri S_DTLZ2 z $M = 5$ je algoritem NSGAII-SBX boljši od algoritma DEMOwSA le na indikatorju $I_{\bar{H}}$ pri 5e+3 ovrednotenjih, pri ostalih aproksimacijskih množicah na tem indikatorju in indikatorju I_R je povsod za to funkcijo boljši algoritem DEMOwSA.

Rezultati nakazujejo, da je algoritem NSGAII-SBX na danem testnem ogrodju in z uporabljenou nastavitvijo krmilnih parametrov v povprečju boljši od algoritma DEMOwSA. Razlike gre pripisati uporabljenemu lokalnemu iskanju oz. uspešni hibridizaciji algoritma z lokalnim iskanjem.

Tabela 4.30: Razlika vrednosti indikatorja I_R za algoritma NSGAII-SBX in DEMoWSA na testnih funkcijah 1–7.

<i>FES</i>		1. OKA2	2. SYMPART	3. S_ZDT1	4. S_ZDT2	5. S_ZDT4	6. R_ZDT4	7. S_ZDT6
5e+3	Najboljši	9,6566e–04	1,7174e–02	2,1457e–02	6,0037e–02	4,3431e–02	1,2561e–02	1,2025e–01
	Srednji	1,8020e–03	3,1363e–02	2,9356e–02	6,1820e–02	4,6355e–02	2,3432e–02	1,2762e–01
	Najslabši	–1,4439e–02	4,8955e–02	3,6953e–02	6,5860e–02	4,2428e–02	2,0940e–02	1,3133e–01
	Poprečje	$8,8540e-04^\ddagger$	$3,0497e-02^\ddagger$	$3,0266e-02^\ddagger$	$5,7890e-02^\ddagger$	$4,5655e-02^\ddagger$	$2,0168e-02^\ddagger$	$1,2730e-01^\ddagger$
	Std	–2,8691e–03	7,4144e–03	3,4996e–03	–1,2915e–03	1,2360e–04	2,2843e–03	–1,2839e–03
5e+4	Najboljši	1,8035e–04	–7,6389e–05	–6,1764e–05	6,8157e–05	4,1504e–03	–9,3650e–05	–1,7410e–02
	Srednji	4,7224e–04	–1,2641e–04	–7,5615e–05	4,4722e–04	1,0410e–02	–7,4640e–04	–5,6780e–03
	Najslabši	2,1687e–03	–1,8674e–04	1,8415e–04	4,0272e–02	1,6438e–02	2,0458e–03	2,4527e–02
	Poprečje	$5,6882e-04^\ddagger$	$–1,3271e-04^\dagger$	$–6,4418e-05^\dagger$	$1,7729e-02^\ddagger$	$1,0781e-02^\ddagger$	4,9100e–05	$–4,2300e-03^\dagger$
	Std	4,3768e–04	–2,7853e–05	4,3210e–05	2,0223e–02	3,0744e–03	5,9080e–04	1,3617e–02
5e+5	Najboljši	6,5880e–05	–1,2838e–05	5,0853e–08	2,0629e–04	–3,7406e–09	–2,0905e–04	–8,5150e–03
	Srednji	1,5495e–04	–1,7436e–05	2,9542e–07	2,0737e–04	3,0909e–04	–1,3836e–03	–1,1911e–02
	Najslabši	9,9992e–04	–2,7133e–05	4,6161e–06	4,0396e–02	6,8866e–04	–3,5634e–03	3,1180e–02
	Poprečje	$2,2320e-04^\ddagger$	$–1,8643e-05^\dagger$	$7,2819e-07^\ddagger$	$1,7840e-02^\ddagger$	$2,6826e-04^\ddagger$	$–1,5514e-03^\dagger$	$–8,2876e-03^\dagger$
	Std	2,2104e–04	–4,5449e–06	9,8680e–07	2,0304e–02	1,8108e–04	–9,8451e–04	1,0866e–02

Tabela 4.31: Razlika vrednosti indikatorja I_R za algoritma NSGAII-SBX in DEMoWSA na testnih funkcijah 8–13 z $M = 3$.

<i>FES</i>		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Najboljši	4,5105e–05	1,3775e–04	–1,0446e–03	5,6007e–02	–1,3746e–02	–8,3514e–03
	Srednji	7,5910e–05	1,6258e–03	–3,3010e–03	5,6305e–02	–1,1564e–02	–9,7187e–03
	Najslabši	1,7701e–04	4,0559e–03	–5,5647e–03	5,6453e–02	–8,1404e–03	–8,7003e–03
	Poprečje	$9,6630e-05^\ddagger$	$1,6755e-03^\ddagger$	$–3,2959e-03^\dagger$	$5,6268e-02^\ddagger$	$–1,1162e-02^\dagger$	$–9,5824e-03^\dagger$
	Std	4,2197e–05	9,4465e–04	–1,1802e–03	1,1978e–04	1,3666e–03	–1,2600e–04
5e+4	Najboljši	–6,2990e–05	1,8425e–04	8,9306e–03	5,3514e–02	–2,4940e–02	3,9500e–05
	Srednji	–7,8088e–05	7,2183e–04	7,8474e–03	5,4642e–02	–2,2843e–02	–3,7400e–05
	Najslabši	–7,6475e–05	1,9434e–03	6,0031e–03	5,5080e–02	–2,0654e–02	1,3165e–03
	Poprečje	$–7,7361e-05^\dagger$	$8,1913e-04^\ddagger$	$7,6902e-03^\ddagger$	$5,4472e-02^\ddagger$	$–2,2823e-02^\dagger$	–1,9700e–05
	Std	–3,4430e–06	5,2326e–04	–7,5240e–04	4,1565e–04	1,1082e–03	2,4945e–04
5e+5	Najboljši	–6,7643e–05	1,9487e–04	1,2800e–02	3,7756e–02	–2,6671e–02	3,4298e–03
	Srednji	–8,1594e–05	2,7219e–04	1,2491e–02	4,1315e–02	–2,5825e–02	3,8516e–03
	Najslabši	–8,2048e–05	1,8595e–03	1,1766e–02	4,3175e–02	–2,4263e–02	4,7992e–03
	Poprečje	$–7,5566e-05^\dagger$	$4,4422e-04^\ddagger$	$1,2382e-02^\ddagger$	$4,0940e-02^\ddagger$	$–2,5818e-02^\dagger$	$3,9580e-03^\ddagger$
	Std	–5,8379e–06	3,9116e–04	–2,8387e–04	1,6026e–03	6,0301e–04	2,8819e–04

Tabela 4.32: Razlika vrednosti indikatorja I_R za algoritma NSGAII-SBX in DEMoWSA na testnih funkcijah 8–13 z $M = 5$.

<i>FES</i>		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Najboljši	–1,5630e–04	–7,2739e–02	8,2914e–03	4,6989e–02	–5,5819e–02	7,1850e–03
	Srednji	–8,8300e–05	–7,3819e–02	6,4636e–03	4,7183e–02	–5,4482e–02	9,0490e–03
	Najslabši	8,0190e–05	–7,5486e–02	–7,9477e–04	4,7335e–02	–5,3427e–02	9,2560e–03
	Poprečje	$–5,8150e-05^\ddagger$	$–7,4014e-02^\dagger$	$5,4402e-03^\ddagger$	$4,7169e-02^\ddagger$	$–5,4270e-02^\dagger$	$8,8350e-03^\ddagger$
	Std	6,7150e–05	–7,0676e–04	–2,6602e–03	8,1578e–05	1,0430e–04	7,3070e–04
5e+4	Najboljši	–9,8880e–05	–5,0174e–02	1,2928e–02	4,6886e–02	–4,5423e–02	3,8594e–03
	Srednji	–1,2210e–04	–5,4205e–02	9,2247e–03	4,7040e–02	–4,4463e–02	4,7015e–03
	Najslabši	–1,3363e–04	–5,6748e–02	9,1221e–03	4,7206e–02	–4,3014e–02	5,9649e–03
	Poprečje	$–1,1712e-04^\dagger$	$–5,3451e-02^\dagger$	$9,8128e-03^\ddagger$	$4,7052e-02^\ddagger$	$–4,4235e-02^\dagger$	$4,8116e-03^\ddagger$
	Std	–1,1837e–05	–2,7414e–03	–1,2568e–03	8,7930e–05	2,4200e–04	5,1218e–04
5e+5	Najboljši	–6,4338e–06	–1,5356e–02	1,4201e–02	4,5518e–02	–3,5467e–02	3,6874e–03
	Srednji	–1,0774e–05	–2,3167e–02	9,2810e–03	4,6016e–02	–3,6359e–02	4,7633e–03
	Najslabši	–1,2087e–05	–2,9349e–02	9,2451e–03	4,6345e–02	–3,7059e–02	5,4915e–03
	Poprečje	$–1,0550e-05^\dagger$	$–2,2441e-02^\dagger$	$1,0212e-02^\ddagger$	$4,6010e-02^\ddagger$	$–3,6288e-02^\dagger$	$4,6183e-03^\ddagger$
	Std	–1,5058e–06	–3,5768e–03	–1,8986e–03	2,2404e–04	–6,7510e–04	4,7173e–04

Tabela 4.33: Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma NSGAII-SBX in DEMoWSA na testnih funkcijah 1–7.

<i>FES</i>		1. OKA2	2. SYMPART	3. S_ZDT1	4. S_ZDT2	5. S_ZDT4	6. R_ZDT4	7. S_ZDT6
5e+3	Najboljši	2,3754e–02	4,9097e–02	8,5500e–02	1,5733e–01	1,3326e–01	3,7162e–02	8,0010e–02
	Srednji	2,8480e–02	8,8876e–02	1,1102e–01	1,5940e–01	1,4525e–01	7,0429e–02	7,4800e–02
	Najslabši	1,1334e–02	1,3710e–01	1,3118e–01	1,4499e–01	1,3679e–01	6,0690e–02	7,0880e–02
	Poprečje	2,7850e–02 [‡]	8,6361e–02 [‡]	1,0979e–01 [†]	1,5647e–01 [‡]	1,4139e–01 [‡]	6,1657e–02 [‡]	7,6090e–02 [‡]
	Std	–1,4265e–03	2,0588e–02	1,1084e–02	4,4800e–04	1,6360e–03	4,8650e–03	–3,6824e–03
5e+4	Najboljši	1,2738e–02	–2,2667e–04	–3,7296e–04	1,1977e–02	1,2697e–02	6,2900e–05	–3,9175e–02
	Srednji	1,7899e–02	–3,7484e–04	–4,7100e–04	1,2797e–02	3,1021e–02	–2,2391e–03	–1,8619e–02
	Najslabši	3,1766e–02	–5,5273e–04	5,6390e–04	6,0298e–02	4,8596e–02	6,0390e–03	5,7030e–02
	Poprečje	1,8747e–02 [‡]	–3,9323e–04 [†]	–4,2454e–04 [†]	3,3093e–02 [‡]	3,2118e–02 [‡]	1,1250e–04	–9,6520e–03 [†]
	Std	4,6088e–03	–8,2072e–05	1,6822e–04	2,4016e–02	8,9903e–03	1,9754e–03	3,1252e–02
5e+5	Najboljši	6,3490e–03	–4,0968e–05	–1,3477e–04	1,2331e–02	–3,8692e–07	–1,1503e–03	–1,8621e–02
	Srednji	8,6716e–03	–5,4698e–05	–1,6014e–04	1,2322e–02	1,4811e–03	–3,9600e–03	–2,6480e–02
	Najslabši	2,5911e–02	–8,3885e–05	–2,0811e–04	6,0296e–02	2,5803e–03	–1,0537e–02	7,0770e–02
	Poprečje	1,0380e–02 [‡]	–5,8242e–05 [†]	–1,6293e–04 [†]	3,3294e–02 [‡]	1,2139e–03 [‡]	–4,7570e–03 [†]	–1,8245e–02 [†]
	Std	5,7857e–03	–1,3678e–05	–2,0525e–05	2,4143e–02	7,3049e–04	–2,8734e–03	2,4629e–02

Tabela 4.34: Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma NSGAII-SBX in DEMoWSA na testnih funkcijah 8–13 z $M = 3$.

<i>FES</i>		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Najboljši	1,0137e–03	1,2854e–02	2,6661e–01	2,8690e–01	–9,6992e–02	1,9761e–01
	Srednji	1,5192e–03	1,9253e–02	2,3400e–01	2,8824e–01	–8,4217e–02	1,6917e–01
	Najslabši	2,2516e–03	3,0791e–02	2,0764e–01	2,8901e–01	–7,1622e–02	1,4312e–01
	Poprečje	1,7038e–03 [‡]	1,9476e–02 [‡]	2,3408e–01 [†]	2,8806e–01 [‡]	–8,3901e–02 [†]	1,7040e–01 [‡]
	Std	4,1426e–04	4,4736e–03	–1,4041e–02	5,3945e–04	5,9267e–03	–1,1587e–02
5e+4	Najboljši	–4,5475e–04	1,1257e–02	3,7921e–01	2,7334e–01	–1,5768e–01	3,2018e–01
	Srednji	–1,0460e–03	1,5106e–02	3,3706e–01	2,7863e–01	–1,4560e–01	3,0217e–01
	Najslabši	–2,0788e–03	2,1188e–02	3,0759e–01	2,8095e–01	–1,3856e–01	2,5539e–01
	Poprečje	–1,0133e–03 [†]	1,5992e–02 [‡]	3,4055e–01 [†]	2,7798e–01 [‡]	–1,4697e–01 [†]	2,9608e–01 [‡]
	Std	–3,0984e–04	2,9353e–03	–1,8451e–02	1,9888e–03	4,8168e–03	–1,7354e–02
5e+5	Najboljši	–4,4857e–04	9,4432e–03	4,2460e–01	1,9568e–01	–1,6428e–01	3,7253e–01
	Srednji	–1,2538e–03	1,1602e–02	3,9110e–01	2,1337e–01	–1,6045e–01	3,6663e–01
	Najslabši	–1,8385e–03	1,9873e–02	3,7420e–01	2,2276e–01	–1,5359e–01	3,3646e–01
	Poprečje	–1,2111e–03 [†]	1,2067e–02 [‡]	3,9910e–01 [‡]	2,1157e–01 [‡]	–1,6011e–01 [†]	3,6249e–01 [‡]
	Std	–3,5132e–04	2,4892e–03	–1,7560e–02	8,1102e–03	2,6295e–03	–1,1637e–02

Tabela 4.35: Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma NSGAII-SBX in DEMoWSA na testnih funkcijah 8–13 z $M = 5$.

<i>FES</i>		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Najboljši	2,6851e–03	–6,3468e–01	4,7703e–02	5,3012e–01	–7,2941e–01	2,8016e–01
	Srednji	4,8800e–03	–6,4199e–01	3,3396e–02	5,3119e–01	–7,3350e–01	2,9078e–01
	Najslabši	5,4740e–03	–6,5103e–01	–4,7150e–03	5,3254e–01	–7,3598e–01	2,6593e–01
	Poprečje	4,6910e–03 [‡]	–6,4180e–01 [†]	3,1178e–02 [‡]	5,3109e–01 [‡]	–7,3229e–01 [†]	2,8555e–01 [‡]
	Std	1,4550e–04	–4,6382e–03	–1,5673e–02	6,4437e–04	–2,6141e–03	–1,6840e–03
5e+4	Najboljši	–3,0458e–04	–5,1769e–01	5,8637e–02	5,2886e–01	–7,0383e–01	1,1039e–01
	Srednji	–5,5477e–04	–5,2609e–01	4,6139e–02	5,3053e–01	–7,0370e–01	1,1582e–01
	Najslabši	–8,4070e–04	–5,2852e–01	3,3460e–02	5,3310e–01	–6,9201e–01	1,2359e–01
	Poprečje	–5,7231e–04 [†]	–5,2446e–01 [†]	4,6202e–02 [‡]	5,3062e–01 [‡]	–7,0310e–01 [†]	1,1622e–01 [‡]
	Std	–1,4699e–04	–4,0342e–03	–7,0092e–03	9,5449e–04	8,5250e–04	3,9085e–03
5e+5	Najboljši	–1,7736e–06	–3,1377e–01	7,7460e–02	5,1664e–01	–6,2349e–01	8,6100e–02
	Srednji	–4,3496e–06	–4,1615e–01	4,8770e–02	5,2177e–01	–6,3785e–01	9,1460e–02
	Najslabši	–1,8810e–05	–4,5040e–01	3,6310e–02	5,2427e–01	–6,3687e–01	9,7230e–02
	Poprečje	–5,2495e–06 [†]	–4,1429e–01 [†]	5,1580e–02 [‡]	5,2144e–01 [‡]	–6,3365e–01 [†]	9,1940e–02 [‡]
	Std	–3,3693e–06	–2,6476e–02	–1,1340e–02	2,1147e–03	–4,2400e–03	2,9023e–03

4.4.5 Algoritem NSGAII-PCX

NSGAII-PCX [109] Kumarja in sodelavcev prav tako uporablja osnovni algoritem NSGA-II [79] in njegovo večkriterijsko selekcijo. Pri mutaciji v genetskem procesu uporablja operator PCX (ang. *Parent-Centric based recombination*). Prav tako kot zgornji algoritem NSGAII-SBX, uporablja še isti tip lokalnega iskanja, na enak način.

V tabelah 4.36–4.41 vidimo razlike med rezultati ocen indikatorjev I_R in $I_{\bar{H}}$ za algoritma DEMOwSA in NSGAII-PCX. Algoritem DEMOwSA je na obeh indikatorjih pri vseh vzorčenjih aproksimacijskih množic boljši za funkcije S_ZDT1, S_ZDT6, S_DTLZ2 z $M = 3$ ali $M = 5$, WFG8 z $M = 3$ ali $M = 5$ in R_DTLZ2 z $M = 5$. Obratno lahko isto opažanje za algoritem NSGAII-PCX podamo pri funkcijah S_ZDT4, R_DTLZ2 z $M = 3$, S_DTLZ3 z $M = 3$ ali $M = 5$, WFG1 z $M = 3$ ali $M = 5$ in WFG9 z $M = 3$.

Na indikatorju I_R algoritem DEMOwSA daje boljše rezultate pri $5e+3$ ovrednotenjih in $5e+4$ ovrednotenjih za funkcijo OKA2, pri $5e+4$ ovrednotenjih in $5e+5$ ovrednotenjih za funkcijo SYM-PART na obeh indikatorjih, tudi pri $5e+3$ ovrednotenjih za funkcijo S_ZDT2 in pri $5e+5$ ovrednotenjih za funkcijo R_ZDT4 ter na indikatorju $I_{\bar{H}}$ pri $5e+5$ ovrednotenjih za funkcijo WFG9 z $M = 5$. Na indikatorju I_R algoritem NSGAII-PCX daje boljše rezultate pri $5e+5$ ovrednotenjih za funkcijo OKA2, pri $5e+3$ ovrednotenjih za funkcijo SYM-PART, pri $5e+4$ ovrednotenjih in $5e+5$ ovrednotenjih za funkcijo S_ZDT2, pri $5e+3$ ovrednotenjih in $5e+4$ ovrednotenjih za funkcijo R_ZDT4 ter pri funkciji WFG9 z $M = 5$. Na indikatorju $I_{\bar{H}}$ opazimo za slednje funkcije podobno, tu je algoritem NSGAII-PCX boljši še pri $5e+3$ ovrednotenjih in $5e+4$ ovrednotenjih za funkcijo OKA2. Vendar pri $5e+5$ ovrednotenjih za funkcijo OKA2 ni več opaziti signifikantne razlike na indikatorju $I_{\bar{H}}$. Indikatorja za funkcijo WFG9 z $M = 3$ kažeta nasprotne ocene, pri tem indikator I_R za vse vzorčene aproksimacijske množice kaže, da je algoritem DEMOwSA boljši, indikator $I_{\bar{H}}$ pa bolje oceni aproksimacijske množice algoritma NSGAII-PCX.

Iz rezultatov je sklepati, da sta algoritma dokaj izenačena po kakovosti dobljenih aproksimacijskih množic. Vendar, če pogledamo pri $5e+5$ ovrednotenjih za vse funkcije, vidimo, da je algoritem DEMOwSA na obeh indikatorjih večkrat boljši.

Tabela 4.36: Razlika vrednosti indikatorja I_R za algoritma NSGAII-PBX in DEMoWSA na testnih funkcijah 1–7.

<i>FEs</i>		1. OKA2	2. SYMPART	3. S_ZDT1	4. S_ZDT2	5. S_ZDT4	6. R_ZDT4	7. S_ZDT6
5e+3	Najboljši	9,0866e-04	1,7739e-02	-2,7713e-02	-2,6173e-02	1,8181e-02	1,6854e-02	-4,4700e-03
	Srednji	-2,7925e-03	3,2383e-02	-3,8914e-02	-2,5060e-02	2,5650e-03	2,9424e-02	-9,0000e-03
	Najslabši	-4,4679e-02	5,0551e-02	-4,3837e-02	-1,6210e-02	-6,8020e-03	3,4859e-02	-9,2700e-03
	Poprečje	-7,3691e-03[†]	3,1526e-02[‡]	-3,5534e-02[†]	-2,6120e-02[†]	1,4350e-03	2,7033e-02[‡]	-6,1900e-03[†]
	Std	-1,0372e-02	7,7017e-03	-3,7434e-03	-1,0915e-03	-7,0354e-03	4,2663e-03	-3,1639e-03
5e+4	Najboljši	-3,3650e-05	-1,3529e-05	-2,7529e-03	-6,7340e-03	1,8540e-04	3,4415e-04	-5,2191e-02
	Srednji	-2,9408e-03	-2,3057e-05	-4,5570e-03	-7,5493e-03	4,0000e-04	1,3741e-03	-4,1498e-02
	Najslabši	-2,2114e-02	-3,1624e-05	-6,2302e-03	1,5817e-02	-2,2020e-03	5,3478e-03	-1,1773e-02
	Poprečje	-5,4172e-03[†]	-2,2586e-05[‡]	-4,6601e-03[†]	8,8620e-03[‡]	2,3800e-04	1,9611e-03[‡]	-3,9720e-02[†]
	Std	-6,3320e-03	-4,4183e-06	-3,5300e-05	1,6892e-02	-5,5240e-04	1,5873e-03	1,2425e-02
5e+5	Najboljši	6,5880e-05	-3,5678e-05	-1,9560e-04	-2,1641e-04	-2,5365e-07	-1,5575e-04	-1,5471e-02
	Srednji	-6,0645e-04	-6,4256e-05	-7,0501e-04	-8,6503e-04	3,0710e-04	-2,6503e-04	-1,8651e-02
	Najslabši	-2,4478e-04	-8,4573e-05	-1,9446e-03	3,7050e-02	6,5205e-04	-4,8740e-04	2,4950e-02
	Poprečje	8,0220e-04[‡]	-6,2413e-05[†]	-7,9726e-04[†]	1,6672e-02[‡]	2,6294e-04[‡]	-2,8902e-04[†]	-1,4968e-02[†]
	Std	-5,1496e-04	-1,5418e-05	-5,0130e-04	1,9690e-02	1,7321e-04	-5,8410e-05	1,1220e-02

Tabela 4.37: Razlika vrednosti indikatorja I_R za algoritma NSGAII-PBX in DEMoWSA na testnih funkcijah 8–13 z $M = 3$.

<i>FEs</i>		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Najboljši	-1,6156e-04	-4,4150e-05	1,3098e-02	5,6007e-02	-1,3746e-02	-9,2401e-03
	Srednji	-4,5194e-04	1,4176e-03	9,2720e-03	5,6128e-02	-1,1571e-02	-1,0017e-02
	Najslabši	-8,6035e-04	3,7469e-03	6,6113e-03	5,6000e-02	-8,4507e-03	-8,9873e-03
	Poprečje	-4,3529e-04[†]	1,4588e-03[‡]	9,6281e-03[‡]	5,6070e-02[‡]	-1,1203e-02[†]	-9,7874e-03[†]
	Std	-1,4858e-04	9,0397e-04	-1,9692e-03	-4,8230e-05	1,2810e-03	-5,7000e-05
5e+4	Najboljši	-1,3262e-04	1,3719e-04	2,1725e-02	5,3386e-02	-2,4940e-02	-4,9475e-03
	Srednji	-2,7507e-04	6,5682e-04	1,7817e-02	5,4379e-02	-2,3189e-02	-7,2082e-03
	Najslabši	-6,4758e-04	1,8395e-03	1,3043e-02	5,4701e-02	-2,1277e-02	-8,1715e-03
	Poprečje	-2,9115e-04[†]	7,5232e-04[‡]	1,7427e-02[‡]	5,4203e-02[‡]	-2,3111e-02[†]	-7,1700e-03[†]
	Std	-1,1311e-04	5,1137e-04	-2,1727e-03	3,6479e-04	8,9030e-04	-9,4205e-04
5e+5	Najboljši	-5,9493e-05	1,7229e-04	2,9250e-02	3,7739e-02	-2,6671e-02	-1,1932e-03
	Srednji	-6,7974e-05	2,4146e-04	2,7221e-02	4,1191e-02	-2,6251e-02	-6,7164e-03
	Najslabši	-2,5715e-04	1,8195e-03	2,4056e-02	4,3009e-02	-2,4913e-02	-8,4278e-03
	Poprečje	-8,1156e-05[†]	4,1400e-04[‡]	2,7002e-02[‡]	4,0818e-02[‡]	-2,6209e-02[†]	-6,1970e-03[†]
	Std	-4,1328e-05	3,8657e-04	-1,3916e-03	1,5691e-03	4,3251e-04	-1,9201e-03

Tabela 4.38: Razlika vrednosti indikatorja I_R za algoritma NSGAII-PBX in DEMoWSA na testnih funkcijah 8–13 z $M = 5$.

<i>FEs</i>		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Najboljši	-1,6260e-04	-5,5719e-02	8,7034e-03	4,6986e-02	-4,1019e-02	9,3088e-03
	Srednji	-1,6340e-04	-5,6189e-02	5,5756e-03	4,7143e-02	-3,8032e-02	1,1870e-02
	Najslabši	3,9000e-07	-5,6405e-02	5,5063e-04	4,7167e-02	-3,5637e-02	1,2585e-02
	Poprečje	-1,3275e-04[†]	-5,6104e-02[†]	5,3942e-03[‡]	4,7118e-02[‡]	-3,8060e-02[†]	1,1589e-02[‡]
	Std	6,1960e-05	-1,2716e-04	-2,3912e-03	4,1566e-05	9,6640e-04	7,5370e-04
5e+4	Najboljši	-3,0160e-05	-5,5424e-02	1,0438e-02	4,6879e-02	-4,3483e-02	3,7214e-03
	Srednji	-1,7000e-04	-5,6275e-02	6,9637e-03	4,6969e-02	-4,2063e-02	3,6475e-03
	Najslabši	-2,1133e-04	-5,7058e-02	5,2501e-03	4,6963e-02	-4,0414e-02	3,4136e-03
	Poprečje	-1,5182e-04[†]	-5,6291e-02[†]	7,0058e-03[‡]	4,6977e-02[‡]	-4,1995e-02[†]	3,7073e-03[‡]
	Std	-5,4707e-05	-3,9030e-04	-1,1108e-03	3,8080e-05	4,4530e-04	-1,6900e-04
5e+5	Najboljši	-7,7938e-06	-4,9766e-02	1,2321e-02	4,5516e-02	-4,4637e-02	3,4684e-03
	Srednji	-2,6814e-05	-5,1927e-02	8,9750e-03	4,5943e-02	-4,3559e-02	3,4116e-03
	Najslabši	-4,7417e-05	-5,3709e-02	7,1221e-03	4,6102e-02	-4,3979e-02	2,3864e-03
	Poprečje	-2,6060e-05[†]	-5,2011e-02[†]	9,4240e-03[‡]	4,5920e-02[‡]	-4,3968e-02[†]	3,2085e-03[‡]
	Std	-8,5748e-06	-1,0918e-03	-1,6906e-03	1,5672e-04	1,4590e-04	-2,3192e-04

Tabela 4.39: Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma NSGAII-PBX in DEMoWSA na testnih funkcijah 1–7.

<i>FES</i>		1. OKA2	2. SYMPART	3. S_ZDT1	4. S_ZDT2	5. S_ZDT4	6. R_ZDT4	7. S_ZDT6
5e+3	Najboljši	2,1477e–02	5,0739e–02	–7,5970e–02	–7,7340e–02	5,2260e–02	4,9134e–02	–1,0490e–02
	Srednji	1,8037e–02	9,1820e–02	–1,0868e–01	–1,0063e–01	–4,2500e–03	8,8009e–02	–2,6900e–02
	Najslabši	–2,4356e–02	1,4170e–01	–1,4517e–01	–1,3181e–01	–2,1210e–02	9,3680e–02	–2,7020e–02
	Poprečje	1,3418e–02[‡]	8,9335e–02[‡]	–1,0798e–01[†]	–1,0128e–01[†]	–1,6100e–03	8,2107e–02[‡]	–1,8810e–02[†]
	Std	–1,0902e–02	2,1414e–02	–1,4939e–02	–1,6932e–02	–2,1634e–02	1,1541e–02	–8,6124e–03
5e+4	Najboljši	1,2173e–02	–3,9685e–05	–1,2525e–02	–1,3043e–02	7,3800e–04	1,3499e–03	–1,2391e–01
	Srednji	1,3863e–02	–6,8242e–05	–1,6210e–02	–1,7043e–02	8,1200e–04	3,7639e–03	–1,0560e–01
	Najslabši	–5,5540e–03	–9,3228e–05	–2,0728e–02	2,0448e–02	–6,2890e–03	1,5567e–02	–3,1920e–02
	Poprečje	1,0305e–02[‡]	–6,6733e–05[†]	–1,6703e–02[†]	2,5930e–03	6,3900e–04	5,8405e–03[‡]	–9,6882e–02[†]
	Std	–4,7832e–03	–1,3182e–05	–2,2011e–03	2,0850e–02	–1,8296e–03	4,6654e–03	2,7940e–02
5e+5	Najboljši	1,3260e–03	–1,0920e–04	–1,2665e–03	–9,9061e–04	–2,2539e–06	–5,2656e–04	–3,4341e–02
	Srednji	–9,1940e–04	–1,9336e–04	–1,5038e–03	–1,2775e–03	1,4718e–03	–1,1010e–03	–4,1220e–02
	Najslabši	1,1050e–02	–2,5368e–04	–2,1903e–03	4,6022e–02	2,4081e–03	–1,6335e–03	5,6870e–02
	Poprečje	7,6990e–04	–1,8792e–04[†]	–1,5455e–03[†]	1,9652e–02[‡]	1,1806e–03[‡]	–1,2010e–03[†]	–3,3385e–02[†]
	Std	3,1649e–03	–4,5718e–05	–2,0426e–04	2,3908e–02	6,8259e–04	–2,5342e–04	2,5446e–02

Tabela 4.40: Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma NSGAII-PBX in DEMoWSA na testnih funkcijah 8–13 z $M = 3$.

<i>FES</i>		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Najboljši	–1,1997e–02	6,7560e–03	7,5751e–02	2,8688e–01	–9,6992e–02	1,8781e–01
	Srednji	–1,8313e–02	1,0841e–02	6,5992e–02	2,8622e–01	–8,4272e–02	1,5687e–01
	Najslabši	–4,1585e–02	1,3641e–02	3,9972e–02	2,7083e–01	–9,0239e–02	1,5042e–01
	Poprečje	–2,1454e–02[†]	1,0501e–02[‡]	6,3397e–02[‡]	2,8397e–01[†]	–8,5240e–02[†]	1,5940e–01[‡]
	Std	–8,1583e–03	2,3338e–03	3,4520e–04	–5,1551e–03	1,9193e–03	–8,9466e–03
5e+4	Najboljši	–1,9949e–03	1,1130e–02	1,2861e–01	2,7234e–01	–1,5768e–01	2,5428e–01
	Srednji	–3,8430e–03	1,4885e–02	1,0786e–01	2,7457e–01	–1,7098e–01	2,0807e–01
	Najslabši	–7,1218e–03	2,0570e–02	8,3603e–02	2,7001e–01	–2,0362e–01	1,6129e–01
	Poprečje	–3,9663e–03[†]	1,5755e–02[‡]	1,0405e–01[‡]	2,7336e–01[‡]	–1,7215e–01[†]	2,0548e–01[‡]
	Std	–1,2675e–03	2,8313e–03	–1,2541e–02	–2,8290e–04	–1,6593e–02	–2,3954e–02
5e+5	Najboljši	–4,1037e–04	9,4373e–03	1,7410e–01	1,9562e–01	–1,6428e–01	3,0463e–01
	Srednji	–1,7018e–03	1,1588e–02	1,5780e–01	2,1251e–01	–2,0477e–01	2,0093e–01
	Najslabši	–2,8175e–03	1,9848e–02	1,3590e–01	2,2104e–01	–2,7479e–01	1,6026e–01
	Poprečje	–1,7071e–03[†]	1,2053e–02[‡]	1,5790e–01[‡]	2,1061e–01[‡]	–2,0364e–01[†]	2,1369e–01[‡]
	Std	–6,4172e–04	2,4833e–03	–1,1040e–02	7,6496e–03	–2,2520e–02	–3,9327e–02

Tabela 4.41: Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma NSGAII-PBX in DEMoWSA na testnih funkcijah 8–13 z $M = 5$.

<i>FES</i>		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Najboljši	3,8210e–04	–2,7568e–01	5,1073e–02	5,3006e–01	–5,7801e–01	3,0826e–01
	Srednji	–8,7800e–03	–2,7439e–01	1,1734e–02	5,3076e–01	–5,6590e–01	3,2788e–01
	Najslabši	–1,0726e–02	–2,6973e–01	–9,8050e–03	5,2959e–01	–5,5568e–01	2,8903e–01
	Poprečje	–6,7890e–03[†]	–2,7341e–01[†]	1,6893e–02[‡]	5,3045e–01[‡]	–5,6689e–01[†]	3,2005e–01[‡]
	Std	–2,9745e–03	2,0898e–03	–1,7333e–02	–5,1150e–05	5,7569e–03	–5,4040e–03
5e+4	Najboljši	–2,5914e–05	–2,7559e–01	4,6317e–02	5,2877e–01	–6,7633e–01	1,0789e–01
	Srednji	–1,7296e–03	–2,7849e–01	2,6359e–02	5,2972e–01	–6,7220e–01	9,9819e–02
	Najslabši	–2,9689e–03	–2,7812e–01	9,0014e–03	5,2566e–01	–6,6551e–01	8,3090e–02
	Poprečje	–1,5705e–03[†]	–2,7826e–01[†]	2,6972e–02[‡]	5,2957e–01[‡]	–6,7290e–01[†]	9,7819e–02[‡]
	Std	–8,9199e–04	–5,3820e–04	–9,3812e–03	–4,5423e–04	3,3635e–03	–6,7865e–03
5e+5	Najboljši	–2,6596e–06	–2,5167e–01	7,2530e–02	5,1661e–01	–7,0539e–01	–1,0863e–01
	Srednji	–3,7577e–05	–2,6045e–01	4,7310e–02	5,2097e–01	–7,0175e–01	–1,0445e–01
	Najslabši	–1,2377e–04	–2,6770e–01	2,8440e–02	5,1975e–01	–7,0157e–01	–9,9840e–02
	Poprečje	–3,7345e–05[†]	–2,6059e–01[†]	4,8150e–02[‡]	5,2043e–01[‡]	–7,0235e–01[†]	–1,0427e–01[†]
	Std	–2,7992e–05	–4,3180e–03	–1,3010e–02	1,1536e–03	2,1270e–03	2,1507e–03

4.4.6 Algoritem MTS

MTS [110] je varianta lokalnega iskanja, ki po naključni inicializaciji s hevristiko in naključnim vzorčenjem izbira nove iskalne (hiper)podprostore. Iskanje poteka z agenti, ki so medsebojno neodvisni in med njimi ni komunikacije. Večkriterijska selekcija se izvaja po izdelavi vsake nove rešitve in v kolikor ta dominira katero od rešitev, nova rešitev v aproksimacijski množici zamenja najbližjo rešitev v evklidskem prostoru rešitev.

V tabelah 4.42–4.47 vidimo, da algoritem DEMOwSA preseže algoritem MTS na obeh indikatorjih za vse aproksimacijske množice pri funkcijah OKA2, WFG8 z $M = 3$ ali $M = 5$, WFG9 z $M = 3$ ali $M = 5$, S_DTLZ2 z $M = 5$ in R_DTLZ2 z $M = 5$. Algoritem MTS je boljši le na funkcijah S_ZDT2, S_ZDT4, R_DTLZ2 z $M = 3$ in WFG1 z $M = 3$ ali $M = 5$.

Ocene razlik so za oba indikatorja enake za funkcije SYM-PART, S_ZDT1 in S_ZDT6, kjer je po $5e+4$ ovrednotenjih in $5e+5$ ovrednotenjih boljši algoritem DEMOwSA ter za funkcijo S_DTLZ3 z $M = 3$, kjer je algoritem DEMOwSA boljši le po oceni dobljene aproksimacijske množice za $5e+4$ ovrednotenjih.

Za funkciji S_DTLZ2 z $M = 3$ ali $M = 5$ je po indikatorju I_R boljši algoritem DEMOwSA, po indikatorju $I_{\overline{H}}$ pa le za $5e+3$ ovrednotenjih in $5e+4$ ovrednotenjih oz. za $5e+4$ ovrednotenjih in $5e+5$ ovrednotenjih za $M = 3$ oz. $M = 5$.

Iz rezultatov lahko sklepamo, da ima algoritem DEMOwSA v povprečju za dano testno ogrodje večkrat bolje ocenjene dobljene aproksimacijske množice in za večino danih testnih funkcij prostor preiskuje učinkoviteje, glede na upoštevana indikatorja.

Tabela 4.42: Razlika vrednosti indikatorja I_R za algoritma MTS in DEMOwSA na testnih funkcijah 1–7.

<i>FEs</i>		1. OKA2	2. SYMPART	3. S_ZDT1	4. S_ZDT2	5. S_ZDT4	6. R_ZDT4	7. S_ZDT6
5e+3	Najboljši	-1,5032e-02	1,4677e-02	1,7090e-02	4,6453e-02	5,5225e-02	5,7240e-03	2,7790e-02
	Srednji	-2,1941e-02	2,6820e-02	2,2546e-02	5,3623e-02	5,7084e-02	1,0857e-02	3,0780e-02
	Najslabši	-2,7826e-02	4,2912e-02	2,7328e-02	7,7935e-02	5,4823e-02	8,0170e-03	2,9120e-02
	Poprečje	-2,0946e-02†	$2,5748e-02^\ddagger$	$2,3830e-02^\ddagger$	$5,3822e-02^\ddagger$	$5,6819e-02^\ddagger$	$7,5660e-03^\ddagger$	$2,9640e-02^\ddagger$
	Std	-3,0439e-03	6,3572e-03	2,4300e-03	4,9481e-03	3,3310e-04	1,0817e-03	-6,9000e-06
5e+4	Najboljši	-7,0833e-03	-3,9110e-04	-5,9420e-04	-1,6820e-03	4,1650e-03	-1,7968e-03	-3,1100e-02
	Srednji	-1,2265e-02	-4,7685e-04	-1,5293e-03	-1,3252e-02	1,0456e-02	-8,2584e-03	-1,7498e-02
	Najslabši	-1,5520e-02	-5,8323e-04	-3,2678e-03	1,1087e-02	1,6721e-02	-1,0360e-02	1,2193e-02
	Poprečje	-1,1973e-02†	-4,7180e-04†	-1,1770e-03†	$3,9080e-03^\ddagger$	$1,0815e-02^\ddagger$	-7,6109e-03†	-1,6370e-02†
	Std	-1,9669e-03	-5,2434e-05	-8,5939e-04	1,1624e-02	3,0765e-03	-2,1429e-03	1,3711e-02
5e+5	Najboljši	-3,3791e-03	-2,9987e-04	-1,3365e-04	-3,4060e-04	-1,6717e-05	-9,3825e-04	-1,2435e-02
	Srednji	-7,4449e-03	-3,9347e-04	-1,0031e-03	-3,3747e-03	1,8817e-04	-6,1218e-03	-1,3764e-02
	Najslabši	-1,6499e-02	-4,9050e-04	-3,7535e-03	3,0032e-02	3,5089e-04	-1,2263e-02	3,0492e-02
	Poprečje	-7,3886e-03†	-3,8572e-04†	-9,1506e-04†	$1,4425e-02^\ddagger$	$1,5545e-04^\ddagger$	-5,6458e-03†	-1,0109e-02†
	Std	-2,6120e-03	-4,5870e-05	-7,6689e-04	1,7348e-02	9,7910e-05	-3,1071e-03	1,1790e-02

Tabela 4.43: Razlika vrednosti indikatorja I_R za algoritma MTS in DEMOwSA na testnih funkcijah 8–13 z $M = 3$.

<i>FEs</i>		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Najboljši	-1,5509e-04	1,2553e-04	8,6134e-03	5,5934e-02	-1,3797e-02	-9,1082e-03
	Srednji	-3,8559e-04	1,4013e-03	3,6047e-04	5,6163e-02	-1,1653e-02	-9,1925e-03
	Najslabši	-6,8712e-04	3,3796e-03	-9,9267e-03	5,6261e-02	-8,2847e-03	-8,6552e-03
	Poprečje	-3,2967e-04†	$1,4745e-03^\ddagger$	$1,4399e-03^\ddagger$	$5,6147e-02^\ddagger$	-1,1243e-02†	-9,1049e-03†
	Std	-1,4841e-04	7,9501e-04	-3,7782e-03	9,0183e-05	1,3424e-03	-2,4300e-05
5e+4	Najboljši	-5,2125e-05	1,8137e-04	6,8645e-03	5,3464e-02	-2,4944e-02	-7,5843e-03
	Srednji	-3,3611e-04	6,9814e-04	-2,8047e-03	5,4532e-02	-2,2851e-02	-8,9618e-03
	Najslabši	-2,8866e-03	1,8411e-03	-1,1253e-02	5,4858e-02	-2,0667e-02	-1,1043e-02
	Poprečje	-1,8916e-04†	$8,0203e-04^\ddagger$	-3,1339e-03†	$5,4372e-02^\ddagger$	-2,2832e-02†	-8,6585e-03†
	Std	-5,5749e-04	5,0290e-04	-4,2471e-03	3,7768e-04	1,1062e-03	-1,1230e-03
5e+5	Najboljši	-4,8758e-05	1,8971e-04	1,4347e-02	3,7704e-02	-2,6671e-02	-4,8494e-03
	Srednji	-5,7900e-04	2,5045e-04	8,7058e-03	4,1243e-02	-2,5825e-02	-7,8233e-03
	Najslabši	-5,6692e-03	1,7375e-03	-1,9887e-03	4,3050e-02	-2,4265e-02	-1,1482e-02
	Poprečje	-1,6142e-04	$4,2984e-04^\ddagger$	$8,6417e-03^\ddagger$	$4,0869e-02^\ddagger$	-2,5818e-02†	-7,3637e-03†
	Std	-1,2425e-03	3,6449e-04	-4,1077e-03	1,5906e-03	6,0261e-04	-1,6168e-03

Tabela 4.44: Razlika vrednosti indikatorja I_R za algoritma MTS in DEMOwSA na testnih funkcijah 8–13 z $M = 5$.

<i>FEs</i>		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Najboljši	-1,4685e-04	-5,5237e-02	5,1008e-03	4,6941e-02	-1,6549e-02	3,4765e-03
	Srednji	-5,3110e-05	-5,7732e-02	-4,0717e-03	4,7135e-02	-1,3507e-02	3,7938e-03
	Najslabši	1,5602e-04	-5,8298e-02	-1,3256e-02	4,7294e-02	-1,1221e-02	5,6916e-03
	Poprečje	-2,0750e-05†	-5,8080e-02†	-4,0609e-03†	$4,7122e-02^\ddagger$	-1,3514e-02†	$3,9449e-03^\ddagger$
	Std	8,5294e-05	-8,2996e-04	-4,9901e-03	8,4435e-05	8,5365e-04	3,9640e-04
5e+4	Najboljši	-1,8238e-04	-4,8993e-02	1,1017e-02	4,6844e-02	-1,7016e-02	1,5079e-03
	Srednji	-2,3661e-04	-5,2307e-02	1,8956e-04	4,6989e-02	-1,5497e-02	-2,1624e-03
	Najslabši	-3,1427e-04	-5,4122e-02	-1,0064e-02	4,7193e-02	-1,3735e-02	-3,3928e-03
	Poprečje	-2,3111e-04†	-5,2529e-02†	$1,0682e-03^\ddagger$	$4,7003e-02^\ddagger$	-1,5535e-02†	-2,5603e-03†
	Std	-3,2696e-05	-1,4434e-03	-5,9443e-03	9,4243e-05	4,9848e-04	-1,5217e-03
5e+5	Najboljši	-1,3358e-04	-4,5396e-02	1,0144e-02	4,5491e-02	-1,9531e-02	7,7301e-04
	Srednji	-1,7209e-04	-4,7373e-02	3,5168e-03	4,5978e-02	-1,8042e-02	-1,1365e-03
	Najslabši	-2,0745e-04	-4,8439e-02	-4,9002e-03	4,6298e-02	-1,6679e-02	-5,1095e-03
	Poprečje	-1,7165e-04†	-4,7412e-02†	$4,4178e-03^\ddagger$	$4,5971e-02^\ddagger$	-1,8366e-02†	-1,0796e-03†
	Std	-1,8705e-05	-7,2980e-04	-3,7853e-03	2,1783e-04	7,3060e-04	-1,4517e-03

Tabela 4.45: Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma MTS in DEMOwSA na testnih funkcijah 1–7.

<i>FEs</i>	1. OKA2	2. SYMPART	3. S_ZDT1	4. S_ZDT2	5. S_ZDT4	6. R_ZDT4	7. S_ZDT6
5e+3	Najboljši –3,1875e–02	4,1884e–02	7,1447e–02	1,2680e–01	1,6497e–01	1,1112e–02	7,7340e–02
	Srednji –4,2833e–02	7,5787e–02	8,4355e–02	1,4071e–01	1,7718e–01	3,0539e–02	8,1710e–02
	Najslabši –5,0656e–02	1,1973e–01	1,0198e–01	1,5809e–01	1,7451e–01	2,4127e–02	7,9610e–02
	Poprečje –4,4433e–02†	7,2689e–02‡	8,4622e–02‡	1,4295e–01‡	1,7459e–01‡	2,2321e–02‡	7,7910e–02‡
	Std –4,6264e–03	1,7556e–02	8,7636e–03	1,1778e–02	2,6920e–03	1,5680e–03	–9,7700e–05
5e+4	Najboljši –6,8690e–03	–1,1475e–03	–1,9775e–03	–3,7639e–03	1,2515e–02	–5,3987e–03	–7,7559e–02
	Srednji –1,1081e–02	–1,3962e–03	–3,5521e–03	–1,0346e–02	3,0858e–02	–2,5074e–02	–4,8090e–02
	Najslabši –3,1060e–03	–1,7077e–03	–5,6090e–03	2,3282e–02	4,9180e–02	–3,1338e–02	2,8961e–02
	Poprečje –1,0390e–02†	–1,3818e–03†	–3,3252e–03†	1,1135e–02‡	3,2054e–02‡	–2,3668e–02†	–3,8995e–02†
	Std 1,3197e–03	–1,5292e–04	–1,1942e–03	1,7986e–02	9,0403e–03	–5,9616e–03	3,2466e–02
5e+5	Najboljši –5,5770e–03	–8,8283e–04	–7,0782e–04	–8,1081e–04	–5,1169e–05	–3,4472e–03	–2,7526e–02
	Srednji –7,7654e–03	–1,1554e–03	–1,9736e–03	–2,3724e–03	1,1169e–03	–1,8655e–02	–3,0836e–02
	Najslabši –2,9900e–03	–1,4384e–03	–4,7178e–03	4,2900e–02	1,5936e–03	–3,7662e–02	6,9345e–02
	Poprečje –4,9451e–03†	–1,1329e–03†	–1,5432e–03†	1,8904e–02‡	8,8695e–04‡	–1,8505e–02†	–2,3001e–02†
	Std 2,3316e–03	–1,3368e–04	–1,0695e–03	2,3054e–02	5,1041e–04	–9,1924e–03	2,6619e–02

Tabela 4.46: Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma MTS in DEMOwSA na testnih funkcijah 8–13 z $M = 3$.

<i>FEs</i>	8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Najboljši 8,0800e–05	1,4508e–02	4,4493e–02	2,8601e–01	–9,7731e–02	–1,0738e–01
	Srednji –1,4289e–03	2,0812e–02	1,0940e–02	2,8432e–01	–8,6113e–02	–1,4083e–01
	Najslabši –3,8717e–03	3,0450e–02	–2,4988e–02	2,7882e–01	–7,5433e–02	–1,8552e–01
	Poprečje –1,3154e–03†	2,1290e–02‡	1,2635e–02‡	2,8409e–01‡	–8,5515e–02†	–1,4273e–01†
	Std –8,3634e–04	4,1178e–03	–1,2667e–02	–1,3474e–03	5,0577e–03	–2,2122e–02
5e+4	Najboljši –3,1315e–04	1,1249e–02	3,4040e–02	2,7225e–01	–1,5769e–01	–1,3815e–01
	Srednji –4,2799e–03	1,5027e–02	–1,0332e–02	2,7567e–01	–1,4565e–01	–2,0494e–01
	Najslabši –2,4403e–02	2,0650e–02	–3,2321e–02	2,7416e–01	–1,3867e–01	–2,7710e–01
	Poprečje –3,2986e–03†	1,5940e–02‡	–1,3105e–02†	2,7504e–01‡	–1,4701e–01†	–2,0260e–01†
	Std –4,9209e–03	2,8320e–03	–1,5250e–02	6,2870e–04	4,7856e–03	–3,8044e–02
5e+5	Najboljši –2,4433e–04	9,4423e–03	8,1406e–02	1,9480e–01	–1,6428e–01	–1,6039e–02
	Srednji –7,3964e–03	1,1537e–02	5,1440e–02	2,1085e–01	–1,6045e–01	–2,2083e–01
	Najslabši –3,6926e–02	1,9264e–02	–1,9666e–02	2,1725e–01	–1,5361e–01	–2,9575e–01
	Poprečje –4,7908e–03†	1,2050e–02‡	5,2438e–02‡	2,0918e–01‡	–1,6011e–01†	–2,4415e–01†
	Std –8,0806e–03	2,3458e–03	–2,2803e–02	6,8946e–03	2,6256e–03	–6,5826e–02

Tabela 4.47: Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma MTS in DEMOwSA na testnih funkcijah 8–13 z $M = 5$.

<i>FEs</i>	8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Najboljši 5,2096e–03	–2,8287e–01	–1,5753e–03	5,2949e–01	–5,6741e–01	–7,5156e–02
	Srednji 9,6086e–03	–2,8875e–01	–3,6336e–02	5,3046e–01	–5,6008e–01	–9,3600e–02
	Najslabši 1,2869e–02	–2,8466e–01	–7,1495e–02	5,3175e–01	–5,5438e–01	–1,0827e–01
	Poprečje 9,7885e–03‡	–2,8976e–01†	–3,6154e–02†	5,3035e–01‡	–5,6198e–01†	–9,5030e–02†
	Std 1,8026e–03	2,0800e–04	–1,8457e–02	6,2342e–04	2,2545e–03	–1,0719e–02
5e+4	Najboljši –9,9718e–04	–2,5630e–01	3,7128e–02	5,2837e–01	–6,1633e–01	–1,6741e–01
	Srednji –2,1629e–03	–2,6609e–01	–1,6863e–02	5,2992e–01	–6,2155e–01	–3,0535e–01
	Najslabši –3,4961e–03	–2,7488e–01	–8,1795e–02	5,3245e–01	–6,1107e–01	–3,8356e–01
	Poprečje –2,1665e–03†	–2,6561e–01†	–1,5930e–02†	5,3001e–01‡	–6,2442e–01†	–3,0758e–01†
	Std –6,1225e–04	–4,8245e–03	–3,0326e–02	9,1192e–04	–6,1880e–04	–5,2157e–02
5e+5	Najboljši –5,4418e–04	–2,4246e–01	5,6373e–02	5,1648e–01	–6,4210e–01	–1,1071e–01
	Srednji –9,5900e–04	–2,5154e–01	1,3468e–02	5,2133e–01	–6,4291e–01	–1,8250e–01
	Najslabši –1,5973e–03	–2,5456e–01	–3,9831e–02	5,2368e–01	–6,3318e–01	–2,7204e–01
	Poprečje –9,1789e–04†	–2,5338e–01†	1,2928e–02‡	5,2096e–01‡	–6,4333e–01†	–1,9000e–01†
	Std –2,7320e–04	–3,9714e–03	–2,2859e–02	2,0049e–03	4,3480e–03	–4,4681e–02



4.4.7 Algoritem MO_{DE}

MO_{DE} [111] Zielinskijeve [41] in sodelavca uporablja metriko nakopičenosti [79] za večkriterijsko selekcijo in diferencialno evolucijo. Pri slednji samoprilagaja krmilna parametra F in CR , tako da z analizo variance vzporedno za štiri njune nove kombinacije fiksnih sprememb primerja kakovost rešitev vsake od četrte populacije.

Podajmo krajšo diskusijo o tabelah 4.48–4.53, kjer vidimo razliko in statistično primerjavo rezultatov algoritmov MO_{DE} in DEMOwSA.

Na obeh indikatorjih za vse aproksimacijske množice pri funkcijah S_ZDT1, WFG1 z $M = 3$ ali $M = 5$, S_DTLZ2 z $M = 5$ in R_DTLZ2 z $M = 5$ algoritem DEMOwSA preseže algoritem MO_{DE}. MO_{DE} je boljši od DEMOwSA na funkcijah S_ZDT6, S_DTLZ3 z $M = 3$ in WFG1 z $M = 3$ ali $M = 5$.

Oba indikatorja dajeta enake rezultate na funkcijah SYM-PART, S_ZDT4, ter R_ZDT4 in S_DTLZ3 z $M = 5$, kjer na slednji funkciji daje DEMOwSA boljše rezultate na začetku, le pri $5e+3$ ovrednotenjih, pri prvih treh pa proti koncu, po $5e+4$ ovrednotenjih in $5e+5$ ovrednotenjih. Na funkcijah S_ZDT6, S_DTLZ3 z $M = 3$ in WFG1 z $M = 3$ ali $M = 5$ daje na obeh indikatorjih boljše rezultate algoritem MO_{DE}.

Pri funkciji OKA2 algoritem DEMOwSA daje boljše rezultate na indikatorju I_R za vse tri aproksimacijske množice, na indikatorju $I_{\overline{H}}$ pa pri $5e+5$ ovrednotenjih ni opaziti signifikantnih razlik. Tudi pri funkciji S_ZDT2 ni signifikantnih razlik, na obeh indikatorjih. Pri $5e+5$ ovrednotenjih in $5e+3$ ovrednotenjih daje boljše rezultate MO_{DE}, pri $5e+4$ ovrednotenjih je na indikatorju $I_{\overline{H}}$ boljši DEMOwSA, na indikatorju I_R , MO_{DE}. Pri funkciji R_DTLZ2 z $M = 3$ na indikatorju I_R daje boljše rezultate algoritem MO_{DE}, na indikatorju $I_{\overline{H}}$ pri $5e+3$ ovrednotenjih daje boljše rezultate algoritem DEMOwSA, pri $5e+4$ ovrednotenjih ni signifikantnih razlik na slednjem indikatorju, pri $5e+5$ ovrednotenjih v povprečju daje boljše rezultate algoritem MO_{DE}. Funkcijo WFG9 z $M = 5$ indikator I_R bolje oceni pri algoritmu MO_{DE}, indikator $I_{\overline{H}}$ le pri $5e+3$ ovrednotenjih, a pri $5e+4$ ovrednotenjih in $5e+5$ ovrednotenjih je bolje ocenjen algoritem DEMOwSA.

Rezultati kažejo, da algoritem DEMOwSA na izbranem testnem ogrodju večkrat daje signifikantno boljše rezultate od algoritma MO_{DE}.

Tabela 4.48: Razlika vrednosti indikatorja I_R za algoritma MO_DE in DEMOWSA na testnih funkcijah 1–7.

<i>FEs</i>		1. OKA2	2. SYMPART	3. S_ZDT1	4. S_ZDT2	5. S_ZDT4	6. R_ZDT4	7. S_ZDT6
5e+3	Najboljši	-1,0798e-02	1,0262e-02	-6,5130e-03	1,0522e-02	1,2775e-02	3,2120e-03	5,0100e-03
	Srednji	-1,6970e-02	1,5720e-02	-6,2130e-03	3,9300e-03	1,3468e-02	2,8310e-03	3,8300e-03
	Najslabši	-2,2092e-02	1,5830e-02	-5,3220e-03	1,4790e-02	4,5200e-04	-2,1490e-03	4,3900e-03
	Poprečje	-1,6838e-02†	1,3486e-02‡	-5,2940e-03†	5,6800e-03‡	1,1760e-02‡	1,3940e-03‡	4,2700e-03‡
	Std	-2,9631e-03	-7,9340e-04	-1,2701e-03	-2,9650e-04	-3,2174e-03	-4,7870e-04	-3,2480e-04
5e+4	Najboljši	-2,9071e-03	5,3494e-06	7,1570e-06	3,1987e-05	-3,4320e-03	2,0747e-04	-8,9040e-04
	Srednji	-5,2656e-03	-1,1826e-05	-1,0757e-05	4,8270e-04	-9,1400e-04	-1,6261e-03	1,3314e-02
	Najslabši	-1,1161e-02	-4,9717e-04	-3,9541e-03	4,0341e-02	-1,6594e-02	-7,9732e-03	1,9796e-02
	Poprečje	-6,1711e-03†	-4,5033e-05†	-5,7186e-04†	1,7768e-02‡	-1,7900e-03†	-3,6298e-03†	1,0159e-02
	Std	-2,6101e-03	-1,0670e-04	-1,1830e-03	2,0244e-02	-2,8149e-03	-2,8172e-03	-9,0269e+00
5e+5	Najboljši	-1,0938e-03	7,5721e-07	1,1774e-08	7,3772e-08	-1,8708e-09	-3,7000e-07	1,0216e+00
	Srednji	-2,8098e-03	-5,6600e-08	-9,1198e-07	-6,3255e-04	9,0990e-05	-6,4413e-04	-4,1489e-05
	Najslabši	-1,0305e-02	-6,1855e-05	-4,5766e-05	1,3400e-04	-3,6973e-03	-2,6943e-03	1,2769e-02
	Poprečje	-3,9595e-03†	-2,3948e-06†	-1,3523e-05†	-1,9900e-04	-3,5630e-04†	-6,8242e-04†	1,4949e-03
	Std	-2,5514e-03	-1,2313e-05	-1,9399e-05	2,0100e-04	-9,3680e-04	-5,6081e-04	5,1718e-03

Tabela 4.49: Razlika vrednosti indikatorja I_R za algoritma MO_DE in DEMOWSA na testnih funkcijah 8–13 z $M = 3$.

<i>FEs</i>		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Najboljši	1,6005e-05	-1,5804e-04	7,8178e-03	5,5852e-02	-1,3954e-02	-1,2278e-02
	Srednji	-1,6000e-06	1,2917e-03	5,0748e-03	5,6043e-02	-1,1830e-02	-1,4064e-02
	Najslabši	-5,6630e-05	3,6038e-03	2,6316e-03	5,6159e-02	-8,4600e-03	-1,3133e-02
	Poprečje	4,1400e-06	1,3337e-03‡	5,2159e-03‡	5,6027e-02‡	-1,1427e-02†	-1,3580e-02†
	Std	-4,5890e-06	8,8868e-04	-1,4995e-03	8,5989e-05	1,3424e-03	-5,5700e-04
5e+4	Najboljši	-2,1196e-05	-1,0807e-04	1,9740e-02	5,3424e-02	-2,5052e-02	-5,5549e-03
	Srednji	-4,0346e-05	3,6006e-04	1,4371e-02	5,4516e-02	-2,2983e-02	-5,8251e-03
	Najslabši	-4,0546e-03	1,5207e-03	1,0325e-02	5,4933e-02	-2,0826e-02	-4,7606e-03
	Poprečje	-3,8865e-05†	4,5360e-04‡	1,4538e-02‡	5,4349e-02‡	-2,2962e-02†	-5,7244e-03†
	Std	-8,1670e-06	4,9414e-04	-2,5989e-03	3,9793e-04	1,0915e-03	-1,1760e-05
5e+5	Najboljši	-1,3789e-05	1,4835e-04	2,5040e-02	3,7659e-02	-2,6797e-02	-1,0286e-03
	Srednji	-4,4891e-05	1,0267e-04	2,3416e-02	4,1218e-02	-2,5986e-02	-1,2938e-03
	Najslabši	-7,6338e-05	1,5061e-03	2,1223e-02	4,3059e-02	-2,4453e-02	-1,2667e-03
	Poprečje	-4,6575e-05†	2,5453e-04‡	2,3333e-02‡	4,0840e-02‡	-2,5976e-02†	-1,3542e-03†
	Std	-1,7906e-05	3,1432e-04	-8,8206e-04	1,6035e-03	5,8568e-04	6,3120e-05

Tabela 4.50: Razlika vrednosti indikatorja I_R za algoritma MO_DE in DEMOWSA na testnih funkcijah 8–13 z $M = 5$.

<i>FEs</i>		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Najboljši	-2,9698e-04	-6,8925e-02	5,2369e-03	4,6827e-02	-4,0811e-02	1,3411e-03
	Srednji	-2,4105e-04	-7,7008e-02	-2,4567e-03	4,6997e-02	-4,0895e-02	3,3215e-03
	Najslabši	-6,5600e-05	-7,8959e-02	-8,4883e-03	4,7121e-02	-4,4312e-02	4,7670e-03
	Poprečje	-2,1325e-04†	-7,6770e-02†	-2,3536e-03†	4,6983e-02‡	-4,2454e-02†	2,8911e-03‡
	Std	6,6364e-05	-1,7525e-03	-3,4618e-03	6,8584e-05	-2,1284e-03	5,9760e-04
5e+4	Najboljši	-2,4738e-04	-5,6261e-02	9,5706e-03	4,6772e-02	-4,2817e-02	5,8500e-05
	Srednji	-2,9563e-04	-5,6966e-02	7,9953e-03	4,6893e-02	-4,0817e-02	4,1510e-04
	Najslabši	-4,5813e-04	-5,8129e-02	3,2708e-03	4,7081e-02	-3,8745e-02	1,0348e-03
	Poprečje	-3,0879e-04†	-5,7080e-02†	7,0863e-03‡	4,6908e-02‡	-4,0727e-02†	4,8620e-04‡
	Std	-4,6423e-05	-5,5754e-04	-1,9269e-03	8,6434e-05	6,9476e-04	1,9023e-04
5e+5	Najboljši	-2,0396e-04	-5,0832e-02	1,3569e-02	4,5426e-02	-4,6446e-02	6,0295e-04
	Srednji	-2,5533e-04	-5,4291e-02	1,2911e-02	4,5907e-02	-4,4488e-02	1,5261e-03
	Najslabši	-4,8684e-04	-5,5582e-02	8,9052e-03	4,6214e-02	-4,2995e-02	2,1076e-03
	Poprečje	-2,6047e-04†	-5,4145e-02†	1,1981e-02‡	4,5901e-02‡	-4,4732e-02†	1,4374e-03‡
	Std	-5,1967e-05	-1,1021e-03	-1,6910e-03	2,1473e-04	8,4744e-04	3,7479e-04

Tabela 4.51: Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma MO_DE in DEMOWSA na testnih funkcijah 1–7.

<i>FEs</i>		1. OKA2	2. SYMPART	3. S_ZDT1	4. S_ZDT2	5. S_ZDT4	6. R_ZDT4	7. S_ZDT6
5e+3	Najboljši	-2,1854e-02	2,9214e-02	2,2500e-03	4,6050e-02	3,9910e-02	-3,7320e-03	2,0960e-02
	Srednji	-3,3088e-02	4,4138e-02	-8,1900e-03	1,1320e-02	4,4920e-02	9,2040e-03	1,0500e-02
	Najslabši	-4,6426e-02	4,3420e-02	-2,0540e-02	-2,5200e-02	3,4500e-03	-5,8700e-03	1,2460e-02
	Poprečje	-3,3687e-02[†]	3,7855e-02[‡]	-7,1000e-03[†]	1,3960e-02[‡]	3,7540e-02[‡]	4,4850e-03[‡]	1,2150e-02[‡]
	Std	-4,5318e-03	-2,5840e-03	-6,1250e-03	-8,6140e-03	-9,6840e-03	-9,7000e-04	-1,7390e-03
5e+4	Najboljši	-1,7030e-03	1,6552e-05	3,7950e-05	2,8737e-04	-9,9840e-03	4,7300e-04	-2,0372e-03
	Srednji	-2,6670e-03	-3,6435e-05	-1,0240e-05	-2,7448e-01	-2,6720e-03	-5,0241e-03	2,4717e-02
	Najslabši	4,0320e-03	-1,4402e-03	-1,3572e-02	-4,1631e-01	-4,9219e-02	-2,4335e-02	4,4378e-02
	Poprečje	-1,9960e-03[†]	-1,3259e-04[†]	-2,0509e-03[†]	-1,7613e-01[†]	-5,2590e-03[†]	-1,0772e-02[†]	2,3407e-02[‡]
	Std	5,1830e-04	-3,0919e-04	-4,3134e-03	-1,6364e-01	-8,3956e-03	-8,2236e-03	1,5471e-02
5e+5	Najboljši	-9,6880e-04	2,3007e-06	-2,8400e-06	8,0300e-06	5,7270e-08	-5,9000e-07	1,0000e-06
	Srednji	-1,0543e-03	-1,5480e-07	-7,5400e-06	-1,3342e-04	9,5712e-04	-1,6038e-03	-9,2040e-05
	Najslabši	8,5580e-03	-1,7970e-04	-1,9034e-04	4,0500e-04	-1,0722e-02	-7,7825e-03	2,7950e-02
	Poprečje	4,0900e-05	-6,8750e-06[†]	-5,5950e-05[†]	-1,9000e-05	-6,3750e-04[†]	-1,8835e-03[†]	3,3474e-03
	Std	3,0472e-03	-3,5784e-05	-8,0112e-05	8,4000e-05	-2,6986e-03	-1,6358e-03	1,1542e-02

Tabela 4.52: Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma MO_DE in DEMOWSA na testnih funkcijah 8–13 z $M = 3$.

<i>FEs</i>		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Najboljši	1,0230e-03	-3,5620e-03	6,0857e-02	2,8524e-01	-1,0333e-01	-9,2260e-03
	Srednji	1,2025e-03	-2,1050e-03	5,0407e-02	2,8514e-01	-9,3034e-02	-3,7352e-02
	Najslabši	-1,6636e-03	3,1860e-03	4,0495e-02	2,8493e-01	-8,3430e-02	-4,8921e-02
	Poprečje	9,7980e-04[†]	-2,1740e-03[†]	4,9560e-02[‡]	2,8504e-01[‡]	-9,2945e-02[†]	-3,5013e-02[†]
	Std	-4,9924e-04	1,3356e-03	-4,2372e-03	-1,2332e-04	4,5318e-03	-9,3606e-03
5e+4	Najboljši	-5,1311e-05	-9,3500e-04	1,1145e-01	2,7294e-01	-1,5955e-01	1,3113e-01
	Srednji	-1,5870e-04	-6,0400e-04	8,9217e-02	2,7789e-01	-1,4855e-01	9,5850e-02
	Najslabši	-3,1026e-04	5,4000e-05	6,7399e-02	2,7990e-01	-1,4421e-01	7,6543e-02
	Poprečje	-1,7466e-04[†]	-2,1000e-05	8,8634e-02[‡]	2,7724e-01[‡]	-1,5021e-01[†]	9,7561e-02[‡]
	Std	-5,9341e-05	9,3620e-04	-1,2033e-02	1,8364e-03	3,8444e-03	-1,1526e-02
5e+5	Najboljši	-7,3272e-05	9,4219e-03	1,4308e-01	1,9503e-01	-1,6963e-01	1,5554e-01
	Srednji	-1,8785e-04	1,0231e-02	1,2859e-01	2,1213e-01	-1,6984e-01	1,1610e-01
	Najslabši	-7,7472e-04	8,0600e-03	1,1698e-01	2,2031e-01	-1,6728e-01	5,9112e-02
	Poprečje	-2,1665e-04[†]	9,1326e-03[‡]	1,2888e-01[‡]	2,1033e-01[‡]	-1,6968e-01[†]	1,1390e-01[‡]
	Std	-1,4650e-04	-6,8080e-04	-5,5865e-03	7,7307e-03	1,9820e-04	-1,9378e-02

Tabela 4.53: Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma MO_DE in DEMOWSA na testnih funkcijah 8–13 z $M = 5$.

<i>FEs</i>		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Najboljši	-1,1397e-02	-3,3633e-01	2,2088e-02	5,2467e-01	-5,9511e-01	2,7486e-02
	Srednji	-1,4032e-02	-3,7348e-01	-1,3578e-02	5,2433e-01	-6,0325e-01	2,6040e-02
	Najslabši	-1,9159e-02	-3,7661e-01	-3,8041e-02	5,2450e-01	-6,5167e-01	1,8430e-02
	Poprečje	-1,3344e-02[†]	-3,7160e-01[†]	-1,2745e-02[‡]	5,2427e-01[‡]	-6,2068e-01[†]	2,2690e-02[‡]
	Std	-2,0875e-03	-5,8562e-03	-1,5760e-02	-1,0222e-04	-2,4797e-02	-2,7600e-03
5e+4	Najboljši	-6,0887e-03	-2,7858e-01	5,7234e-02	5,2552e-01	-6,7124e-01	-3,3455e-02
	Srednji	-7,1248e-03	-2,8143e-01	4,2150e-02	5,2613e-01	-6,6124e-01	-6,0047e-02
	Najslabši	-2,2506e-02	-2,8229e-01	1,3503e-02	5,2744e-01	-6,4084e-01	-6,3426e-02
	Poprečje	-8,5861e-03[†]	-2,8167e-01[†]	3,7480e-02[‡]	5,2622e-01[‡]	-6,6148e-01[†]	-5,7544e-02[†]
	Std	-3,7354e-03	-1,2744e-03	-1,3224e-02	3,2363e-04	5,7474e-03	-6,6755e-03
5e+5	Najboljši	-2,3166e-03	-2,5443e-01	7,6954e-02	5,1432e-01	-7,2316e-01	-3,0448e-02
	Srednji	-3,6936e-03	-2,6969e-01	6,9728e-02	5,1853e-01	-7,1079e-01	-4,2614e-02
	Najslabši	-1,7888e-02	-2,7336e-01	4,9974e-02	5,1890e-01	-6,9214e-01	-5,3547e-02
	Poprečje	-4,2231e-03[†]	-2,6896e-01[†]	6,7370e-02[‡]	5,1807e-01[‡]	-7,0988e-01[†]	-4,2869e-02[†]
	Std	-2,9243e-03	-4,5436e-03	-8,7839e-03	1,3308e-03	8,9280e-03	-7,1983e-03

4.4.8 Algoritem MO_PSO

MO_PSO [40] Zielinskijeve in Laura vsebuje enak princip prilagajanja krmilnih parametrov kot algoritem MO_DE, a je tu uporabljen algoritem za optimizacijo z rojem delcev [112]. Izbrana je večkriterijska selekcija nakopičenosti [79].

V tabelah 4.54–4.59 vidimo razliko in statistično primerjavo rezultatov, oceno aproksimacijskih množic, algoritmov MO_PSO in DEMOwSA.

Oba indikatorja, I_R in $I_{\bar{H}}$, ob vseh treh aproksimacijskih množicah, signifikantno bolje ocenita algoritem DEMOwSA na funkcijah OKA2, S_ZDT1, S_ZDT2, WFG8 z $M = 3$ ali $M = 5$, R_DTLZ2 z $M = 5$ in WFG9 z $M = 5$. Le na funkcijah S_DTLZ3 z $M = 3$ in WFG1 z $M = 3$ ali $M = 5$ je vedno bolje ocenjen algoritem MO_PSO. Na funkcijah S_ZDT4, R_ZDT4, S_ZDT6 in S_DTLZ2 z $M = 5$ oba indikatorja ocenita enako, da v začetku pri 5e+3 ovrednotenjih daje signifikantno boljše rezultate algoritem MO_PSO, po 5e+4 ovrednotenjih in 5e+5 ovrednotenjih pa algoritem DEMOwSA. Tudi pri funkcijah SYM-PART in S_DTLZ3 pri $M = 5$ sta oceni indikatorjev enaki, po dvakrat je boljši MO_PSO in po enkrat DEMOwSA, ki je boljši pri funkciji SYM-PART pri 5e+4 ovrednotenjih ter pri slednji funkciji pri 5e+3 ovrednotenjih.

Pri treh funkcijah, S_DTLZ2 z $M = 3$, R_DTLZ2 z $M = 3$ in WFG9 z $M = 3$, se oceni indikatorjev istih aproksimacijskih množic razlikujeta. Za prvo, S_DTLZ2 z $M = 3$, indikator $I_{\bar{H}}$ pri 5e+3 ovrednotenjih sicer bolje oceni algoritem MO_PSO, a pri ostalih dveh aproksimacijskih množicah oceni bolje algoritem DEMOwSA, indikator I_R pa vedno bolje oceni algoritem DEMOwSA. Pri funkciji R_DTLZ2 z $M = 3$ obratno, indikator $I_{\bar{H}}$, vselej bolje oceni algoritem DEMOwSA, indikator I_R pa le po 5e+5 ovrednotenjih. Pri funkciji WFG9 z $M = 3$ je tokrat na indikatorju I_R vselej bolje ocenjen algoritem DEMOwSA, na indikatorju $I_{\bar{H}}$ pa je DEMOwSA boljši le pri 5e+3 ovrednotenjih.

Rezultati nakazujejo, da algoritem DEMOwSA večkrat daje signifikantno boljše rezultate na izbranem testnem ogrodju.

Tabela 4.54: Razlika vrednosti indikatorja I_R za algoritma MO_PSO in DEMOWSA na testnih funkcijah 1–7.

<i>FEs</i>		1. OKA2	2. SYMPART	3. S_ZDT1	4. S_ZDT2	5. S_ZDT4	6. R_ZDT4	7. S_ZDT6
5e+3	Najboljši	-2,2383e-02	1,7615e-02	-1,5636e-02	-9,0840e-03	2,3760e-02	8,7486e-03	-2,2600e-03
	Srednji	-3,0524e-02	3,2134e-02	-2,1509e-02	-1,3110e-02	1,7911e-02	1,4908e-02	1,3500e-03
	Najslabši	-3,4905e-02	4,9935e-02	-2,8026e-02	-2,9000e-04	1,2350e-02	1,1487e-02	-4,3000e-04
	Poprečje	-2,9220e-02[†]	3,1301e-02[‡]	-2,0970e-02[†]	-1,0440e-02[†]	1,8864e-02[‡]	1,2913e-02[‡]	7,4000e-04[‡]
	Std	-2,8788e-03	7,5534e-03	-3,7595e-03	5,0000e-05	-4,4733e-03	8,0000e-04	4,3070e-04
5e+4	Najboljši	-1,1354e-02	4,0770e-07	-1,1486e-02	-6,8694e-02	-2,3315e-02	-1,7462e-03	-1,0431e-01
	Srednji	-3,1889e-02	-3,9051e-05	-1,8845e-02	-9,3514e-02	-2,9916e-02	-5,7852e-03	-1,0223e-01
	Najslabši	-3,0245e-02	-1,4928e-04	-3,0980e-02	-8,7123e-02	-3,7528e-02	-6,3372e-03	-7,6833e-02
	Poprečje	-2,8441e-02[†]	-6,0023e-05[†]	-1,9267e-02[†]	-7,8637e-02[†]	-2,9388e-02[†]	-5,1455e-03[†]	-1,0077e-01[†]
	Std	-5,4579e-03	-4,8292e-05	-5,0631e-03	4,6630e-03	-4,1165e-03	-1,0867e-03	9,8885e-03
5e+5	Najboljši	-6,3418e-03	1,1737e-06	-1,9191e-03	-1,9461e-02	-1,3852e-02	-1,9859e-03	-8,9468e-02
	Srednji	-3,2207e-02	1,4065e-06	-3,5424e-03	-6,6369e-02	-2,2046e-02	-5,9301e-03	-1,0312e-01
	Najslabši	-3,1416e-02	1,8568e-06	-1,1276e-02	-5,0134e-02	-4,1784e-02	-1,0763e-02	-7,4840e-02
	Poprečje	-2,6131e-02[†]	1,4334e-06[‡]	-4,6126e-03[†]	-4,7376e-02[†]	-2,4700e-02[†]	-5,9855e-03[†]	-9,9638e-02[‡]
	Std	-8,5518e-03	1,4595e-07	-2,3905e-03	3,9310e-03	-7,7813e-03	-2,2439e-03	5,5324e-03

Tabela 4.55: Razlika vrednosti indikatorja I_R za algoritma MO_PSO in DEMOWSA na testnih funkcijah 8–13 z $M = 3$.

<i>FEs</i>		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Najboljši	-8,1255e-05	-3,7831e-04	6,6651e-03	5,5771e-02	-1,4019e-02	-1,8821e-02
	Srednji	-1,1114e-04	9,6366e-04	3,3317e-03	5,6018e-02	-4,0116e+00	-6,4304e-02
	Najslabši	-4,2101e-03	3,2082e-03	5,5520e-04	5,6127e-02	-8,6049e-03	-1,8638e-02
	Poprečje	-1,1135e-04[†]	1,0075e-03[‡]	3,3929e-03[‡]	5,5986e-02[‡]	-1,1563e-02[†]	-1,9182e-02[†]
	Std	-2,0397e-05	8,5285e-04	-1,6677e-03	9,8547e-05	1,3199e-03	-2,9380e-04
5e+4	Najboljši	-6,9036e-05	-3,3625e-04	1,6156e-02	5,3388e-02	-2,5175e-02	-1,1326e-02
	Srednji	-8,5948e-05	1,3403e-04	1,3212e-02	5,4490e-02	-2,3123e-02	-1,2375e-02
	Najslabši	-8,9055e-05	1,2436e-03	1,0562e-02	5,4895e-02	-2,0968e-02	-1,2239e-02
	Poprečje	-8,2331e-05[†]	2,2666e-04[‡]	1,3131e-02[‡]	5,4318e-02[‡]	-2,3099e-02[†]	-1,2450e-02[†]
	Std	-4,1670e-06	4,8069e-04	-1,4361e-03	3,9883e-04	1,0881e-03	-3,5855e-04
5e+5	Najboljši	-5,9598e-05	-2,5724e-04	1,9649e-02	3,7650e-02	-2,6865e-02	-4,2948e-03
	Srednji	-7,7304e-05	-2,5514e-04	1,6649e-02	4,1192e-02	-2,6059e-02	-4,3515e-03
	Najslabši	-9,2288e-05	1,2731e-03	1,4954e-02	4,3048e-02	-2,4532e-02	-4,7483e-03
	Poprečje	-7,8640e-05[†]	-7,9060e-05[†]	1,6849e-02[‡]	4,0816e-02[‡]	-2,6054e-02[†]	-4,4522e-03[†]
	Std	-7,7179e-06	3,6011e-04	-1,2547e-03	1,6012e-03	5,8418e-04	-5,6790e-05

Tabela 4.56: Razlika vrednosti indikatorja I_R za algoritma MO_PSO in DEMOWSA na testnih funkcijah 8–13 z $M = 5$.

<i>FEs</i>		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Najboljši	-1,3158e-04	-7,8857e-02	4,7981e-03	4,6917e-02	-4,5295e-02	-2,1204e-03
	Srednji	-2,5310e-05	-8,1244e-02	-1,5755e-03	4,7102e-02	8,8476e-03	-1,8870e-03
	Najslabši	1,2891e-04	-8,2886e-02	-8,3339e-03	4,7240e-02	-4,6170e-02	-2,7250e-03
	Poprečje	-5,6300e-06	-8,1008e-02[†]	-1,9430e-03[†]	4,7088e-02[‡]	-4,7577e-02[†]	-3,7160e-03[†]
	Std	7,6407e-05	-8,9666e-04	-3,7586e-03	7,6389e-05	-1,9800e-04	-2,9230e-04
5e+4	Najboljši	-2,7011e-04	-7,4105e-02	9,6270e-03	4,6811e-02	-4,8513e-02	-1,7004e-03
	Srednji	-3,0020e-04	-7,6727e-02	6,9899e-03	4,6949e-02	-4,9234e-02	-2,0025e-03
	Najslabši	-3,3525e-04	-7,8696e-02	3,3871e-03	4,7154e-02	-4,8470e-02	-2,7368e-03
	Poprečje	-2,9948e-04[†]	-7,6665e-02[†]	6,8058e-03[‡]	4,6966e-02[‡]	-4,9210e-02[†]	-2,0425e-03[†]
	Std	-1,9648e-05	-1,1844e-03	-1,4478e-03	9,2471e-05	1,9880e-05	-2,4558e-04
5e+5	Najboljši	-2,5233e-04	-6,9999e-02	1,1705e-02	4,5426e-02	-5,2677e-02	-4,5890e-04
	Srednji	-2,8870e-04	-7,4374e-02	1,0459e-02	4,5899e-02	-5,2306e-02	-1,8020e-04
	Najslabši	-3,4000e-04	-7,5553e-02	8,5009e-03	4,6213e-02	-5,0870e-02	-2,6530e-04
	Poprečje	-2,9612e-04[†]	-7,4047e-02[†]	1,0383e-02[‡]	4,5895e-02[‡]	-5,2429e-02[†]	-2,0880e-04[†]
	Std	-2,3008e-05	-1,3076e-03	-9,5164e-04	2,1390e-04	5,2252e-04	1,4516e-04

Tabela 4.57: Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma MO_PSO in DEMOWSA na testnih funkcijah 1–7.

<i>FEs</i>		1. OKA2	2. SYMPART	3. S_ZDT1	4. S_ZDT2	5. S_ZDT4	6. R_ZDT4	7. S_ZDT6
5e+3	Najboljši	-3,6315e-02	5,0380e-02	-1,4710e-02	-1,0990e-02	7,3860e-02	2,5973e-02	-2,4000e-03
	Srednji	-3,1658e-02	9,1102e-02	-2,6180e-02	-4,3890e-02	5,9400e-02	4,5704e-02	4,6000e-03
	Najslabši	-5,0088e-02	1,3992e-01	-3,9900e-02	-7,3260e-02	4,3050e-02	3,4754e-02	-1,4800e-03
	Poprečje	-3,4773e-02[†]	8,8687e-02[‡]	-2,4800e-02[†]	-3,5010e-02[†]	6,0210e-02[‡]	4,1563e-02[‡]	2,3500e-03[‡]
	Std	-3,2934e-03	2,0987e-02	-7,0710e-03	-1,2255e-02	-1,3247e-02	1,0220e-03	8,6160e-04
5e+4	Najboljši	-2,6103e-02	1,4670e-06	-3,4935e-02	-1,4436e-01	-6,8407e-02	-5,1683e-03	-2,5580e-01
	Srednji	-4,1466e-02	-1,1338e-04	-6,2933e-02	-2,1474e-01	-8,8668e-02	-1,6661e-02	-2,6517e-01
	Najslabši	-3,4293e-02	-4,3558e-04	-1,0645e-01	-2,7920e-01	-1,1257e-01	-1,8523e-02	-2,0406e-01
	Poprečje	-3,7788e-02[†]	-1,7393e-04[†]	-6,2369e-02[†]	-2,0414e-01[†]	-8,7191e-02[†]	-1,4911e-02[†]	-2,5545e-01[†]
	Std	-1,4962e-03	-1,4109e-04	-1,6357e-02	-2,5083e-02	-1,2716e-02	-3,1221e-03	2,0273e-02
5e+5	Najboljši	-1,9707e-02	3,4932e-06	-5,9777e-03	-4,0976e-02	-4,0906e-02	-6,0501e-03	-2,1614e-01
	Srednji	-5,0697e-02	4,3262e-06	-1,0781e-02	-1,2801e-01	-6,4423e-02	-1,7892e-02	-2,5128e-01
	Najslabši	-3,3591e-02	5,4167e-06	-3,8528e-02	-1,5585e-01	-1,2335e-01	-3,1657e-02	-1,9540e-01
	Poprečje	-4,1238e-02[†]	4,3761e-06[‡]	-1,4256e-02[†]	-1,0798e-01[†]	-7,2517e-02[†]	-1,7967e-02[†]	-2,4425e-01[†]
	Std	-5,3201e-03	4,0255e-07	-8,2847e-03	-1,5502e-02	-2,2916e-02	-6,6107e-03	9,4200e-03

Tabela 4.58: Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma MO_PSO in DEMOWSA na testnih funkcijah 8–13 z $M = 3$.

<i>FEs</i>		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Najboljši	7,3462e-04	-9,2240e-03	5,0026e-02	2,8447e-01	-1,0542e-01	-4,4755e-02
	Srednji	8,7460e-04	-4,5590e-03	4,3986e-02	2,8489e-01	-9,5359e-02	-1,0412e-01
	Najslabši	4,5080e-04	2,9330e-03	2,9846e-02	2,8407e-01	-8,5081e-02	-1,2629e-01
	Poprečje	9,2450e-04[‡]	-4,5870e-03[†]	4,1510e-02[‡]	2,8457e-01[‡]	-9,5212e-02[†]	-1,0363e-01[†]
	Std	7,4490e-05	3,4422e-03	-6,5480e-03	-1,0794e-04	4,5845e-03	-1,6828e-02
5e+4	Najboljši	-3,3575e-04	-3,9250e-03	9,9328e-02	2,7287e-01	-1,6375e-01	3,8307e-02
	Srednji	-3,9098e-04	-3,9670e-03	8,6662e-02	2,7800e-01	-1,5372e-01	1,7673e-02
	Najslabši	-5,4062e-04	-2,1480e-03	7,5485e-02	2,8015e-01	-1,4792e-01	1,5192e-02
	Poprečje	-3,9676e-04[†]	-2,8420e-03[†]	8,6857e-02[‡]	2,7735e-01[‡]	-1,5491e-01[†]	2,1890e-02[‡]
	Std	-3,5960e-05	8,6280e-04	-7,4704e-03	1,8988e-03	3,9049e-03	-8,0530e-03
5e+5	Najboljši	-3,5974e-04	-4,8768e-03	1,1658e-01	1,9515e-01	-1,7177e-01	1,5906e-01
	Srednji	-5,5392e-04	-4,7070e-03	1,0283e-01	2,1272e-01	-1,6958e-01	1,3555e-01
	Najslabši	-7,8334e-04	4,8400e-04	9,2210e-02	2,2191e-01	-1,6567e-01	1,1226e-01
	Poprečje	-5,6547e-04[†]	-4,6020e-03[†]	1,0349e-01[‡]	2,1091e-01[‡]	-1,6952e-01[†]	1,3498e-01[‡]
	Std	-1,0590e-04	1,0927e-03	-6,7550e-03	8,0339e-03	1,5510e-03	-1,0217e-02

Tabela 4.59: Razlika vrednosti indikatorja $I_{\bar{H}}$ za algoritma MO_PSO in DEMOWSA na testnih funkcijah 8–13 z $M = 5$.

<i>FEs</i>		8. S_DTLZ2	9. R_DTLZ2	10. S_DTLZ3	11. WFG1	12. WFG8	13. WFG9
5e+3	Najboljši	3,1880e-04	-3,8618e-01	3,0791e-02	5,2970e-01	-6,3956e-01	-3,3896e-02
	Srednji	4,9880e-03	-3,9438e-01	-5,9503e-03	5,3067e-01	-6,7316e-01	-3,7300e-02
	Najslabši	8,6750e-03	-3,9721e-01	-4,7538e-02	5,3187e-01	-6,7241e-01	-3,8880e-02
	Poprečje	5,2140e-03[‡]	-3,9282e-01[†]	-8,3772e-03[†]	5,3058e-01[‡]	-6,7296e-01[†]	-3,6580e-02[†]
	Std	1,9158e-03	-1,6437e-03	-1,8386e-02	6,0508e-04	-4,6811e-03	-8,8500e-04
5e+4	Najboljši	-5,5402e-03	-3,6618e-01	5,4982e-02	5,2823e-01	-7,3728e-01	-8,1546e-02
	Srednji	-6,8528e-03	-3,7689e-01	3,9593e-02	5,2968e-01	-7,5187e-01	-1,0126e-01
	Najslabši	-7,6598e-03	-3,8255e-01	2,0161e-02	5,3216e-01	-7,4608e-01	-1,2895e-01
	Poprečje	-6,7595e-03[†]	-3,7650e-01[†]	3,8655e-02[‡]	5,2976e-01[‡]	-7,5349e-01[†]	-1,0133e-01[†]
	Std	-5,9108e-04	-4,2130e-03	-8,7498e-03	8,7101e-04	-8,2080e-04	-1,2424e-02
5e+5	Najboljši	-4,4548e-03	-3,4884e-01	6,4006e-02	5,1546e-01	-7,9455e-01	-7,3924e-02
	Srednji	-5,2083e-03	-3,6695e-01	5,7339e-02	5,1990e-01	-7,9595e-01	-7,5976e-02
	Najslabši	-6,4602e-03	-3,7058e-01	4,8138e-02	5,2182e-01	-7,7847e-01	-9,2313e-02
	Poprečje	-5,3907e-03[†]	-3,6570e-01[†]	5,7645e-02[‡]	5,1960e-01[‡]	-7,9377e-01[†]	-7,7217e-02[†]
	Std	-5,8209e-04	-5,3636e-03	-4,5838e-03	1,7916e-03	5,8882e-03	-3,4866e-03

4.4.9 Povzetek rezultatov primerjav

V tabeli 4.60 smo pripravili povzetek rezultatov primerjav ocen algoritmov pri $5e+5$ ovrednotenjih kriterijske funkcije. Tabela za vsak z DEMOwSA primerjan algoritmom prikazuje število boljših, slabših in nesignifikantnih razlik pri indikatorjih I_R in $I_{\bar{H}}$. V zadnjem stolpcu je navedena še vsota obeh razlik v vrednostih indikatorjev. Vnosi v tabeli so urejeni po omenjeni vsoti. Kot vidimo, je algoritom DEMOwSA v povprečju boljši od prvih šestih algoritmov, zadnja dva v tabeli pa v povprečju na danem testnem ogrodju dajeta nekoliko boljše rezultate. Omenimo še, da smo ocene indikatorjev za primerjane algoritme dobili iz člankov, ki so preizkušali primerjane algoritme. Le za osnovni algoritmom DEMO smo uporabili enako inicializirano populacijo vektorjev ter fiksne krmilne parametre in izvedli potrebne simulacije, saj zanj potrebni eksperimentalni rezultati pred našo primerjavo še niso bili znani.

Tabela 4.60: Povzetek rezultatov primerjanih algoritmov pri $5e+5$ ovrednotenjih kriterijske funkcije.

Ime	Vir	Indikator I_R			Indikator $I_{\bar{H}}$			Vsota
		Boljše	Slabše	Nesig.	Boljše	Slabše	Nesig.	
MO_DE	[40]	14	5	0	13	6	0	16
MTS	[110]	11	7	1	12	7	0	9
MO_PSO	[111]	11	6	2	10	6	3	9
DEMO	[33]	11	7	1	10	5	4	9
MOSaDE	[104]	9	10	0	12	7	0	4
NSGAII-PBX	[109]	10	9	0	10	8	1	3
NSGAII-SBX	[108]	8	11	0	9	10	0	-4
GDE3	[106]	6	12	1	6	13	0	-13

4.5 Analiza dinamike samoprilagodljivih parametrov

Za namene študije učinkovitosti samoprilagajanja smo analizirali še vrednosti parametrov skozi tek programa, v odvisnosti od števila ovrednotenj kriterijske funkcije (*FEs*).

Iz tabele 4.61 vidimo povprečja vrednosti krmilnih parametrov F in CR pri vsaki od funkcij po $5e+5$ ovrednotenjih. Posamezna vrednost predstavlja povprečje (oz. v oklepajih standardni odklon) krmilnega parametra skozi zadnjih 25 generacij za vse neodvisne zagone, za primerke, ki so izboljšali starševsko rešitev. Kot vidimo, se krmilna parametra samoprilagodita na precej različne vrednosti, zato ni možno sklepati o fiksni vrednosti posameznega parametra, ki bi enako dobro delovala za vse funkcije.

Tabela 4.61: Vrednosti krmilnih parametrov po $5e+5$ ovrednotenjih kriterijske funkcije.

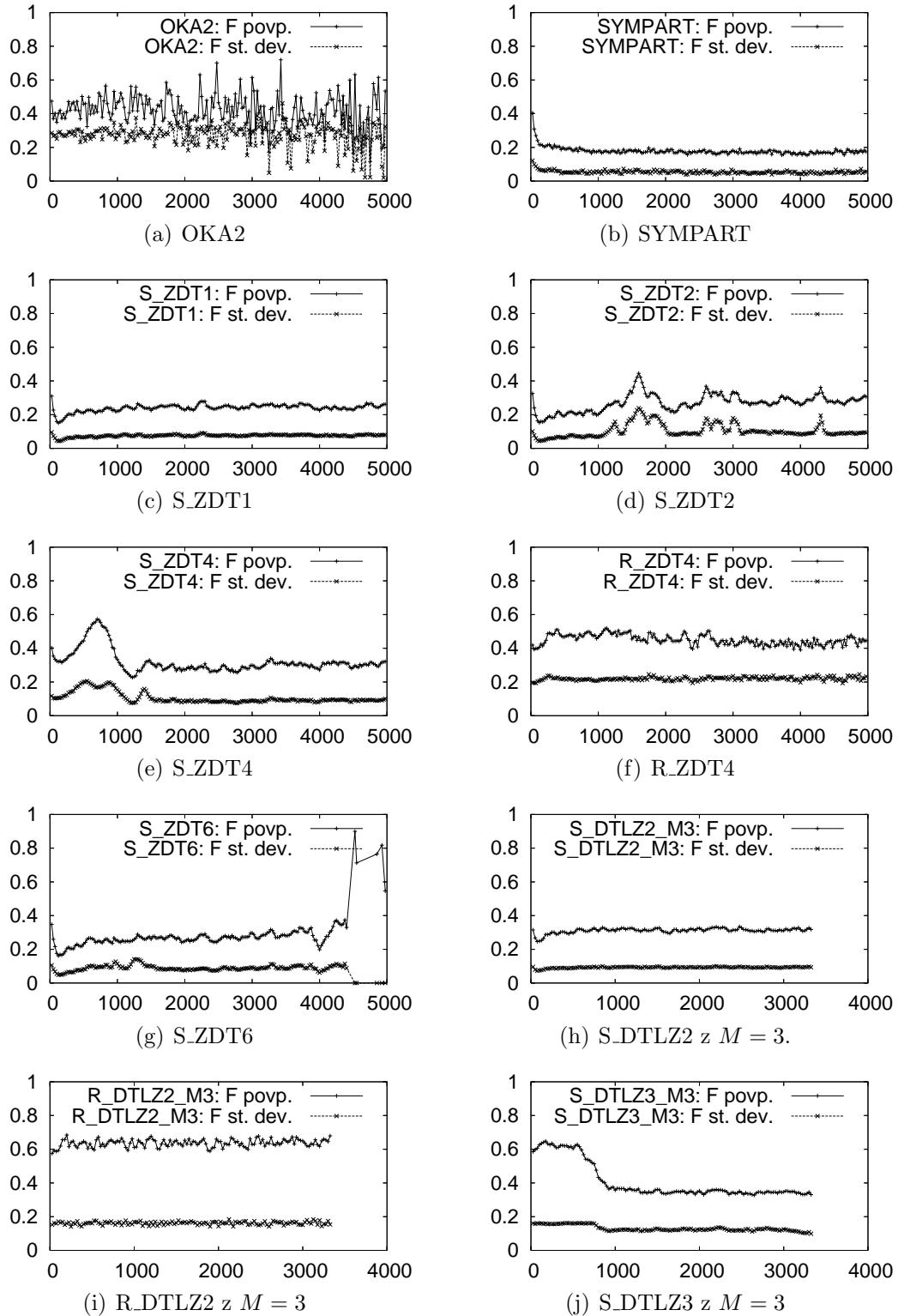
Zap. št.	Funkcija	\bar{F}_{imp}	\bar{CR}_{imp}
1.	OKA2	0,53 (0,32)	0,40 (0,19)
2.	SYMPART	0,17 (0,06)	0,75 (0,16)
3.	S_ZDT1	0,26 (0,08)	0,55 (0,14)
4.	S_ZDT2	0,31 (0,09)	0,56 (0,14)
5.	S_ZDT4	0,32 (0,10)	0,55 (0,14)
6.	R_ZDT4	0,45 (0,23)	0,26 (0,17)
7.	S_ZDT6	0,55 (0,00)	0,40 (0,00)
8.	S_DTLZ2_M3	0,32 (0,09)	0,49 (0,13)
9.	R_DTLZ2_M3	0,68 (0,15)	0,35 (0,11)
10.	S_DTLZ3_M3	0,33 (0,10)	0,48 (0,12)
11.	WFG1_M3	0,52 (0,16)	0,24 (0,08)
12.	WFG8_M3	0,35 (0,12)	0,46 (0,13)
13.	WFG9_M3	0,63 (0,18)	0,31 (0,10)
14.	S_DTLZ2_M5	0,36 (0,11)	0,32 (0,09)
15.	R_DTLZ2_M5	0,61 (0,16)	0,23 (0,07)
16.	S_DTLZ3_M5	0,62 (0,16)	0,06 (0,02)
17.	WFG1_M5	0,63 (0,16)	0,24 (0,07)
18.	WFG8_M5	0,54 (0,15)	0,40 (0,10)
19.	WFG9_M5	0,66 (0,16)	0,27 (0,08)

Pri vseh funkcijah vidimo, da začetne vrednosti $CR_{init} = 0,3$ ni med njimi, kar dodatno nakazuje na uspešno samoprilagajanje krmilnih parametrov. Po $5e+5$ ovrednotenjih znaša razpon dobljenih povprečnih vrednosti za parameter F od $0,17 \pm 0,06$ (SYM-PART) do $0,68 \pm 0,15$ (R_DTLZ2_M3) ter za parameter CR od $0,06 \pm 0,02$ (S_DTLZ3_M5) do $0,75 \pm 0,16$ (SYM-PART). Krmilna parametra F in CR zajameta

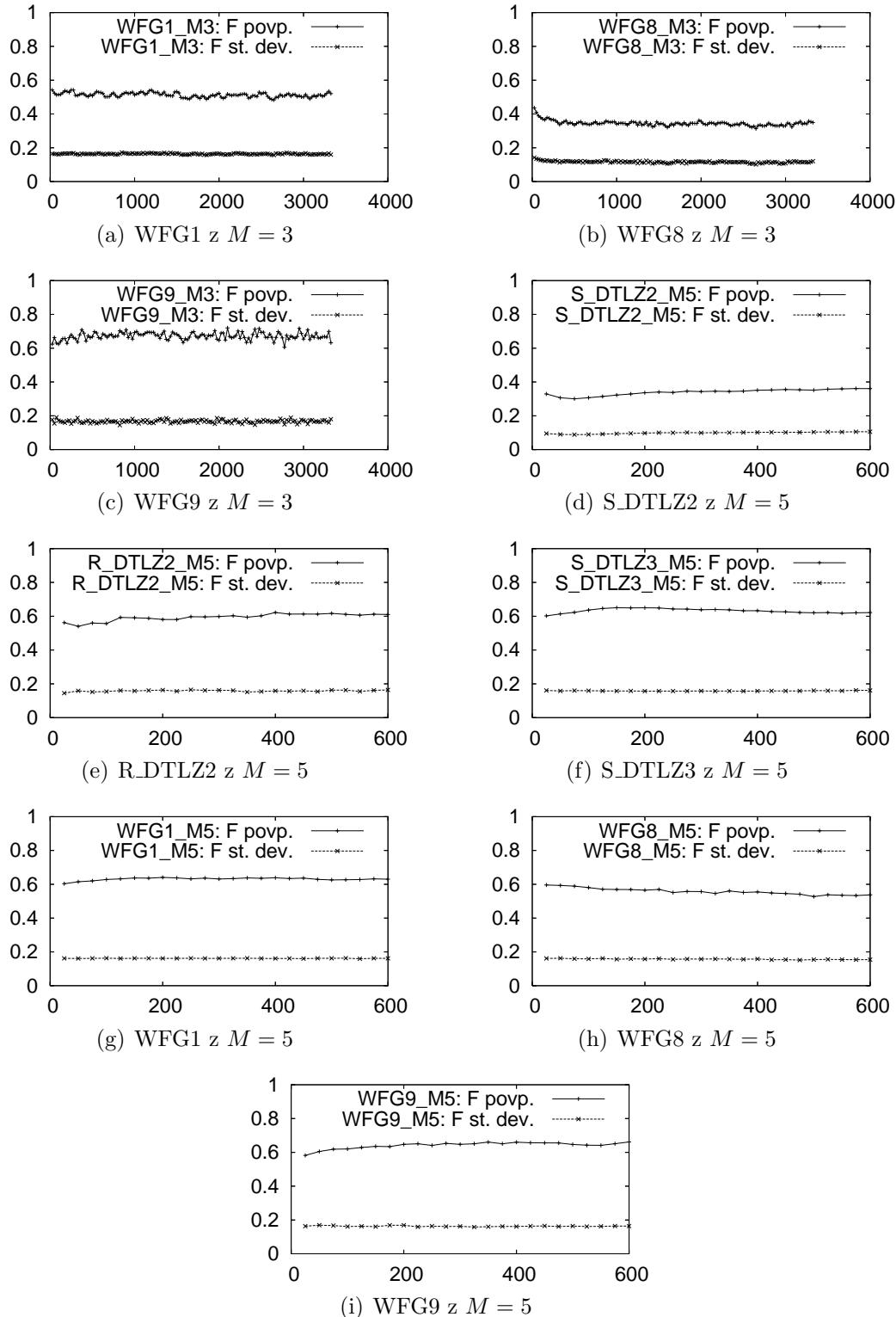
različne kombinacije nižjih in višjih vrednosti. Tako sta si pri funkciji S_DTLZ2_M5 podobna ter pri funkcijah SYM-PART in S_DTLZ3_M5 precej različna.

Ker je za nekatere izmed funkcij dinamika krmilnih parametrov F in CR precej različna, na slikah 4.12–4.15 prikazujemo tudi grafe vrednosti teh parametrov skozi generacije. Iz slik vidimo, da vrednosti pogosto menjajo smer samoprilagajanja, saj algoritem menjuje faze lokalnega in globalnega iskanja, ki ju krmilna parametra nadzorujeta.

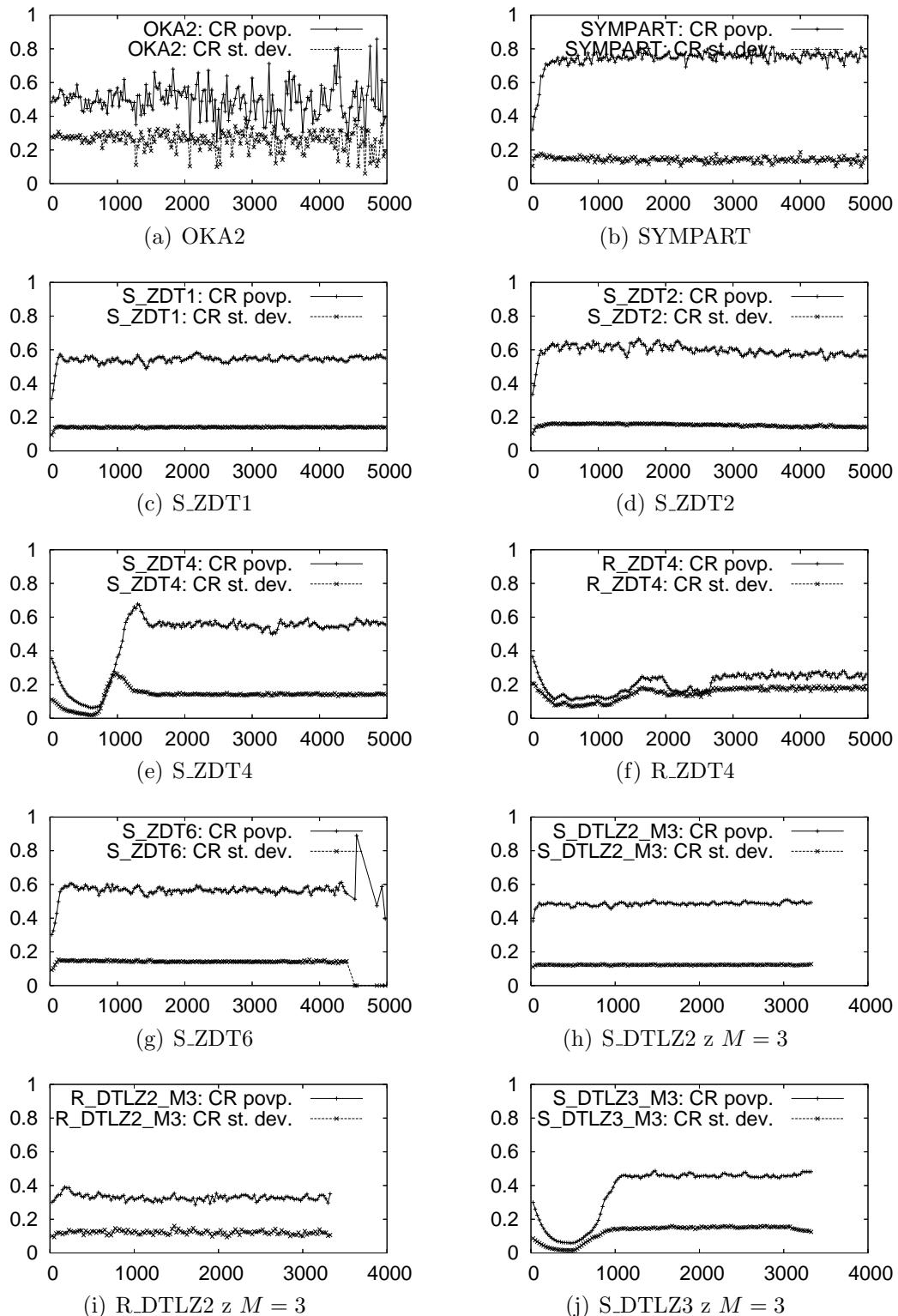
Ker je iz študije dinamike krmilnih parametrov F in CR razvidno, da se ti precej spreminjajo in jih ni moč nastaviti enako za poljuben problem, samoprilagajanje še toliko bolj pomaga pri reševanju industrijskih problemov [26]. Za te probleme praviloma pogosto velja, da ročno nastavljanje krmilnih parametrov in znovični zagon eksperimentov z ovrednotenji posameznih poskusnih rešitev z novimi parametri podraži razvoj izdelka zaradi vključenega potrošnjega materiala in podaljšanega razvojnega časa. Kot smo pokazali že v tabelah 4.12–4.17, smo z uporabo samoprilagajanja povečali robustnost algoritma na večini testnih primerov, tj. da je algoritem z večjo verjetnostjo uspešno reševal optimizacijske probleme. S tem statistično potrjujemo signifikantni vpliv, kot tudi povečanje učinkovitosti optimizacije zaradi vpeljanega mehanizma samoprilagajanja krmilnih parametrov.



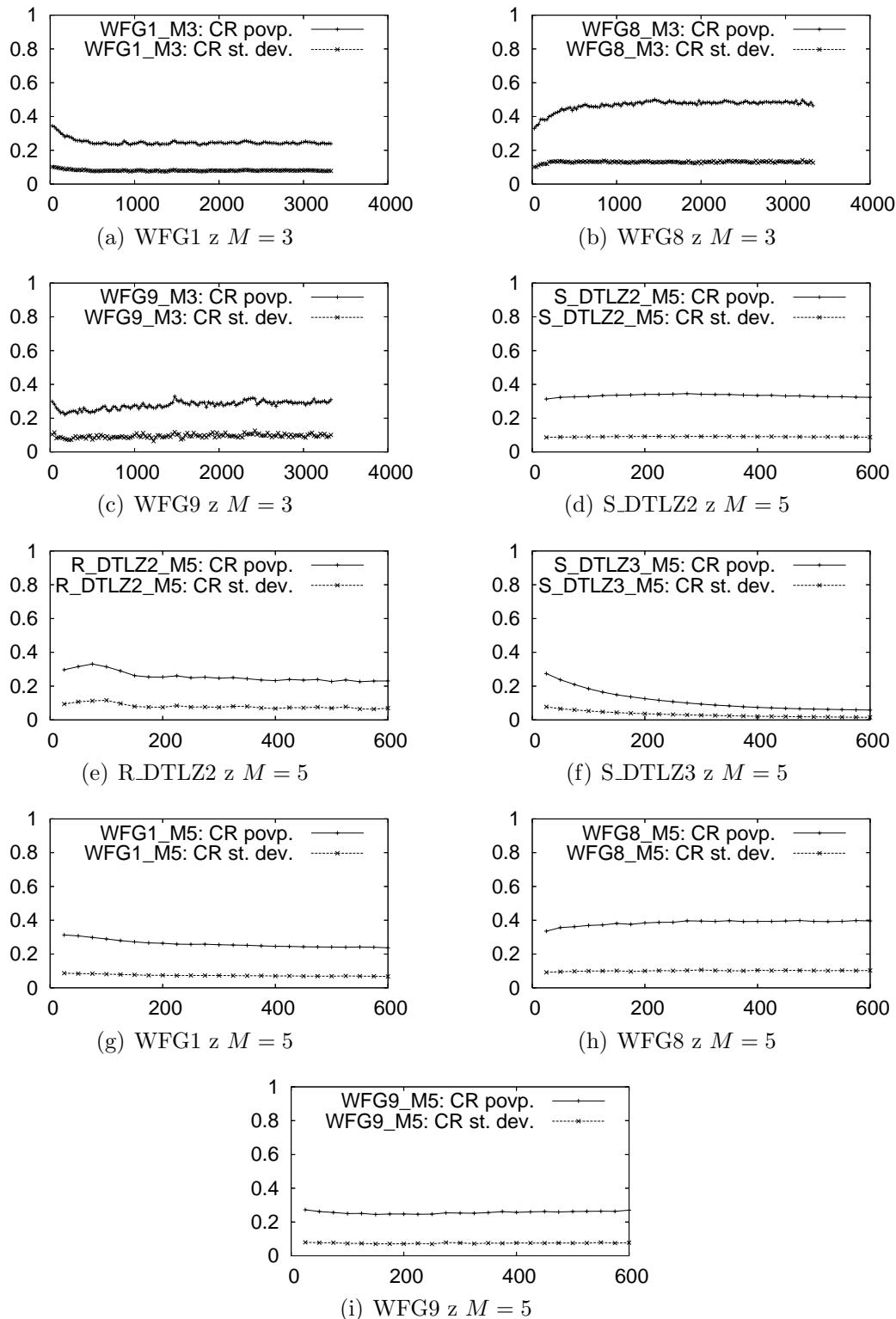
Slika 4.12: Na 25 generacijah povprečena vrednost povprečne izboljšane vrednosti krmilnega parametra F za 25 neodvisnih zagonov (funkcije 1–7 in DTLZ z $M = 3$).



Slika 4.13: Na 25 generacijah povprečna vrednost povprečne izboljšane vrednosti krmilnega parametra F za 25 neodvisnih zagonov (funkcije 8–13 in ostale z $M = 5$).



Slika 4.14: Na 25 generacijah povprečena vrednost povprečne izboljšane vrednosti krmilnega parametra CR za 25 neodvisnih zagonov (funkcije 1–7 in DTLZ z $M = 3$).



Slika 4.15: Na 25 generacijah povprečna vrednost povprečne izboljšane vrednosti krmilnega parametra CR za 25 neodvisnih zagonov (funkcije 8–13 in ostale z $M = 5$).

Poglavlje 5

Zaključek

Predstavili smo področje evolucijskih algoritmov, natančneje njihovo uporabo pri večkriterijski optimizaciji. Pri tem smo osvetlili ključne dele algoritmov, ki so trenutno aktualni pri raziskavah na tem področju.

Po uvodu smo podali ključna sorodna dela, kjer smo predstavili tudi splošne cilje večkriterijske optimizacije in navedli nekaj preprostih metod za reševanje. Ker so pri reševanju večkriterijske optimizacije precej bolj primerni populacijski evolucijski algoritmi, smo po njihovem ozadju in navedbi ključnih njihovih skupin predstavili algoritem diferencialne evolucije ter mehanizem samoprilagajanja. Tukaj smo podali tudi nekaj izboljšav osnovnega algoritma in mehanizmov samoprilagajanja, uporabljenih v literaturi. Pregled sorodnih del smo zaključili s seznamom nekaterih predstavnikov posameznih skupin evolucijskih algoritmov, izmed katerih je poudarek še posebej na algoritmih diferencialne evolucije za večkriterijsko optimizacijo.

Predstavili smo tudi svoj pristop k večkriterijski optimizaciji z diferencialno evolucijo, ki uporablja mehanizem samoprilagajanja krmilnih parametrov. Po opisu kodiranja teh parametrov in izračunu prilagajanja novih vrednosti smo za lažjo razumevanje zapisali še psevdokod. Zapisan psevdokod smo s pridom uporabili za prikaz analitične ocene računske zahtevnosti algoritma, ki je sledila primeru zagona algoritma. Slednja, skupaj z analizo kakovosti algoritma in njeno statistično primerjavo ter dinamiko samoprilagodljivih parametrov, predstavljata ključne empirične rezultate tega dela. Z zmožnostjo samoprilagajanja krmilnih parametrov dosega algoritem v povprečju boljše rezultate pri večkriterijski optimizaciji in je bolj robusten za uporabo na novih funkcijah, saj uporabniku ni potrebno nastaviti krmilnih parametrov F in CR . S tem smo z novim algoritmom dosegli dva pomembna cilja.

Glavna omejitev algoritma je ta, da lahko z njim rešujemo le numerične optimizacijske probleme oz. probleme, ki jih lahko pretvorimo v numerične matematične modele. Za reševanje kombinatoričnih optimizacijskih problemov bi morali algoritem dopolniti, tako da bi spremenili način mutacije v algoritmu diferencialne evolucije.



Za pripravo poskusov je bilo potrebno preučiti nabor testnih funkcij (programske kode), različne algoritme za večkriterijsko optimizacijo, implementirati mehanizem samoprilagajanja krmilnih parametrov v programu DEMO in izdelati programsko okolje za izvajanje ter vrednotenje poskusov.

Izvajanje poskusov predstavljenega algoritma je potekalo paralelno na večprocesorskem računalniškem sistemu. Izvedba poskusov in vrednotenje rezultatov 25 neodvisnih zagonov pri 19 funkcijah sta skupaj trajala približno šest ur. Če smo povečali velikost aproksimacijske množice, je bilo potrebno čakati na rezultate več kot en teden. Z namenom pohitritve smo poskuse izvajali vzporedno na dveh večprocesorskih računalniških sistemih.

V nadaljevanju nameravamo še izboljšati opisani algoritem za večkriterijsko optimizacijo, posebej mutacijo in selekcijo. Pri mutaciji želimo natančneje preučiti različne načine samoprilagajanja, pri selekciji pa kombinacije različnih seleksijskih kriterijev. Nadgradili bi lahko tudi mehanizem arhiviranja primerkov z rezanjem tik pred izpisom vsake aproksimacijske množice [106]. Zanimivo bi bilo videti tudi analizo potrebnega števila ovrednotenj v odvisnosti od zahtevane različne stopnje uspešnosti na indikatorjih kakovosti ali pri drugem ustavitevem pogoju in tako morda empirično potrditi postavljen analitični model za hipotezo, da ima algoritem za nekatere funkcije manjši zahtevnostni razred, kot algoritem naključnega preiskovanja. Z algoritmom bi se lahko lotili optimizacije ne le umetno ustvarjenih, temveč tudi praktičnih problemov. Pri tem bi morda morali najprej pripraviti matematični model, ki bi ga želeli optimizirati. Primer takšnega praktičnega problema iz ekologije je optimalna raba tal [113, 114, 115, 116]. Algoritem bi lahko še hibridizirali s klasično optimizacijsko metodo sekvenčnega kvadratičnega programiranja (SQP) [117, 118, 108].

Dodatek A

Enovit psevdokod algoritma DEMOwSA

Vhod: $func_type$ – zaporedna številka funkcije, ki jo optimiziramo.

Vhod: $\mathbf{g}(\mathbf{x}) \geq 0$ – omejitve funkcije, ki jo optimiziramo.

Vhod: D in M – dimenzija prostora spremenljivk in dimenzija prostora kriterijev.

Vhod: $MAXFEs$ – maksimalno število ovrednotenj (ustavitevni pogoj).

Vhod: NP – velikost populacije in aproksimacijske množice \bar{N} .

Vhod: s_{DE} – številka strategije mutacije DE (strategija preiskovanja).

Vhod: sel_typeMO – strategija večkriterijske selekcije (strategija sekanja).

Vhod: F_{init} – začetna vrednost parametrov F .

Vhod: CR_{init} – začetna vrednost parametrov CR .

Vhod: F_{min} (F_{max}) – najmanjša (največja) možna vrednost parametra F .

Vhod: CR_{min} (CR_{max}) – najmanjša (največja) možna vrednost parametra CR .

Vhod: τ – stopnja učenja za prilagajanje samoprilagodljivih parametrov F in CR .

Vhod: RNi_{seed} – seme generatorja enakomerno porazdeljenih naključnih števil.

Vhod: **front_gen** – množica intervalov vzorčenj populacije \mathbf{P}_g na FEs .

Izhod: Seznam aproksimacijskih množic iskalnih parametrov \mathcal{P} (množica matrik).

- 1: $RNj_{seed} := \text{rand}(RNi)$ {Generiraj seme generatorja normalno porazdeljenih števil RNj_{seed} .}
- 2: $g = 0$ {Z g štej generacije populacij od 0 naprej.}
- 3: **for** $i = 0$ to $NP - 1$ **do**
- 4: **a** = \emptyset
- 5: **for** $j = 0$ to D **do**
- 6: $a_j := \text{rand}_{RNi}(x_{min,i}, x_{max,i})$ { $x_{min,i}$ in $x_{max,i}$ sta vsebovana v omejitvah $\mathbf{g}(\mathbf{x}) \geq 0$.}
- 7: **end for**

```

8:   s = ( $F_{init}$ ,  $CR_{init}$ )
9:   G =  $\mathbf{0}_{M+1}$  { $M$  kriterijev in nedominantnost.}
10:   $\mathbf{p}_{0,i} = (\mathbf{a}, \mathbf{s}, \mathbf{G})$  {Sestavi primerek iz iskalnih parametrov in krmilnih parametrov.}
11:  {Ovrednoti primerek v začetni populaciji.}
12:   $x_1, x_2, x_3 \leftarrow \mathbf{p}$  {Iz  $\mathbf{p}$  izlušči iskalne parametre.}
13:   $f_1(\mathbf{x}) = x_1$  {Funkcija OKA2.}
14:   $f_2(\mathbf{x}) = 1 - \frac{1}{4\pi^2} (x_1 + \pi)^2 + |x_2 - 5 \cos(x_1)|^{\frac{1}{3}} + |x_3 - 5 \sin(x_1)|^{\frac{1}{3}}$ 
15:   $\mathbf{f}(\mathbf{x}) = (f_1, f_2)$ 
16:   $\mathbf{p}_{0,i} \leftarrow \mathbf{f}(\mathbf{x})$  {V  $\mathbf{p}$  shrani vrednosti kriterijev.}
17:   $\mathbf{P}_0 := \mathbf{P}_0 \cup \mathbf{p}_{0,i}$  {Dopolnjuj začetno populacijo  $\mathbf{P}_0$ .}
18: end for
19:  $FEs := NP$  {Ovrednotenih je  $NP$  primerkov.}
20: while  $FEs < MAXFEs$  do
21:   if  $FEs \geq front\_gen_0$  then
22:      $\mathcal{P} := \mathcal{P} \cup \mathbf{P}_g^{NP,D}$  {V  $\mathcal{P}$  dodaj  $D$  iskalnih parametrov za vseh  $NP$  primerkov iz populacije.}
23:     front_gen := front_gen \  $front\_gen_0$  {Odstrani prvi element množice.}
24:   end if
25:   {Naključno oštevilči osebke iz populacije  $\mathbf{P}_g$ .}
26:    $\mathbf{r} := \emptyset$  {Seznam naključnih števil.}
27:    $\mathbf{r}' := \emptyset$  {Seznam indeksov, glej sliko 3.2.}
28:   for  $i = 0$  to  $NP$  do
29:      $\mathbf{r} := \mathbf{r} \cup \text{rand}_{RNi}(0, 1)$  {Dodajaj enakomerno porazdeljena naključna števila – slika 3.2b.}
30:      $\mathbf{r}' := \mathbf{r}' \cup i$  {Pred vsak element seznama  $\mathbf{r}$  shrani indeks  $i$  – slika 3.2c.}
31:   end for
32:   Seznam  $\mathbf{rr}'$  uredi po  $\mathbf{r}$ . {Hitro urejanje:  $\Theta(NP \log NP)$  – slika 3.2d.}
33:    $\mathbf{P}'_g := \emptyset$  {Nov seznam primerkov v populaciji.}
34:   for  $i = 1$  to  $NP$  do
35:      $\mathbf{P}'_g := \mathbf{P}'_g \cup \mathbf{p}_{g,r_i}$  {Primerke na indeksih  $r_i$  iz populacije  $\mathbf{P}_g$  dodaj v  $\mathbf{P}'_g$  – slika 3.2e–h.}
36:   end for
37:    $\mathbf{P}_g := \mathbf{P}'_g$  { $\mathbf{P}'_g$  bomo preimenovali v  $\mathbf{P}_g$ , ni dejanskega kopiranja.}
38:   for  $i = 0$  to  $NP - 1$  do
39:     {Diferencialna Evolucija.}
40:     repeat
41:        $i_1 := \text{rand}_{RNi}(0, NP); i_2 := \text{rand}_{RNi}(0, NP); i_3 := \text{rand}_{RNi}(0, NP)$ 

```

```

42: until not ( $i = i_3$  or  $i_1 = i_3$  or  $i_2 = i_3$ ) {Izberi paroma različne indekse
    $\mathbf{p}_{g,i_1}, \mathbf{p}_{g,i_2}, \mathbf{p}_{g,i_3} \in \mathcal{P}_g\}$ 
43:  $\langle F_G \rangle_i = \frac{F_{i,G} + F_{r_1,G} + F_{r_2,G} + F_{r_3,G}}{4}$  {Globalno povprečenje primerkov.}
44:  $F_{i,G+1} = \langle F_G \rangle_i \times e^{\tau N(0,1)}$  {Logaritmično normalna porazdelitev samoprila-
   gajanja.}
45: if  $F_{i,G+1} < F_{min}$  then
46:    $F_{i,G+1} := F_{min}$  {Popravimo  $F$  v meje.}
47: else if  $F_{i,G+1} > F_{max}$  then
48:    $F_{i,G+1} := F_{max}$ 
49: end if
50:  $\mathbf{v}_{i,G+1} = \mathbf{x}_{r_1,G} + F_{i,G+1} \times (\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G})$ 
51:  $\langle CR_G \rangle_i = \frac{CR_{i,G} + CR_{r_1,G} + CR_{r_2,G} + CR_{r_3,G}}{4}$  {Globalno povprečenje skozi
   primerke.}
52:  $CR_{i,G+1} = \langle CR_G \rangle_i \times e^{\tau N(0,1)}$  {Logaritmično normalna porazdelitev samo-
   prilagajanja.}
53: if  $CR_{i,G+1} < CR_{min}$  then
54:    $CR_{i,G+1} := CR_{min}$  {Popravimo  $CR$  v meje.}
55: else if  $CR_{i,G+1} > CR_{max}$  then
56:    $CR_{i,G+1} := CR_{max}$ 
57: end if
58:  $j_{rand,g,i} := \text{rand}_{RNi}(0, D)$  {Vsaj enega parametra pri križanju ne bomo za-
   menjali.}
59: for  $j = 0$  to  $D$  do
60:   if  $\text{rand}_{RNi}(0, 1) \leq CR_{i,G+1}$  ali  $j = j_{rand,g,i}$  then
61:      $u_{i,j,G+1} = v_{i,j,G+1}$  {Križamo.}
62:   else
63:      $u_{i,j,G+1} = x_{i,j,G}$  {Ne križamo.}
64:   end if
65:   if  $u_{i,j,G+1} < x_{min,i}$  then
66:      $u_{i,j,G+1} := x_{min,i}$  { $x_{min,i}$  in  $x_{max,i}$  sta vsebovana v omejitvah  $\mathbf{g}(\mathbf{x}) \geq 0$ .}
67:   else if  $u_{i,j,G+1} > x_{max,i}$  then
68:      $u_{i,j,G+1} := x_{max,i}$ 
69:   end if
70: end for
71: {Ovrednoti primerek glede na funkcijo func_type.}
72:  $x_1, x_2, x_3 \leftarrow \mathbf{p}$  {Iz  $\mathbf{p}$  izlušči iskalne parametre.}
73:  $f_1(\mathbf{x}) = x_1$ 

```

```

74:    $f_2(\mathbf{x}) = 1 - \frac{1}{4\pi^2} (x_1 + \pi)^2 + |x_2 - 5 \cos(x_1)|^{\frac{1}{3}} + |x_3 - 5 \sin(x_1)|^{\frac{1}{3}}$ 
75:    $\mathbf{f}(\mathbf{x}) = (f_1, f_2)$ 
76:    $\mathbf{u} \leftarrow \mathbf{f}(\mathbf{x}) \quad \{V \mathbf{p} shrani vrednosti kriterijev.\}$ 
77:    $FEs := FEs + 1 \quad \{Beleži število ovrednotenj.\}$ 
78:    $\{Ugotovi dominiranost primerka in starša v prostoru kriterijev.\}$ 
79:    $d_{new} = 0 \quad \{Zastavica beleži, ali je nov kandidatni primerek že dominiran po enem kriteriju.\}$ 
80:    $d_{parent} = 0 \quad \{Zastavica beleži, ali je starševski primerek že dominiran po enem od kriterijev.\}$ 
81:   for  $j = 0$  to  $M$  do
82:     if  $\mathbf{u}_j < \mathbf{p}_{i,j}$  then
83:        $d_{parent} = 1$ 
84:     else if  $\mathbf{p}_{i,j} < \mathbf{u}_j$  then
85:        $d_{new} = 1$ 
86:     end if
87:     if  $d_{parent} = 1$  and  $d_{new} = 1$  then
88:        $ND := 0 \quad \{Rešitvi sta neprimerljivi, nobena ne dominira druge.\}$ 
89:       break  $\{Takoj prenehamo s primerjavo – izboljšava iz DEMO na DEMOWSA.\}$ 
90:     end if
91:   end for
92:    $ND_u := d_{new} - d_{parent} \quad \{1, če je dominanten nov, -1, če je dominanten starševski primerek.\}$ 
93:   if  $ND_u = 1$  then
94:      $\mathbf{P}_g := (\mathbf{P}_g \setminus \mathbf{P}_i) \cup \mathbf{u} \quad \{\text{Primerek } \mathbf{u} \text{ dominanten, prepiši starša s primerkom.}\}$ 
95:   else
96:     if  $ND_u = -1$  then
97:        $\mathbf{P}_g := \mathbf{P}_g \quad \{\text{Starš } \mathbf{p}_i \text{ dominanten, primerek } \mathbf{u} \text{ izbriši}\}.$ 
98:     else
99:        $\mathbf{P}_g := \mathbf{P}_g \cup \mathbf{u} \quad \{\text{Dodaj primerek } \mathbf{u} \text{ v aproksimacijsko množico rešitev.}\}$ 
100:    end if
101:   end if
102: end for
103:  $\{Poreži populacijo \mathbf{P}_g \text{ do velikosti } NP.\}$ 
104:  $\{Najprej izmerimo kakovost vseh rešitev: tj. nakopičenost.\}$ 
105:  $\mathbf{S}_g := \mathbf{0} \quad \{\text{Primerkom iz } \mathbf{P}_g \text{ inicializiraj moč na 0 – moč je predstavljena z dominantnostjo.}\}$ 

```

```

106:   for all  $\mathbf{p}_{g+1,i} \in \mathbf{P}_{g+1}$  do
107:     for all  $\mathbf{p}_{g+1,j} \in \mathbf{P}_{g+1}$  do
108:       if Primerjaj rešitvi( $\mathbf{p}_{g,i}, \mathbf{p}_{g,j}, M$ ) = 1 then
109:          $S_{g,i} := S_{g,i} + 1$  {Prešteje dominirane primerke  $S_{\mathbf{p}_{g,i}} = |\{\mathbf{p}_{g,j} | \mathbf{p}_{g,j} \in \mathbf{P}_g\} \wedge \mathbf{p}_{g,i} \prec \mathbf{p}_{g,j}\|$ }
110:       end if
111:     end for
112:   end for
113:    $\mathbf{R}_g := \mathbf{0}$  {Iz  $\mathbf{S}_g$  izračunamo grobo uspešnost  $R_g$ , ki je vsota moči vseh dominiranih od  $\mathbf{p}_{g,i}$ .}
114:   for all  $\mathbf{p}_{g+1,i} \in \mathbf{P}_{g+1}$  do
115:     for all  $\mathbf{p}_{g+1,j} \in \mathbf{P}_{g+1}$  do
116:       if Primerjaj rešitvi( $\mathbf{p}_{g,i}, \mathbf{p}_{g,j}, M$ ) = 1 then
117:          $R_{g,i} := R_{g,i} + R_{g,j}$  {Čim manjša vrednost je boljša.}
118:       end if
119:     end for
120:   end for
121:    $\mathbf{n}_n = -\mathbf{1}_{|\mathbf{P}_{g+1}| \times |\mathbf{P}_{g+1}|}$  {Kvadratna matrika velikosti  $|\mathbf{P}_{g+1}|$ , za kreiranje potrebuje  $\Theta((|\mathbf{P}_{g+1}|)^2)$ .}
122:    $\mathbf{c} = \mathbf{1}_{|\mathbf{P}_{g+1}|}$  {Vektor enic velikosti  $|\mathbf{P}_{g+1}|$ .}
123:   for all  $\mathbf{p}_{g+1,i} \in \mathbf{P}_{g+1}$  do
124:      $n_{n;i,0} := i$  {Zgradimo vektor razdalj za vsak primerek.}
125:     for  $j := 0$  to  $i$  do
126:        $\sigma_{i,j} := \sigma_{j,i}$  {Že izračunane dolžine povezav samo prekopiramo.}
127:     end for
128:      $\sigma_{i,i} := 0$  {Razdalja do samega sebe je nič.}
129:     for  $j := i + 1$  to  $|\mathbf{P}_{g+1}|$  do
130:        $\sigma_{i,j} := \sum_{i=0}^M (f_i(\mathbf{x}_{g+1,i}) - f_i(\mathbf{x}_{g+1,j}))^2$  {Geometrijska razdalja,  $\Theta(M)$ .}
131:       if  $\sigma_{i,j} = 0$  then
132:          $n_{n;i,c_i} := j$  {Beležimo indekse točk, ki so podvojene: na indeksu  $i$ , zadnja ( $c_i$ ).}
133:          $c_i := c_i + 1$  {Štejemo število kopij enake točke.}
134:          $c_j := c_j + 1$  {Štejemo število kopij enake točke — tudi v nasprotni smeri.}
135:       end if
136:     end for
137:   end for
138:   for  $i := 0$  to  $|\mathbf{P}_{g+1}|$  do

```

```

139:    $F(\mathbf{p}_{g,i}) := \frac{R_{g,i} + 1}{\sigma_{i,k} + 2}$  {Metriko nakopičenosti izračunamo iz moči in inverza
      razdalje do  $k$ -tega najbližjega soseda.}
140: end for
141:  $\mathbf{P}_{g+1} := \emptyset$  {Tu se prične kopiranje primerkov v novo generacijo.}
142: for all  $\mathbf{p}_{g,i}$  with  $F(\mathbf{p}_{g,i}) < 1$  do
143:    $\mathbf{P}_{g+1} := \mathbf{P}_{g+1} \cup \mathbf{p}_{g,i}$  {Prekopiraj prvo fronto: vse nedominirane posameznike
      iz  $\mathbf{P}_g$  v  $\mathbf{P}_{g+1}$ .}
144:    $\mathbf{P}_g := \mathbf{P}_g \setminus \mathbf{p}_{g,i}$  {V stari generaciji primerke prve fronte odstranjujemo, da
      bo  $F(\mathbf{p}_{g,i}) \geq 1$ .}
145: end for
146: if  $|\mathbf{P}_{g+1}| < NP$  then
147:   Uredi  $\mathbf{P}_{g+1}$  naraščajoče po  $F(\mathbf{p}_{g,i})$  {Hitro urejanje:  $\Theta(|\mathbf{P}_{g+1}| \log |\mathbf{P}_{g+1}|)$ .}
148:    $i := 0$  {Boljši (tj. s čim manjšo nakopičenostjo  $F(\mathbf{p}_{g,i})$ ) primerki so po
      urejanju spredaj.}
149:   repeat
150:      $\mathbf{P}_{g+1} := \mathbf{P}_{g+1} \cup \mathbf{p}_{g,i}$ 
151:      $i := i + 1$  {Kopiramo v smeri od boljših primerkov k slabšim, do zapol-
      nitve arhiva.}
152:   until  $|\mathbf{P}_{g+1}| = NP$ 
153: else if  $|\mathbf{P}_{g+1}| > NP$  then
154:   repeat
155:      $k := 0$  {Poiščemo najbližji primerek  $\mathbf{p}_{g+1,i}$ :}
156:     for all  $\mathbf{p}_{g+1,i} \in \mathbf{P}_{g+1}$  do
157:       for all  $\mathbf{p}_{g+1,j} \in \mathbf{P}_{g+1}$  do
158:         if  $\forall 0 < k < |\mathbf{P}_{g+1}| : \sigma_i^k = \sigma_j^k \vee \exists 0 < l < |\mathbf{P}_{g+1}| :$ 
            $[(\forall 0 < l < k : \sigma_i^l = \sigma_j^l)] \wedge \sigma_i^k < \sigma_j^k$  then
159:            $\mathbf{P}_{g+1} := \mathbf{P}_{g+1} \setminus \mathbf{p}_{g+1,k}$  {Odstranimo najbolj nakopičen primerek,
             pogoj je  $\Theta(M)$ .}
160:           break for  $i$  {Dokler velikost presega  $NP$ , še izvajaj operator
             sekanja.}
161:         end if
162:       end for
163:     end for
164:   until  $|\mathbf{P}_{g+1}| = NP$ 
165: end if
166:    $g := g + 1$  {Povečaj števec generacij.}
167: end while

```

Dodatek B

Testne funkcije CEC 2007

Nabor testnih funkcij CEC 2007 [2] ($\mathbf{f}_i = \{f_{j=1..M}\}, i \in [1, 13]\}$) sestoji iz nekaterih novejših funkcij, kot sta funkciji OKA2 [119] in SYM-PART [120]. V nabor so vključene tudi nekoliko starejše funkcije, kot so ZDT [121] (Zitzler, Deb, Thiele), ki so bile nadgrajene z rotacijo in s skaliranjem (glej [2]). Naslednice funkcij ZDT, DTLZ [122] (Deb, Thiele, Laumanns, Zitzler) so prav tako vključene in rotirane ter skalirane.

Nabor testnih funkcij za obravnavano testno ogrodje je zaključen z do sedaj najboljšimi testnimi funkcijami za večkriterijsko optimizacijo, to je nabor WFG (Walking Fish Group) [95]. Te funkcije je možno precej parametrizirati in izbrati poljubne težavnosti. Edina njihova pomanjkljivost je še, da so brez omejitev. Za primer večkriterijskih funkcij z omejitvami lahko bralec prebere [123] (funkciji DTLZ8 in DTLZ9) in [124], ki uporablja še starejše funkcije od ZDT in DZLT.

Podajmo nekaj lastnosti uporabljenih funkcij. Ena od pomembnih lastnosti funkcije je, ali je funkcija ločljiva tako, da lahko vsak kriterij rešujemo posebej, brez medsebojnega vpliva na ostale kriterije (tj. ne sklepamo kompromisov). Takšne so funkcije S_ZDT1, S_ZDT2, S_ZDT4, S_ZDT6, S_DTLZ2, S_DTLZ3 in WFG1 ter prvi kriterij funkcije OKA2. Ostale funkcije (teh je 8) ima neločljivo povezane kriterijske funkcije. Nekatere funkcije imajo tudi le en ekstrem, tj. so unimodalne. Takšne so funkcije S_ZDT1, S_ZDT2, S_DTLZ2, WFG1 in WFG8 ter po en kriterij funkcij OKA2 in S_ZDT4.

Za vse vključene funkcije velja, da so brez ekstremnih parametrov (pri ekstremnih parametrih najdemo optimalno funkcijsko vrednost na robu iskalnega prostora spremenljivk), tudi brez medialnih (tj. rešitve na sredi iskalnega prostora). Za vse funkcije so znane analitične Pareto optimalne množice in fronte. Funkcije S_ZDT1, S_ZDT4, R_ZDT4 in delno WFG1, imajo konveksne Pareto fronte, druge so s konkavnimi Pareto frontami, razen WFG1, ki ima konveksne in mešane Pareto fronte. Funkcije SYM-PART in 7–19 imajo prav tako več regij iskalnih parametrov, ki svoje



rešitvene vektorje slikajo v iste vektorje v prostoru kriterijev (ang. *Pareto many-to-one*).

Funkcije WFG vsebujejo tudi nabore iskalnih parametrov, ki v prostoru kriterijev dajejo rešitve brez gradiента (ang. *flat regions*). Takšnih primerov ne moremo reševati s klasičnimi gradientnimi optimizacijskimi metodami, zato tu še posebej dobro do izraza pride uspešnost stohastičnega vzorčenja iskalnega prostora. Funkcija WFG9 vsebuje zavajajoč primer Pareto fronte, ki lahko algoritom zavede v napačno smer iskanja.

Tako funkcije DTLZ kot WFG so v testnem ogrodju podvojene. Prvič so vsebovane njihove lažje variante, z manjšim številom iskalnih parametrov ($D = 24$) in kriterijskih funkcij ($M = 3$), nato pa še z večjim številom iskalnih parametrov ($D = 28$) in kriterijskih funkcij ($M = 5$). Število iskalnih parametrov, število kriterijev in velikost aproksimacijske množice za vse dane funkcije prikazuje tabela B.1.

Tabela B.1: Nastavitev parametrov v testnih funkcijah [2].

Zaporedna številka	Oznaka (CEC 2007)	Ime funkcije	Število kriterijev (M)	Število parametrov (D)	Velikost aproksimacijske množice (\bar{N})
1.	f_1	OKA2	2	3	100
2.	f_2	SYM-PART	2	30	100
3.	f_3	S_ZDT1	2	30	100
4.	f_4	S_ZDT2	2	30	100
5.	f_5	S_ZDT4	2	30	100
6.	f_6	R_ZDT4	2	10	100
7.	f_7	S_ZDT6	2	30	100
8.	$f_8, M = 3$	S_DTLZ2_M3	3	30	150
9.	$f_9, M = 3$	R_DTLZ2_M3	3	30	150
10.	$f_{10}, M = 3$	S_DTLZ3_M3	3	30	150
11.	$f_{11}, M = 3$	WFG1_M3	3	24	150
12.	$f_{12}, M = 3$	WFG8_M3	3	24	150
13.	$f_{13}, M = 3$	WFG9_M3	3	24	150
14.	$f_8, M = 5$	S_DTLZ2_M5	5	30	800
15.	$f_9, M = 5$	R_DTLZ2_M5	5	30	800
16.	$f_{10}, M = 5$	S_DTLZ3_M5	5	30	800
17.	$f_{11}, M = 5$	WFG1_M5	5	28	800
18.	$f_{12}, M = 5$	WFG8_M5	5	28	800
19.	$f_{13}, M = 5$	WFG9_M5	5	28	800

Omejitev iskalnih parametrov, spodnjo mejo (\mathbf{x}_{min}) in zgornjo mejo (\mathbf{x}_{max}), za vsak iskalni parameter bralec najde v poročilu [2] in pripadajočem programskem kodu ogrodja za testiranje večkriterijskih optimizacijskih algoritmov. Tudi analitičnih izračunov Pareto množic, front in aproksimacijskih množic Pareto front tukaj ne podajamo, bralec tudi te lahko zasledi v [2].

B.1 Novejše funkcije

OKA2 (f_1)

$$f_1(\mathbf{x}) = x_1$$

$$f_2(\mathbf{x}) = 1 - \frac{1}{4\pi^2} (x_1 + \pi)^2 + |x_2 - 5 \cos(x_1)|^{\frac{1}{3}} + |x_3 - 5 \sin(x_1)|^{\frac{1}{3}}$$

SYM-PART (f_2)

$$f_1(\mathbf{x}) = \sum_{i=1}^{D/2} (x'_{2i} + a - t_1 c_2)^2 + (x'_{2i+1} - t_2 b)^2$$

$$f_2(\mathbf{x}) = \sum_{i=1}^{D/2} (x'_{2i} - a - t_1 c_2)^2 + (x'_{2i+1} - t_2 b)^2$$

kjer $\mathbf{x}' = \begin{pmatrix} \cos \omega & -\sin \omega & 0 & 0 & \dots \\ \sin \omega & \cos \omega & 0 & 0 & \dots \\ 0 & 0 & \cos \omega & -\sin \omega & \dots \\ 0 & 0 & \sin \omega & \cos \omega & \dots \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix} \mathbf{x}$

$$t_1 = \operatorname{sgn}(x'_1) \left\lceil \frac{|x'_1| - c_2/2}{c_2} \right\rceil, t_2 = \operatorname{sgn}(x'_2) \left\lceil \frac{|x'_2| - b/2}{b} \right\rceil$$

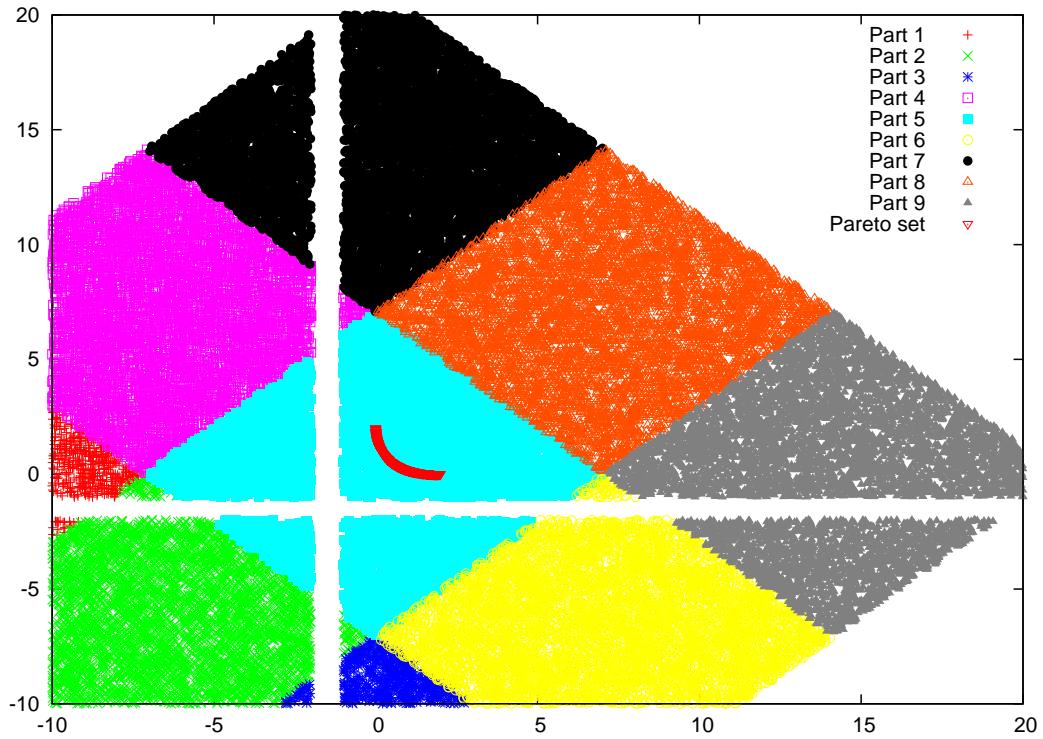
$$a = 1, b = 10, c = 8, c_2 = c + 2, a = 10, \omega = \frac{\pi}{4}$$

B.2 Funkcije ZDT in DTLZ

Rotacija in skaliranje je pri funkcijah ZDT in DTLZ potrebna zaradi nekaterih pomanjkljivosti prvotnih funkcij:

- vse vrednosti iskalnih parametrov so v globalnem optimumu enake,
- globalni optimum se nahaja v sredini iskalnega prostora,
- globalni optimum leži na mejah prostora kriterijev in
- vse funkcije so ločljive.

Originalne funkcije $\mathbf{f}(\mathbf{x}), \mathbf{x} \in [\mathbf{x}_{\min}, \mathbf{x}_{\max}]$ so zato premaknjene v $\mathbf{F}(\mathbf{z}), \mathbf{z} \in [\mathbf{z}_{\min}, \mathbf{z}_{\max}]$. Definirajmo dimenzijo iskalnega prostora kot D , tedaj je premaknjen vektor



Slika B.1: Ločene enakovredne Pareto množice za funkcijo SYM-PART.

v iskalnem prostoru enak $\mathbf{o} = [o_1, o_2, \dots, o_D]$, $\mathbf{d} = [d_1, d_2, \dots, d_D]$ razširjen obseg spodnje meje, $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_D]$ skalirni faktor parametrov in $\mathbf{p} = [p_1, p_2, \dots, p_D]$ kazenska vrednost. Posamezne dejanske vrednosti parametrov \mathbf{o} , \mathbf{d} in λ , uporabljene za funkcije ZDT in DTLZ, lahko pozoren bralec prebere v programskega kodu, ki spremlja poročilo [2].

Razširitev in premik iskalnega prostora

Da bi odpravili pomanjkljivosti pri ZDT, ki imajo globalni optimum na spodnji meji prostora spremenljivk ali v sredini iskalnega prostora, je spodnja meja prostora spremenljivk \mathbf{zmin} razširjena z \mathbf{d} . Po takšni transformaciji je funkcijsko vrednost enaka:

$$\mathbf{f}'(\mathbf{z}') = S(psum)(\mathbf{f}(\mathbf{z}') + f_{bias}),$$

kjer $z'_i = \begin{cases} z_i, & z_i \geq zmin_i \\ zmin_i + \lambda_i(zmin_i - z_i), & z_i < zmin_i \end{cases}$ in

$$S(psum) = \frac{2}{1 + \exp(-psum)}, \quad psum = \sqrt{\sum_{i \in \mathbf{I}} p_i^2},$$

$\mathbf{I} \subseteq \{1, 2, \dots, D\}$ (\mathbf{I} je množica vseh parametrov, vsebovanih v kriterijski funkciji $\mathbf{f}(\mathbf{x})$). Za te funkcije je uporabljen konstanten vektor λ , da iskalni prostor ne bi bil simetričen glede na vse iskalne parametre.

Skalirna funkcija S zagotavlja, da so vrednosti kriterijev v dodatno razširjenem intervalu vedno slabše od tistih v prvotnem intervalu, tj. Pareto optimalna fronta ostaja enaka. Ta trditev velja le, če $\mathbf{f} > \mathbf{0}$. Zato je \mathbf{f} premaknjena v $\mathbf{f} + \mathbf{f}_{bias}$, da bi s tem zagotovili, da so vse funkcijske vrednosti pozitivne. Zaloga vrednosti funkcije S je $[1, 2]$. Ko je funkcija v razširjenem intervalu blizu sredine intervala, bo $psum \rightarrow 0$ in $S \rightarrow 1$. Obratno, če bo funkcija daleč od sredine, bo $S \rightarrow 2$. Zaradi tega povečamo vrednost kriterija v razširjenem območju in funkcijo še vedno obdržimo zvezno. Kazenska vrednost p_i se za vsak parameter izračuna kot:

$$p_i = \begin{cases} 0, & z_i \geq zmin_i \\ \frac{|zmin_i - z_i|}{d_i}, & z_i < zmin_i \end{cases}, \quad i = 1, 2, \dots, D.$$

Po razširitvi iskalnega prostora iskalni parameter \mathbf{z} premaknemo še za vektor \mathbf{o} in dobimo funkcijo $\mathbf{F}(\mathbf{z}') = \mathbf{f}'(\mathbf{z})$, $\mathbf{z}' = \mathbf{z} + \mathbf{o}$. Po tem dobimo naslednjo funkcijo:

$$F_m(\mathbf{z}') = \begin{cases} f_m(\mathbf{z}') + 1, & \text{za vse } z_i \geq zmin_i \\ S(psum_m)(f_m(\mathbf{z}') + 1), & \text{sicer} \end{cases}, \quad m = 1, 2, \dots, M, \quad \mathbf{z}' = \mathbf{z} + \mathbf{o}.$$

Pri tem je $psum_m$ kazenska vrednost ob kršitvi definicijskega območja m -tega iskalnega parametra. V nadaljevanju bomo včasih uporabili še \mathbf{z}'_M , kar bo pomenilo le podvektor kršitev zadnjih $D - k$ iskalnih parametrov.

Glede na pravkar podan opis so bile razširjene in premaknjene funkcije ZDT1, ZDT2, ZDT4, ZDT6, DTLZ2 in DTLZ3. Iz teh so bile dobljene S_ZDT1, S_ZDT2, S_ZDT4, S_ZDT6, S_DTLZ2 in S_DTLZ3.

Rotacija

Iz enakih razlogov kot prenik je bila izvedena tudi rotacija funkcij:

$$F_m(\mathbf{x}) = \begin{cases} f_m(\mathbf{z}') + 1, & \text{za vse } xmin_i \geq x_i \geq xmax_i \\ S(psum_m)(f_m(\mathbf{z}') + 1), & \text{sicer} \end{cases}$$

$$m = 1, 2, \dots, M, \quad \mathbf{z} = \mathbf{M} - \mathbf{x},$$

kjer je \mathbf{M} ortogonalna matrika za linearne transformacije, kjer velja $\kappa(M) = 1$. Tukaj sta vrednosti z'_i in p'_i omejeni kot:

$$z'_i = \begin{cases} z_{min_i} + \lambda_i(z_{min_i} - z_i), & z_i < z_{min_i} \\ z_i, & z_{min_i} \geq z_i \geq z_{max_i} \\ z_{max_i} + \lambda_i(z_i - z_{max_i}), & z_i > z_{max_i} \end{cases}$$

in

$$p_i = \begin{cases} z_{min_i} - z_i, & z_i < z_{min_i} \\ 0, & z_{min_i} \geq z_i \geq z_{max_i}, \quad i = 1, 2, \dots, D \\ z_i - z_{max_i}, & z_i > z_{max_i}. \end{cases}$$

Na podlagi takšne transformacije sta bili razširjeni in rotirani funkciji ZDT4 in DTLZ2, iz katerih sta nastali R_ZDT4 in R_DTLZ2. Sledi nabor funkcij ZDT in DTLZ, ki so vključene v testno ogrodje CEC 2007.

S_ZDT1 (f_3)

$$f_1(\mathbf{z}') = \begin{cases} z'_1 + 1, & z_1 \geq 0 \\ S(p_1)(z'_1 + 1), & z_1 < 0 \end{cases}$$

$$f_2(\mathbf{z}') = \begin{cases} g(\mathbf{z}') [1 - \sqrt{z'_1/g(\mathbf{z}')} + 1, & z_i \geq 0 \\ S(\sqrt{\sum_{i=1}^D p_i^2})(g(\mathbf{z}') [1 - \sqrt{z'_1/g(\mathbf{z}')} + 1]), & \text{sicer} \end{cases}$$

$$g(\mathbf{z}') = 1 + 9(\sum_{i=2}^D z'_i)/(D-1)$$

kjer $z'_i = \begin{cases} z_i, & z_i \geq 0 \\ -\lambda_i z_i, & z_i < 0 \end{cases}$, $p_i = \begin{cases} 0, & z_i \geq 0 \\ |z_i|/d_i, & z_i < 0 \end{cases}$, $i = 1, 2, \dots, D$

S_ZDT2 (f_4)

$$f_1(\mathbf{z}') = \begin{cases} z'_1 + 1, & z_1 \geq 0 \\ S(p_1)(z'_1 + 1), & z_1 < 0 \end{cases}$$

$$f_2(\mathbf{z}') = \begin{cases} g(\mathbf{z}') [1 - (z'_1/g(\mathbf{z}'))^2] + 1, & z_i \geq 0 \\ S\left(\sqrt{\sum_{i=1}^D p_i^2}\right)(g(\mathbf{z}') [1 - (z'_1/g(\mathbf{z}'))^2] + 1), & \text{sicer} \end{cases}$$

$$g(\mathbf{z}') = 1 + 9\left(\sum_{i=2}^D z'_i\right)/(D-1)$$

kjer $z'_i = \begin{cases} z_i, & z_i \geq 0 \\ -\lambda_i z_i, & z_i < 0 \end{cases}, p_i = \begin{cases} 0, & z_i \geq 0 \\ |z_i|/d_i, & z_i < 0 \end{cases}, i = 1, 2, \dots, D$

S_ZDT4 (f₅)

$$f_1(\mathbf{z}') = \begin{cases} z'_1 + 1, & z_1 \geq 0 \\ S(p_1)(z'_1 + 1), & z_1 < 0 \end{cases}$$

$$f_2(\mathbf{z}') = \begin{cases} g(\mathbf{z}')[1 - \sqrt{z'_1/g(\mathbf{z}')}] + 1, & z_i \geq -5 \\ S\left(\sqrt{\sum_{i=1}^D p_i^2}\right)(g(\mathbf{z}')[1 - \sqrt{z'_1/g(\mathbf{z}')}] + 1), & \text{sicer} \end{cases}$$

$$g(\mathbf{z}') = 1 + 10(D-1) + \sum_{i=2}^D [z'^2_i - 10 \cos(4\pi z'_i)]$$

kjer $z'_1 = \begin{cases} z_1, & z_1 \geq 0 \\ -\lambda_1 z_1, & z_1 < 0 \end{cases}, z'_i = \begin{cases} z_i, & z_i \geq -5 \\ -5 - \lambda_i(z_i + 5), & z_i < -5 \end{cases},$

$$p_1 = \begin{cases} 0, & z_1 \geq 0 \\ |z_1|/d_1, & z_1 < 0 \end{cases}, p_i = \begin{cases} 0, & z_i \geq -5 \\ |z_i + 5|/d_i, & z_i < -5 \end{cases}, i = 2, \dots, D$$

R_ZDT4 (f₆)

$$f_1(\mathbf{z}') = \begin{cases} z'_1 + 1, & z_1 \geq 0 \\ S(p_1)(z'_1 + 1), & z_1 < 0 \end{cases}$$

$$f_2(\mathbf{z}') = \begin{cases} g(\mathbf{z}')[1 - \sqrt{z'_1/g(\mathbf{z}')}] + 1, & z_i \geq -5 \\ S\left(\sqrt{\sum_{i=1}^D p_i^2}\right)(g(\mathbf{z}')[1 - \sqrt{z'_1/g(\mathbf{z}')}] + 1), & \text{sicer} \end{cases}$$

$$g(\mathbf{z}') = 1 + 10(D-1) + \sum_{i=2}^D [z'^2_i - 10 \cos(4\pi z'_i)]$$

kjer $z'_1 = \begin{cases} -\lambda_1 z_1, & z_1 < 0 \\ z_1, & 0 \leq z_1 \leq 1 \\ \lambda_1 z_1, & z_1 > 1 \end{cases}, z'_i = \begin{cases} -5 - \lambda_i(z_i + 5), & z_i < -5 \\ z_i, & -5 \leq z_i \leq 5 \\ 5 - \lambda_i(z_i - 5), & z_i > 5 \end{cases},$

$$p_1 = \begin{cases} -z_1, & z_1 < 0 \\ 0, & 0 \leq z_1 \leq 1 \\ z_1 - 1, & z_1 > 1 \end{cases}, p_i = \begin{cases} -5 - z_i, & z_i < -5 \\ 0, & -5 \leq z_i \leq 5 \\ z_i - 5, & z_i > 5 \end{cases}, i = 2, \dots, D$$

S_ZDT6 (f₇)

$$f_1(\mathbf{z}') = \begin{cases} 1 - \exp(-4z'_1) \sin^6(6\pi z'_1) + 1, & z_1 \geq 0 \\ S(p_1)(1 - \exp(-4z'_1) \sin^6(6\pi z'_1) + 1), & z_1 < 0 \end{cases}$$

$$f_2(\mathbf{z}') = \begin{cases} g(\mathbf{z}') [1 - (z'_1/g(\mathbf{z}'))^2] + 1, & z_i \geq 0 \\ S(\sqrt{\sum_{i=1}^D p_i^2})(g(\mathbf{z}') [1 - (z'_1/g(\mathbf{z}'))^2] + 1), & \text{sicer} \end{cases}$$

$$g(\mathbf{z}') = 1 + 9 \left[\left(\sum_{i=2}^D z'_i \right) / (D-1) \right]^{1/4}$$

$$\text{kjer } z'_i = \begin{cases} z_i, & z_i \geq 0 \\ -\lambda_i z_i, & z_i < 0 \end{cases}, p_i = \begin{cases} 0, & z_i \geq 0 \\ |z_i|/d_i, & z_i < 0 \end{cases}, i = 1, 2, \dots, D$$

S_DTLZ2 (f₈)

$$f_1(\mathbf{z}') = \begin{cases} (1 + g(\mathbf{z}'_M)) \prod_{i=1}^{M-1} (\cos(z'_i \pi/2)) + 1, & z_i \geq 0 \\ S(psum_1)((1 + g(\mathbf{z}'_M)) \prod_{i=1}^{M-1} (\cos(z'_i \pi/2)) + 1), & \text{sicer} \end{cases}$$

$$f_2(\mathbf{z}') = \begin{cases} (1 + g(\mathbf{z}'_M)) \prod_{i=1}^{M-2} (\cos(z'_i \pi/2)) \sin(z'_{M-1} \pi/2) + 1, & z_i \geq 0 \\ S(psum_2)((1 + g(\mathbf{z}'_M)) \prod_{i=1}^{M-2} (\cos(z'_i \pi/2)) \sin(z'_{M-1} \pi/2) + 1), & \text{sicer} \end{cases}$$

$$f_3(\mathbf{z}') = \begin{cases} (1 + g(\mathbf{z}'_M)) \prod_{i=1}^{M-3} (\cos(z'_i \pi/2)) \sin(z'_{M-2} \pi/2) + 1, & z_i \geq 0 \\ S(psum_3)((1 + g(\mathbf{z}'_M)) \prod_{i=1}^{M-3} (\cos(z'_i \pi/2)) \sin(z'_{M-2} \pi/2) + 1), & \text{sicer} \end{cases}$$

$$\vdots$$

$$f_{M-1}(\mathbf{z}') = \begin{cases} (1 + g(\mathbf{z}'_M)) \cos(z'_1 \pi/2) \sin(z'_2 \pi/2) + 1, & z_i \geq 0 \\ S(psum_{M-1})((1 + g(\mathbf{z}'_M)) \cos(z'_1 \pi/2) \sin(z'_2 \pi/2) + 1), & \text{sicer} \end{cases}$$

$$f_M(\mathbf{z}') = \begin{cases} (1 + g(\mathbf{z}'_M)) \sin(z'_1 \pi/2) + 1, & z_i \geq 0 \\ S(psum_M)((1 + g(\mathbf{z}'_M)) \sin(z'_1 \pi/2) + 1), & \text{sicer} \end{cases}$$

$$g(\mathbf{z}'_M) = \sum_{i=k+1}^D (z'_i - 1/2)^2$$

$$\text{kjer } z'_i = \begin{cases} z_i, & z_i \geq 0 \\ -\lambda_i z_i, & z_i < 0 \end{cases}, p_i = \begin{cases} 0, & z_i \geq 0 \\ |z_i|/d_i, & z_i < 0 \end{cases}, i = 1, 2, \dots, D$$

R_DTLZ2 (f₉)

$$\begin{aligned}
f_1(\mathbf{z}') &= \begin{cases} (1 + g(\mathbf{z}'_M)) \prod_{i=1}^{M-1} (\cos(z'_i \pi / 2)) + 1, & z_i \geq 0 \\ S(psum_1)((1 + g(\mathbf{z}'_M)) \prod_{i=1}^{M-1} (\cos(z'_i \pi / 2)) + 1), & \text{sicer} \end{cases} \\
f_2(\mathbf{z}') &= \begin{cases} (1 + g(\mathbf{z}'_M)) \prod_{i=1}^{M-2} (\cos(z'_i \pi / 2)) \sin(z'_{M-1} \pi / 2) + 1, & z_i \geq 0 \\ S(psum_2)((1 + g(\mathbf{z}'_M)) \prod_{i=1}^{M-2} (\cos(z'_i \pi / 2)) \sin(z'_{M-1} \pi / 2) + 1), & \text{sicer} \end{cases} \\
f_3(\mathbf{z}') &= \begin{cases} (1 + g(\mathbf{z}'_M)) \prod_{i=1}^{M-3} (\cos(z'_i \pi / 2)) \sin(z'_{M-2} \pi / 2) + 1, & z_i \geq 0 \\ S(psum_3)((1 + g(\mathbf{z}'_M)) \prod_{i=1}^{M-3} (\cos(z'_i \pi / 2)) \sin(z'_{M-2} \pi / 2) + 1), & \text{sicer} \end{cases} \\
&\vdots \\
f_{M-1}(\mathbf{z}') &= \begin{cases} (1 + g(\mathbf{z}'_M)) \cos(z'_1 \pi / 2) \sin(z'_2 \pi / 2) + 1, & z_i \geq 0 \\ S(psum_{M-1})((1 + g(\mathbf{z}'_M)) \cos(z'_1 \pi / 2) \sin(z'_2 \pi / 2) + 1), & \text{sicer} \end{cases} \\
f_M(\mathbf{z}') &= \begin{cases} (1 + g(\mathbf{z}'_M)) \sin(z'_1 \pi / 2) + 1, & z_i \geq 0 \\ S(psum_M)((1 + g(\mathbf{z}'_M)) \sin(z'_1 \pi / 2) + 1), & \text{sicer} \end{cases} \\
g(\mathbf{z}'_M) &= \sum_{i=k+1}^D (z'_i - 1/2)^2 \\
\text{kjer } z'_i &= \begin{cases} -\lambda_i z_i, & z_i < 0 \\ z_i, & 0 \leq z_i \leq 1 \\ \lambda_i z_i, & z_i > 1 \end{cases}, p_i = \begin{cases} -z_i, & z_i < 0 \\ 0, & 0 \leq z_i \leq 1 \\ z_i - 1, & z_i > 1 \end{cases}, i = 1, 2, \dots, D
\end{aligned}$$

S_DTLZ3 (f₁₀)

$$\begin{aligned}
f_1(\mathbf{z}') &= \begin{cases} (1 + g(\mathbf{z}'_M)) \prod_{i=1}^{M-1} (\cos(z'_i \pi / 2)), & z_i \geq 0 \\ S(psum_1)((1 + g(\mathbf{z}'_M)) \prod_{i=1}^{M-1} (\cos(z'_i \pi / 2)) + 1), & \text{sicer} \end{cases} \\
f_2(\mathbf{z}') &= \begin{cases} (1 + g(\mathbf{z}'_M)) \prod_{i=1}^{M-2} (\cos(z'_i \pi / 2)) \sin(z'_{M-1} \pi / 2), & z_i \geq 0 \\ S(psum_2)((1 + g(\mathbf{z}'_M)) \prod_{i=1}^{M-2} (\cos(z'_i \pi / 2)) \sin(z'_{M-1} \pi / 2) + 1), & \text{sicer} \end{cases} \\
f_3(\mathbf{z}') &= \begin{cases} (1 + g(\mathbf{z}'_M)) \prod_{i=1}^{M-3} (\cos(z'_i \pi / 2)) \sin(z'_{M-2} \pi / 2), & z_i \geq 0 \\ S(psum_3)((1 + g(\mathbf{z}'_M)) \prod_{i=1}^{M-3} (\cos(z'_i \pi / 2)) \sin(z'_{M-2} \pi / 2) + 1), & \text{sicer} \end{cases} \\
&\vdots \\
f_{M-1}(\mathbf{z}') &= \begin{cases} (1 + g(\mathbf{z}'_M)) \cos(z'_1 \pi / 2) \sin(z'_2 \pi / 2), & z_i \geq 0 \\ S(psum_{M-1})((1 + g(\mathbf{z}'_M)) \cos(z'_1 \pi / 2) \sin(z'_2 \pi / 2) + 1), & \text{sicer} \end{cases} \\
f_M(\mathbf{z}') &= \begin{cases} (1 + g(\mathbf{z}'_M)) \sin(z'_1 \pi / 2), & z_i \geq 0 \\ S(psum_M)((1 + g(\mathbf{z}'_M)) \sin(z'_1 \pi / 2) + 1), & \text{sicer} \end{cases}
\end{aligned}$$

$$g(\mathbf{z}'_M) = 100(|\mathbf{z}'_M| + \sum_{i=k+1}^D (z_i - 1/2)^2 - \cos(20\pi(z_i - 1/2)))$$

kjer $z'_i = \begin{cases} z_i, & z_i \geq 0 \\ -\lambda_i z_i, & z_i < 0 \end{cases}, p_i = \begin{cases} 0, & z_i \geq 0 \\ |z_i|/d_i, & z_i < 0 \end{cases}, i = 1, 2, \dots, D$

B.3 Funkcije WFG

Pri funkcijah WFG v splošnem minimiziramo M , $D \geq M$ kriterijev in ob tem iskalne parametre $\mathbf{x} = x_1 \dots x_D$, razdelimo na *parametre, vezane na položaj* (ang. **position related parameters**) $\mathbf{y} = y_{1\dots k} = \{x_{i=1\dots k}\}$ in *parametre, vezane na razdaljo* (ang. **distance related parameters**) $\mathbf{z} = z_{1\dots D-k} = \{x_{i=k+1\dots D}\}$. Tukaj je izbran $k = 2(M-1)$, tj. $k = 4$ za $M = 3$ in $k = 8$ za $M = 5$ ter $D - k = 20$ za vse funkcije WFG. Parametre \mathbf{z} dobimo s transformacijami vektorja \mathbf{y} , tako da \mathbf{z} po transformacijah vsebuje naslednje komponente:

$$\mathbf{z} = \{\max(t_M^p, A_1)(t_1^p - 0.5) + 0.5, \dots, \max(t_M^p, A_M - 1)(t_{M-1}^p - 0.5) + 0.5, t_M\},$$

kjer so $A_{i=1..M-1} \in \{0, 1\}$ konstante za vključevanje/izključevanje dimenzij in \mathbf{t}^p vmesni vektorji, ki jih dobimo s **transformacijami funkcijami** (ang. transformation functions) [95] s preslikavo parametrov, vezanih na položaj, \mathbf{y} :

$$\begin{aligned} \text{b_poly}(y, \alpha) &= y^\alpha \\ \text{b_flat}(y, A, B, C) &= A + \min(0, \lfloor y - B \rfloor) \frac{A(B-y)}{B} - \min(0, \\ &\quad \lfloor C - y \rfloor) \frac{(1-A)(y-C)}{1-C} \\ \text{b_param}(y, \mathbf{y}', A, B, C) &= y^{B+(C-B)v(u(\mathbf{y}'))}, \\ &\quad \text{kjer } v(u(\mathbf{y}')) = A - (1 - 2u(\mathbf{y}')) \lfloor 0, 5 - u(\mathbf{y}') \rfloor + A \\ \text{s_linear}(y, A) &= \frac{|y-A|}{|\lfloor A-y \rfloor + A|} \\ \text{s_decept}(y, A, B, C) &= 1 + (|y - A| - B) \left(\frac{\lfloor y - A + B \rfloor (1 - C + \frac{A-B}{B})}{A - B} + \right. \\ &\quad \left. \frac{\lfloor A + B - y \rfloor (1 - C + \frac{1 - A - B}{B})}{1 - A - B} + \frac{1}{B} \right) \\ \text{s_multi}(y, A, B, C) &= \frac{1 + \cos[(4A+2)\pi(0.5 - \frac{|y-C|}{2(\lfloor C-y \rfloor + C)})] + 4B(\frac{|y-C|}{2(\lfloor C-y \rfloor + C)})^2}{B+2} \\ \text{r_sum}(\mathbf{y}, \mathbf{w}) &= \left(\sum_{i=1}^{|\mathbf{y}|} w_i y_i \right) / \sum_{i=1}^{|\mathbf{y}|} w_i \\ \text{r_nonsep}(\mathbf{y}, A) &= \frac{\sum_{j=1}^{|\mathbf{y}|} (y_j + \sum_{k=0}^{A-2} |y_j - y_{1+(j+k) \bmod |\mathbf{y}|}|)}{\frac{|\mathbf{y}|}{A} \lceil A/2 \rceil (1 + 2A - 2\lceil A/2 \rceil)} \end{aligned}$$

ter nato iz normaliziranega vektorja \mathbf{x} ($\forall x_i \in [0, 1]$) najprej izračunamo \mathbf{t}^1 , ki služi kot vhod za izračun \mathbf{t}^2 in tako analogno, do \mathbf{t}^p .

Z dobljenim vektorjem parametrov, vezanih na razdaljo, \mathbf{z} lahko sedaj zapišemo

kriterijske funkcije $\mathbf{f}(\mathbf{z}) = \{f_{m=1..M}(\mathbf{z})\}$, ki so naslednje oblike:

$$f_m(\mathbf{z}) = D_s z_M + S_m h_m(z_1, \dots, z_{M-1}), \quad m \in \{1..M\}$$

kjer je $D_s = 1$ skalirna konstanta razdalj in tudi $S_m = 2m$ skalirne konstante. Oblikovne funkcije h_m določajo tip funkcije za preslikavo parametrov, vezanih na razdaljo, \mathbf{z} . Tukaj uporabljeni obliki so konveksna, konkavna in mešana:

$$\begin{aligned} \text{convex}_1(z_{i=1:M-1}) &= \prod_{i=1}^{M-1} (1 - \cos(z_i \pi / 2)) \\ \text{convex}_{m=2:M-1}(z_{i=1:M-1}) &= (\prod_{i=1}^{M-m} (1 - \cos(z_i \pi / 2))) (1 - \sin(z_{M-m+1} \pi / 2)) \\ \text{convex}_M(z_{i=1:M-1}) &= 1 - \sin(z_1 \pi / 2) \\ \text{concave}_1(z_{i=1:M-1}) &= \prod_{i=1}^{M-1} (1 - \sin(z_i \pi / 2)) \\ \text{concave}_{m=2:M-1}(z_{i=1:M-1}) &= (\prod_{i=1}^{M-m} (1 - \sin(z_i \pi / 2))) (1 - \cos(z_{M-m+1} \pi / 2)) \\ \text{concave}_M(z_{i=1:M-1}) &= \cos(z_1 \pi / 2) \\ \text{mixed}_M(z_{i=1:M-1}) &= (1 - z_1 - \frac{\cos(2A\pi z_1 + \pi/2)}{2A\pi})^\alpha, \end{aligned}$$

kjer α določa tip oblike in $A \in \{1, 2, \dots\}$ število segmentov mešanih oblik.

WFG1 (\mathbf{f}_{11})

Oblika $h_{m=1:M-1} = \text{convex}_m$

$$\begin{aligned} h_M &= \text{mixed}_M(z \ \alpha = 1 \text{ in } A = 5) \\ t^1 &\quad t^1_{i=1:k} = y_i \\ t^1 &\quad t^1_{i=k+1:D} = \text{s_linear}(y_i, 0, 35) \\ t^2 &\quad t^2_{i=1:k} = y_i \\ t^2 &\quad t^2_{i=k+1:D} = \text{b_flat}(y_i, 0, 8, 0, 75, 0, 85) \\ t^3 &\quad t^3_{i=1:n} = \text{b_poly}(y_i, 0, 02) \\ t^4 &\quad t^4_{i=1:M-1} = \text{r_sum}(\{y_{(i-1)k/(M-1)+1}, \dots, y_{ik/(M-1)}\}, \\ &\quad \quad \quad \{2((i-1)k/(M-1) + 1), \dots, 2ik/(M-1)\}) \\ t^4 &\quad t^4_M = \text{r_sum}(\{y_{k+1}, \dots, y_n\}, \{2(k+1), \dots, 2n\}) \end{aligned}$$

WFG8 (f₁₂)

Oblika $h_{m=1:M} = \text{concave}_m$

$$\begin{aligned} t^1 & \quad t^1_{i=1:k} = y_i \\ t^1_{i=k+1:D} & = \text{b_param}(y_i, \text{r_sum}(\{y_1, \dots, y_{i-1}\}, \{1, \dots, 1\}), \\ & \quad \frac{0,98}{49,98}, \quad 0,02, \quad 50) \\ t^2 & \quad t^2_{i=1:k} = y_i \\ t^2_{i=k+1:D} & = \text{s_linear}(y_i, 0, 35) \\ t^3 & \quad t^3_{i=1:M-1} = \text{r_sum}(y_{(i-1)k/(M-1)+1}, \dots, y_{ik/(M-1)}), \{1, \dots, 1\}) \\ t^3_M & = \text{r_sum}(\{y_{k+1}, \dots, y_D\}, \{1, \dots, 1\}) \end{aligned}$$
WFG9 (f₁₃)

Oblika $h_{m=1:M} = \text{concave}_m$

$$\begin{aligned} t^1 & \quad t^1_{i=1:D-1} = \text{b_param}(y_i, \text{r_sum}(\{y_{i+1}, \dots, y_D\}, \{1, \dots, 1\}), \\ & \quad \frac{0,98}{49,98}, \quad 0,02, \quad 50) \\ t^1_D & = y_D \\ t^2 & \quad t^2_{i=1:k} = \text{s_decept}(y_i, 0, 35, 0, 001, 0, 05) \\ t^2_{i=k+1:D} & = \text{s_multi}(y_i, 30, 95, 0, 35) \\ t^3 & \quad t^3_{i=1:M-1} = \text{r_nonsep}(\{y_{(i-1)k/(M-1)+1}, \dots, y_{ik/(M-1)}\}, k/(M-1)) \\ t^3_M & = \text{r_nonsep}(\{y_{k+1}, \dots, y_D\}, l) \end{aligned}$$

Dodatek C

Ogrodje za paralelno procesiranje

Zagon testnih funkcij in indikatorjev kakovosti smo izvedli s pomočjo paralelnega procesiranja. Faze so si sledile v naslednjem vrstnem redu: reševanje testnih funkcij, ocenjevanje dobljenih rezultatov in izdelava tabel. Za izdelavo ogrodja smo uporabili UNIX orodja ukazne lupine za obdelavo tokov (bash, awk, sed, grep, paste), izdelavo slik (xfig), grafov (gnuplot) in dokumentov (\LaTeX).

Postopke za oceno kakovosti večkriterijskih iskalnih algoritmov smo vključili v naš programski sistem, ki nanaiza več heterogenih sistemov za obdelavo podatkov in izdela dokument PDF.

Ogrodje za izdelavo tabel z rezultati ima tri faze, ki imajo tudi lastne nabore funkcionalnosti:

- *test* – požene teste in shrani rezultate testov,
- *metrics* – nad rezultati testov v mapi `test` požene metrike in shrani njihove vrednosti in
- *latex* – vrednosti metrik vstavi v \LaTeX dokument in ga prevede v dokument formata PDF.

V vseh treh fazah veljajo preprosta pravila:

- skripte, ki jih faza obdelave potrebuje, so shranjeni v njeni mapi `bin`,
- rezultati se shranjujejo v mapo `output`,
- vhodni podatki se jemljejo iz mape `output` iz prejšnje faze, parametrizacija trenutne faze pa je shranjena v mapi `input`.

C.1 Faza *test*

Faza *test* uporablja naslednje skripte:

- **testAll.sh** – paralelno požene teste vseh funkcij iz datoteke `input/functs` in meri čas izvajanja, ter počaka na zaključek vseh funkcij, vsaka skripta `testOneFunction.sh` se izvaja paralelno,
- **testAll.sh.all** – neparalelna izvedba zagona testov funkcij,
- **testOneFunction.sh** – testira posamezno funkcijo za vseh 25 zagonov in shranjuje dnevnik (ang. *log*), ki ga program izpisuje (npr. dinamika krmilnih parametrov F in CR),
- **prepareParameters.sh** – pripravi vhodne parametre optimizatorja tako, da nastavi seme, selekcijsko strategijo, velikost populacije (to vzame iz datoteke `input/functs_appset_size`) ter izhodne datoteke,
- **demosa.x86_64** – optimizator za arhitekturo EM64T (x86-64), parametrizira se z datoteko `ini_file.txt`, rezultate aproksimiranih Pareto front shranjuje v `output/<ime funkcije>/front_<ime-funcije>-<št-zagona>`, pri čemer je ime funkcije npr. OKA2, št. zagona pa med 1 in 25 in dejansko določa začetno vrednost generatorja psevdonaključnih števil,
- **demosa.i686** – optimizator za arhitekturo Pentium 4 in naprej,
- **acquire.sh** – pridobi optimizator, tj. binarni, optimizirano prevedeni program za uporabljano arhitekturo,
- **Makefile** – požene program za pridobitev optimizatorja.

C.2 Faza *metrics*

Fazo *test* sestavlja naslednje skripte:

- **assessmentAll.sh** – paralelno izračuna vse metrike za vse funkcije, počaka, nato nariše 2D projekcije aproksimacijskih množic, 3D ploskve dosega, spet počaka na zaključek in nato izračuna še vse metrike hipervolumna, ki trajajo najdlje (približno 5 ur),
- **assessOne.sh** – predprocesira referenčne množice (spremeni tabulatorje v presledke) in ga predela v obliko, skladno z ogrodjem PISA, kriterije normalizira, filtrira dominirane rešitve, nato pa predhodno izračuna metriko hipervolumna



za podane referenčne množice funkcij; sledi enako za izhod iz optimizatorja (aproksimacijske množice v `output/<ime funkcije>/front_<ime-funkcije>-<št-zagona>`), kjer se uporabijo obdelani podatki za referenčne množice; sledi klic PISA ogrodja za izločitev točk za ploskve dosega, na koncu pa se izračunajo vse metrike za aproksimacijske množice, rezultati pa shranijo v mapo `output`,

- `assessHypIndOne.sh` – za vse tri dele (0, 1, 2) izračuna metriko hipervolumna,
- `covered_sets.x86_64` oz.
- `covered_sets.i686` – orodje iz ogrodja PISA za izračun metrike CS,
- `eaf.i686` oz.
- `eaf.x86_64` – orodje iz ogrodja PISA za izračun točk v ploskvi dosega,
- `filter.i686` oz.
- `filter.x86_64` – orodje iz ogrodja PISA za filtriranje (izločanje) dominiranih točk znotraj podane množice,
- `normalize.i686` oz.
- `normalize.x86_64` – orodje iz ogrodja PISA za normalizacijo kriterijev,
- `hyp_ind.i686` oz.
- `hyp_ind.x86_64` – orodje iz ogrodja PISA za izračun metrike hipervolumna,
- `r_ind.i686` oz.
- `r_ind.x86_64` – orodje iz ogrodja PISA za izračun metrike R,
- `plot-attainments2.i686` oz.
- `plot-attainments2.x86_64` – orodje iz ogrodja PISA, ki pripravi točke za izris ploskev dosega
- `draw2Dsurfaces.sh` – s pomočjo ustreznega orodja iz ogrodja PISA zbere, nato pa nariše 2D ploskve dosega,
- `draw3Dplots.sh` – s pomočjo ustreznega orodja iz ogrodja PISA zbere, nato pa nariše 3D ploskve dosega,
- `draw5DFrontsWith2DprojectionOfMedian.sh` – izriše projekcije po metriki R medialnih aproksimacijskih množic,

- `lsTime.sh` – skripta za izračun časovne zahtevnosti,
- `Makefile` – zbirka uporabe skript, ki jih tudi medsebojno poveže v časovno zaporedje
- `makeParts.sh` – razdeli zbirko aproksimacijskih množic v ločene zbirke (0, 1, 2),
- `notabs.sh` – skripta, ki spremeni tabulatorje v presledke,

C.3 Faza *latex*

Faza *latex* v predlogo dokumenta vstavi obdelane rezultate simulacij, pri tem pa uporabi še naslednje skripte:

- `makeAllStats.sh` – pripravi statistike za vsako od metrik,
- `createTables.sh` – kreira tabele iz obdelanih podatkov, ki jih pripravi skripta `makeAllStats.sh`,
- `create-F-CR-stats.sh` – predprocesira dnevnik izhoda programa in izlušči informacije za dinamiko *F* in *CR*,
- `Makefile` – prevede program za izračun statistik,
- `preprocessTestLogs-FCR.sh` – predprocesira izpis iz programa, da izloči parametre *F* in *CR* po generacijah,
- `stats.cpp` – pripravi statistike (povprečja, standardna deviacija, minimum, maksimum),
- `stats.i686` oz. `stats.x86_64` – pripravi statistike (povprečja, standardna deviacija, minimum, maksimum)
- `subtract` – binarni kod za računanje t-testov,
- `subtract.cpp` – izvorni kod za računanje t-testov,
- `subtractStats.sh` – tvorjenje tabel za t-teste iz dveh naborov izračunanih metrik.

Dodatek D

Slovar

Angleško	Slovensko
approximation set	aproksimacijska množica
ant colony optimization	optimizacija s kolonijo mravov
attainment surface	ploskev dosega
50% attainment surface	ploskev 50% dosega
50 percent attainment surface	ploskev 50-odstotnega dosega
bias	nagnjenost
candidate solution	kandidatna rešitev
competition	tekmovanje
computational intelligence	računska inteligenca
concave set	konkavna množica
covered sets	pokrite množice
connected set	povezana množica
convex set	konveksna množica
deceptive problem	zavajajoč problem
degenerate front	degenerirana fronta
density	gostota
differential evolution	diferencialna evolucija
difference vector	diferenčni vektor
discontinuous set	nepovezana množica
donor vector	mutiran vektor
evolution strategies	evolucijske strategije
evolutionary computation	evolucijsko računanje
evolutionary algorithm	evolucijski algoritem
extremal parameter	ekstremen parameter, blizu roba domene
fitness landscape	pokrajina uspešnosti
fitness space	prostor kriterijev
flat region	ravno območje

front	fronta
generation	generacija
incomparable	neprimerljiv
iteration	iteracija
isolated optimum	osamljen optimum
linear set	linearna množica
medial parameter	osrednji parameter, blizu mediane domene
mixed front	fronta iz striktno konveksnih, konkavnih ali linearnih primerov aproksimacijske množice
multimodal	večmodalno, z več ekstremi
multiobjective optimization	večkriterijska optimizacija
mutation	mutacija
mutated vector	mutiran vektor
nondominated front	nedominirana fronta
nonseparable	neločljiv
objective	kriterij
optimal set	optimalna množica
particle swarm optimization	optimizacija z rojem delcev
Pareto many-to-one	več Pareto optimalnih parametrov v isto vrednost Pareto optimalnega kriterija
Pareto optimality	Pareto optimalnost
Pareto optimal front	Pareto optimalna fronta
prunning	rezanje
raw fitness	groba uspešnost
reproduction	reprodukција
search space	iskalni prostor
scalable objective-wise	razširljiv po kriterijih
scalable parameter-wise	razširljiv po iskalnih parametrih
selection	selekcija
separable	ločljiv
strictly concave set	stogo konkavna množica
strictly convex set	stogo konveksna množica
test problem	testni problem
test suite	testno ogrodje
target vector	ciljni vektor
trial vector	poskusni vektor
weighted sum method	metoda utežene vsote
weighted difference vector	utežen diferenčni vektor
unimodal	unimodalno, z enim ekstremom
ϵ -constraint method	metoda ϵ -omejitev

Literatura

- [1] R. Storn in K. Price, “Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces,” *Journal of Global Optimization*, letn. 11, str. 341–359, 1997.
- [2] V. L. Huang, A. K. Qin, K. Deb, E. Zitzler, P. N. Suganthan, J. J. Liang, M. Preuss in S. Huband, “Problem Definitions for Performance Assessment & Competition on Multi-objective Optimization Algorithms,” Nanyang Technological University et. al., Singapore, tehn. por. TR-07-01, 2007.
- [3] W. Duch in J. Mandziuk, “Quo Vadis Computational Intelligence?” v *Machine Intelligence: Quo Vadis?*, ser. Advances in Fuzzy Systems – Applications and Theory, K. H. P. Sincak, J. Vascak, Ur. World Scientific, letn. 21, str. 3–28, 2004.
- [4] K. Deb, M. Mohan in S. Mishra, “Evaluating the ϵ -Domination Based Multi-Objective Evolutionary Algorithm for a Quick Computation of Pareto-Optimal Solutions,” *Evolutionary Computation*, letn. 13, št. 4, str. 501–525, 2005.
- [5] K. Deb in H. Gupta, “Introducing robustness in multi-objective optimization,” *Evolutionary Computation*, letn. 14, št. 4, str. 463–494, 2006.
- [6] A. E. Eiben, R. Hinterding in Z. Michalewicz, “Parameter Control in Evolutionary Algorithms,” *IEEE Trans. on Evolutionary Computation*, letn. 3, št. 2, str. 124–141, 1999.
- [7] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. Chichester, UK: John Wiley & Sons, ISBN 0-471-87339-X, 2001.
- [8] D. Büche, P. Stoll, R. Dornberger in P. Kourmoursakos, “Multiobjective Evolutionary Algorithm for the Optimization of Noisy Combustion Processes,” *IEEE Transactions on Systems, Man, and Cybernetics Part C—Applications and Reviews*, letn. 32, št. 4, str. 460–473, 2002.
- [9] L. Barone, L. While in P. Hingston, “Designing Crushers with a Multi-Objective Evolutionary Algorithm,” v *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002)*, W. Langdon, E. Cantú-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull,

- M. Potter, A. Schultz, J. Miller, E. Burke in N. Jonoska, ured. San Francisco, California: Morgan Kaufmann Publishers, str. 995–1002, 2002.
- [10] J. de la Cruz, B. de Andres-Toro, A. Herrán, E. B. Porta in P. F. Blanco, “Multi-objective Optimization of the Transport in Oil Pipelines,” v *Proceedings of the 9th IEEE International Conference on Emerging Technologies and Factory Automation*, letn. 1. Lisbon, Portugal: IEEE, str. 566–573, 2003.
- [11] P. Engrand, “A multi-objective optimization approach based on simulated annealing and its application to nuclear fuel management,” v *Proceedings of the Fifth International Conference on Nuclear Engineering*. Nice, France: American Society of Mechanical Engineering, str. 416–423, 1997.
- [12] K. J. Shaw, A. L. Nortcliffe, M. Thompson, J. Love, C. M. Fonseca in P. J. Fleming, “Assessing the Performance of Multiobjective Genetic Algorithms for Optimization of a Batch Process Scheduling Problem,” v *1999 Congress on Evolutionary Computation*. Washington, D.C.: IEEE Service Center str. 37–45, 1999.
- [13] S. Watanabe, T. Hiroyasu in M. Miki, “Parallel Evolutionary Multi-Criterion Optimization for Mobile Telecommunication Networks Optimization,” v *Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems. Proceedings of the EUROGEN 2001. Athens. Greece, September 19-21*, K. Giannakoglou, D. Tsahalis, J. Periaux, K. Papailiou in T. Fogarty, ured. Baracelona, Spain: International Center for Numerical Methods in Engineering (CIMNE), str. 167–172, 2001.
- [14] E. J. Hughes, “Swarm Guidance using a Multi-Objective Co-evolutionary On-Line Evolutionary Algorithm,” v *2004 Congress on Evolutionary Computation (CEC 2004)*, letn. 2. Portland, Oregon, USA: IEEE Service Center, str. 2357–2363, 2004.
- [15] T. Tušar, P. Korošec, G. Papa, B. Filipič in J. Šilc, “A comparative study of stochastic optimization methods in electric motor design,” *Applied Intelligence*, letn. 2, št. 27, str. 101–111, 2007.
- [16] J. Knowles, L. Thiele in E. Zitzler, “A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers,” Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland,” 214, feb. 2006, popravljena različica.
- [17] M. P. Hansen in A. Jaszkiewicz, “Evaluating the quality of approximations to the non-dominated set,” Technical University of Denmark, tehn. por. IMM-REP-1998-7, 1998.
- [18] Y. Y. Haimes, L. S. Lasdon in D. A. Wismer, “On a bicriterion formulation of the problems of integrated system identification in system optimization,” *IEEE Transactions on Systems, Man in Cybernetics*, letn. 1, št. 3, str. 296–297, 1971.

- [19] T. Takahama, S. Sakai in N. Iwane, “Solving Nonlinear Constrained Optimization Problems by the ϵ Constrained Differential Evolution,” *Systems, Man and Cybernetics, 2006. SMC '06. IEEE International Conference on*, letn. 3, str. 2322–2327, 8–11 okt. 2006.
- [20] D. B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence, Third Edition (IEEE Press Series on Computational Intelligence)*. Hoboken, New Jersey: Wiley-IEEE Press, 2006.
- [21] C. Darwin, *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*. John Murray, 1859.
- [22] A. Weismann, *The Germ-Plasm: A Theory of Heredity*. Charles Scribner's Sons, 1893.
- [23] G. Mendel, “Experiments in Plant Hybridization,” 1865.
- [24] E. Mezura-Montes, “Alternative Techniques to Handle Constraints in Evolutionary Optimization,” doktorska disertacija, Computer Science Section, Electrical Eng. Department, 2004.
- [25] J. D. Schaffer, “Multiple Objective Optimization with Vector Evaluated Genetic Algorithms,” v *Proceedings of the 1st International Conference on Genetic Algorithms*. Mahwah, NJ, USA: Lawrence Erlbaum Associates, Inc., str. 93–100, 1985.
- [26] C. A. Coello Coello, “20 Years of Evolutionary Multi-Objective Optimization: What Has Been Done and What Remains to be Done,” v *Computational Intelligence: Principles and Practice*, G. Y. Yen in D. B. Fogel, ured. Vancouver, Canada: IEEE Computational Intelligence Society, pogl. 4, str. 73–88, 2006.
- [27] ——, “Evolutionary multi-objective optimization: a historical view of the field,” *Computational Intelligence Magazine, IEEE*, letn. 1, št. 1, str. 28–36, feb. 2006.
- [28] K. Deb, S. Agrawal, A. Pratab in T. Meyarivan, “A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II,” v *Proceedings of the Parallel Problem Solving from Nature VI Conference*, M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo in H.-P. Schwefel, ured. Paris, France: Springer. Lecture Notes in Computer Science No. 1917, str. 849–858, 2000.
- [29] E. Zitzler, M. Laumanns in L. Thiele, “SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization,” v *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, K. C. Giannakoglou, D. T. Tsahalis, J. Périaux, K. D. Papailiou in T. Fogarty, ured. Athens, Greece: International Center for Numerical Methods in Engineering (CIMNE), str. 95–100, 2001.

- [30] H. A. Abbass, R. Sarker in C. Newton, "PDE: A Pareto-frontier Differential Evolution Approach for Multi-objective Optimization Problems," v *Proceedings of the Congress on Evolutionary Computation 2001 (CEC 2001)*, letn. 2. Piscataway, New Jersey: IEEE Service Center, str. 971–978, 2001.
- [31] E. Zitzler in S. Künzli, "Indicator-Based Selection in Multiobjective Search," v *Parallel Problem Solving from Nature (PPSN VIII)*, X. Yao in drugi, ured. Berlin, Germany: Springer-Verlag, str. 832–842, 2004.
- [32] V. L. Huang, P. N. Suganthan, A. K. Qin in S. Baskar, "Multiobjective Differential Evolution with External Archive and Harmonic Distance-Based Diversity Measure," Nanyang Technological University, Singapore, tehn. por. TR-07-01, 2006.
- [33] T. Robič in B. Filipič, "DEMO: Differential Evolution for Multiobjective Optimization," v *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization – EMO 2005*, ser. Lecture Notes in Computer Science, letn. 3410. Springer, str. 520–533, 2005.
- [34] T. Robič, "Performance of demo on new test problems: A comparison study," v *Proceedings of the Fourteenth International Electrotechnical and Computer Science Conference – ERK 2005*, letn. B, str. 121–124, 2005.
- [35] F. di Pierro, S.-T. Khu in D. A. Savić, "An Investigation on Preference Order-Ranking Scheme for Multiobjective Evolutionary Optimization," *IEEE Transactions on Evolutionary Computation*, letn. 11, št. 1, str. 17–45, 2007.
- [36] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, Massachusetts: Addison-Wesley Publishing Company, 1989.
- [37] J. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.
- [38] L. Fogel, A. Owens in M. Walsh, *Artificial Intelligence Through Simulated Evolution*. John Wiley & Sons Inc, 1966.
- [39] X. Yao, Y. Liu in G. Lin, "Evolutionary Programming Made Faster," *IEEE Transactions on Evolutionary Computation*, letn. 3, št. 2, str. 82–102, 1999.
- [40] K. Zielinski in R. Laur, "Adaptive Parameter Setting for a Multi-Objective Particle Swarm Optimization Algorithm," v *The 2007 IEEE Congress on Evolutionary Computation CEC 2007*. IEEE Press, str. 3019–3026, 2007.
- [41] K. Zielinski, D. Peters in R. Laur, "Run Time Analysis regarding Stopping Criteria for Differential Evolution and Particle Swarm Optimization," v *Proceedings of the 1st International Conference on Experiments/Process/System Modelling/Simulation/Optimization, Athens, Greece*. IC-EpsMsO, 2005.

- [42] P. Korošec, J. Šilc, K. Oblak in F. Kosel, "The Differential Ant-Stigmergy Algorithm: An Experimental Evaluation and a Real-World Application," v *The 2007 IEEE Congress on Evolutionary Computation CEC 2007*. IEEE Press, str. 157–164, 2007.
- [43] H.-G. Beyer in H.-P. Schwefel, "Evolution strategies — A comprehensive introduction," *Natural Computing*, letn. 1, str. 3–52, 2002.
- [44] T. Bäck, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. New York: Oxford University Press, 1996.
- [45] H.-P. Schwefel, *Evolution and Optimum Seeking*, ser. Sixth-Generation Computer Technology. New York: Wiley Interscience, 1995.
- [46] N. Hansen in A. Ostermeier, "Completely Derandomized Self-Adaptation in Evolution Strategies," *Evolutionary Computation*, letn. 9, št. 2, str. 159–195, 2001.
- [47] C. Igel, N. Hansen in S. Roth, "Covariance Matrix Adaptation for Multi-objective Optimization," *Evolutionary Computation*, letn. 15, št. 1, str. 1–28, 2007.
- [48] J. Brest, S. Greiner, B. Bošković, M. Mernik in V. Žumer, "Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems," *IEEE Transactions on Evolutionary Computation*, letn. 10, št. 6, str. 646–657, 2006.
- [49] J. Brest, V. Žumer in M. S. Maučec, "Self-adaptive Differential Evolution Algorithm in Constrained Real-Parameter Optimization," v *The 2006 IEEE Congress on Evolutionary Computation CEC 2006*. IEEE Press, str. 919–926, 2006.
- [50] J. Brest, B. Bošković, S. Greiner, V. Žumer in M. S. Maučec, "Performance comparison of self-adaptive and adaptive differential evolution algorithms," *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, letn. 11, št. 7, str. 617–629, 2007.
- [51] J. Brest, A. Zamuda, B. Bošković, M. S. Maučec in V. Žumer, "High-dimensional Real-parameter Optimization Using Self-adaptive Differential Evolution Algorithm with Population Size Reduction," v *2008 IEEE World Congress on Computational Intelligence*. IEEE Press, str. 2032–2039, 2008.
- [52] E. Mezura-Montes in B. C. Lopez-Ramirez, "Comparing bio-inspired algorithms in constrained optimization problems," *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, str. 662–669, 25–28 sept. 2007.
- [53] M. M. Ali in A. Törn, "Population set-based global optimization algorithms: some modifications and numerical studies," *Comput. Oper. Res.*, letn. 31, št. 10, str. 1703–1725, 2004.

- [54] A. W. Iorio in X. Li, "Incorporating Directional Information within a Differential Evolution Algorithm for Multi-objective Optimization," v *2006 Genetic and Evolutionary Computation Conference (GECCO 2006)*, M. K. et al., Ur., letn. 1. Seattle, Washington, USA: ACM Press. ISBN 1-59593-186-4, str. 691–697, 2006.
- [55] J. Lampinen, "A Bibliography of Differential Evolution Algorithm," Lappeenranta University of Technology, Department of Information Technology, Laboratory of Information Processing, tehn. por., 2001, online <http://www2.lut.fi/~jlampine/debiblio.htm>.
- [56] H.-Y. Fan in J. Lampinen, "A Trigonometric Mutation Operation to Differential Evolution," *Journal of Global Optimization*, letn. 27, št. 1, str. 105–129, 2003.
- [57] F. Xue, A. C. Sanderson in R. J. Graves, "Pareto-based Multi-Objective Differential Evolution," v *Proceedings of the 2003 Congress on Evolutionary Computation (CEC 2003)*, letn. 2. Canberra, Australia: IEEE Press, str. 862–869, 2003.
- [58] A. G. Hernández-Díaz, L. V. Santana-Quintero, C. C. Coello, R. Caballero in J. Molina, "A new proposal for multi-objective optimization using differential evolution and rough sets theory," v *Proceedings of the 8th annual conference on Genetic and evolutionary computation — GECCO 2006*, letn. 1, str. 675–682, 2006.
- [59] M. Ali, "Differential evolution with preferential crossover," *European Journal of Operational Research*, letn. 127, št. 3, str. 1137–1147, sept. 2007, online <http://ideas.repec.org/a/eee/ejores/v181y2007i3p1137-1147.html>.
- [60] J. Zhang, Sanderson in A. C., "JADE: Self-adaptive differential evolution with fast and reliable convergence performance," *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, str. 2251–2258, 25–28 sept. 2007.
- [61] Z. Yang, K. Tang in X. Yao, "Differential Evolution for High-Dimensional Function Optimization," v *Proceedings of the 2007 IEEE Congress on Evolutionary Computation CEC 2007*, Singapore, str. 3523–3530, 25–28 sept. 2007.
- [62] A. Zamuda, J. Brest, B. Bošković in V. Žumer, "Large Scale Global Optimization Using Differential Evolution with Self Adaptation and Cooperative Co-evolution," v *2008 IEEE World Congress on Computational Intelligence*.IEEE Press, str. 3719–3726, 2008.
- [63] Z. Yang, J. He in X. Yao, "Making a Difference to Differential Evolution," v *Advances in Metaheuristics for Hard Optimization*, ser. Lecture Notes in Computer Science, Z. Michalewicz in P. Siarry, ured. Berlin: Springer, str. 397–414, 2008.
- [64] B. Bošković, S. Greiner, J. Brest in V. Žumer, "A Differential Evolution for the Tuning of a Chess Evaluation Function," v *The 2006 IEEE Congress on Evolutionary Computation CEC 2006*.IEEE Press, str. 6742–6747, 2006.

- [65] K. V. Price, R. M. Storn in J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*, ser. Natural Computing Series, G. Rozenberg, T. Bäck, A. E. Eiben, J. N. Kok in H. P. Spalink, ured. Berlin, Germany: Springer-Verlag, 2005.
- [66] V. Feoktistov, *Differential Evolution: In Search of Solutions (Springer Optimization and Its Applications)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.
- [67] B. Babu in M. M. L. Jehan, “Differential Evolution for Multi-Objective Optimization,” v *Proceedings of the 2003 Congress on Evolutionary Computation (CEC 2003)*, letn. 4. Canberra, Australia: IEEE Press, str. 2696–2703, 2003.
- [68] B. V. Babu, J. H. S. Mubeen in P. G. Chakole, “Multiobjective Optimization Using Differential Evolution,” *TechGenesis – The Journal of Information Technology*, letn. 2, št. 2, str. 4–12, 2005.
- [69] N. Hansen, S. Muller in P. Koumoutsakos, “Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES),” *Evolutionary Computation*, letn. 11, št. 1, str. 1–18, 2003.
- [70] E. Limpert, W. Stahel in M. Abbt, “Log-normal Distributions across the Sciences: Keys and Clues,” *BioScience*, letn. 51, št. 5, str. 341–352, 2001.
- [71] A. E. Eiben in J. E. Smith, *Introduction to Evolutionary Computing (Natural Computing Series)*. Springer, 2003.
- [72] K. Liang, X. Yao in C. Newton, “Adapting Self-adaptive Parameters in Evolutionary Algorithms,” str. 171–180, 2001.
- [73] J. Liu in J. Lampinen, “A Fuzzy Adaptive Differential Evolution Algorithm,” *Soft Comput.*, letn. 9, št. 6, str. 448–462, 2005.
- [74] M. G. Omran, A. Salman in A. P. Engelbrecht, “Self-adaptive Differential Evolution,” v *CIS (1)*, str. 192–199, 2005.
- [75] A. Qin in P. Suganthan, “Self-adaptive Differential Evolution Algorithm for Numerical Optimization,” v *Proceedings of the 2005 Congress on Evolutionary Computation (CEC 2005)*, letn. 2. IEEE Press, str. 1785– 1791, 2005.
- [76] J. Teo, “Exploring dynamic self-adaptive populations in differential evolution,” *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, letn. 10, št. 8, str. 673–686, 2006.
- [77] J. Brest, V. Žumer in M. S. Maučec, “Population size in differential evolution algorithm,” *Elektrotehniški vestnik*, letn. 74, št. 1-2, str. 55–60, 2007.

- [78] J. Brest, "Differential Evolution with Self-Adaptation," *Encyclopedia of Artificial Intelligence*, ISBN 978-1-59904-849-9, 2008, v tisku.
- [79] K. Deb, S. Agrawal, A. Pratab in T. Meyarivan, "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II," Indian Institute of Technology, Kanpur, India, KanGAL report 200001, 2000.
- [80] N. Srinivas in K. Deb, "Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms," *Evolutionary Computation*, letn. 2, št. 3, str. 221–248, 1994.
- [81] E. Zitzler, M. Laumanns in L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, tehn. por. 103, 2001.
- [82] M. Smid, "Closest-Point Problems in Computational Geometry," *Handbook on Computational Geometry*, edited by J.-R. Sack, North Holland, Amsterdam., str. 877–935, 2000.
- [83] C. K. Goh in K. C. Tan, "An Investigation on Noisy Environments in Evolutionary Multiobjective Optimization," *IEEE Transactions on Evolutionary Computation*, letn. 11, št. 3, str. 354–381, 2007.
- [84] M. Laumanns, L. Thiele in E. Zitzler, "An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method," *European Journal of Operational Research*, letn. 169, št. 3, str. 932–942, mar. 2006.
- [85] K. Tan, Y. Yang in C. Goh, "A distributed Cooperative coevolutionary algorithm for multiobjective optimization," *Evolutionary Computation, IEEE Transactions on*, letn. 10, št. 5, str. 527–549, okt. 2006.
- [86] S. Y. Chong, P. Tino in X. Yao, "Measuring generalization performance in co-evolutionary learning," *IEEE Transactions on Evolutionary Computation*, avg. 2007.
- [87] B. Alatas, E. Akin in A. Karci, "MODENAR: Multi-objective differential evolution algorithm for mining numeric association rules," *Applied Soft Computing*, letn. 8, št. 1, str. 646–656, 2008.
- [88] A. Zamuda, J. Brest, B. Bošković in V. Žumer, "Differential Evolution for Multi-objective Optimization with Self Adaptation," v *The 2007 IEEE Congress on Evolutionary Computation CEC 2007*. IEEE Press, str. 3617–3624, 2007.
- [89] T. Tušar, "Design of an Algorithm for Multiobjective Optimization with Differential Evolution," M.Sc. Thesis, Faculty of Computer and Information Science, University of Ljubljana, 2007.

- [90] R. Storn in K. Price, "Differential Evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces," Berkeley, CA, tehn. por. TR-95-012, 1995.
- [91] T. Tušar in B. Filipič, "Differential Evolution versus Genetic Algorithms in Multi-objective Optimization," v *Proceedings of the Fourth International Conference on Evolutionary Multi-Criterion Optimization – EMO 2007*, ser. Lecture Notes in Computer Science, letn. 4403. Springer, str. 257–271, 2007.
- [92] M. Laumanns, L. Thiele, E. Zitzler, E. Welzl in K. Deb, "Running Time Analysis of Multi-objective Evolutionary Algorithms on a Simple Discrete Optimization Problem," v *PPSN VII: Proceedings of the 7th International Conference on Parallel Problem Solving from Nature*. London, UK: Springer-Verlag, str. 44–53, 2002.
- [93] D. H. Wolpert in W. G. Macready, "No Free Lunch Theorems for Search," Santa Fe Institute, Santa Fe, NM, tehn. por. SFI-TR-95-02-010, 1995.
- [94] ——, "No Free Lunch Theorems for Optimization," *IEEE Transactions on Evolutionary Computation*, letn. 1, št. 1, str. 67–82, April 1997.
- [95] S. Huband, P. Hingston, L. Barone in L. While, "A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit," *IEEE Transactions on Evolutionary Computation*, letn. 10, št. 5, str. 477–506, okt. 2006.
- [96] E. Zitzler in L. Thiele, "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach," *IEEE Transactions on Evolutionary Computation*, letn. 3, št. 4, str. 257–271, 1999.
- [97] L. While, L. Bradstreet, L. Barone in P. Hingston, "Heuristics for Optimising the Calculation of Hypervolume for Multi-objective Optimisation Problems," v *Proceedings of the 2005 Congress on Evolutionary Computation (CEC 2005)*, letn. 3. Edinburgh, UK: IEEE Press, str. 2225–2232, 2005.
- [98] L. While, P. Hingston, L. Barone in S. Huband, "A Faster Algorithm for Calculating Hypervolume," *IEEE Transactions on Evolutionary Computation*, letn. 10, št. 1, str. 29–38, feb. 2006.
- [99] L. Bradstreet, L. While in L. Barone, "A Fast Incremental Hypervolume Algorithm," *IEEE Transactions on Evolutionary Computation*, 2008, DOI: 10.1109/TEVC.2008.919001.
- [100] M. Hansen in A. Jaskiewicz, "Evaluating the Quality of Approximations to the Non-dominated Set," Institut of Mathematical Modelling, Technical University of Denmark, tehn. por. IMM-REP-1998-7, 1998.

- [101] S. Bleuler, M. Laumanns, L. Thiele in E. Zitzler, “PISA — A Platform and Programming Language Independent Interface for Search Algorithms,” v *Evolutionary Multi-Criterion Optimization (EMO 2003)*, ser. Lecture Notes in Computer Science, C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb in L. Thiele, ured. Berlin: Springer, str. 494–508, 2003.
- [102] J. Knowles, “A summary-attainment-surface plotting method for visualizing the performance of stochastic multiobjective optimizers,” v *ISDA '05: Proceedings of the 5th International Conference on Intelligent Systems Design and Applications*. Washington, DC, USA: IEEE Computer Society, str. 552–557, 2005.
- [103] A. Zamuda, “Testne funkcije in metrike za večkriterijsko optimizacijo,” v *Zbirka povzetkov iz delavnic Algoritmi po vzoru iz narave, online <http://labraj.uni-mb.si/avn>*, 2007, online: <http://labraj.uni-mb.si/avn>
- [104] V. L. Huang, A. K. Qin, P. N. Suganthan in M. F. Tasgetiren, “Multi-objective Optimization based on Self-adaptive Differential Evolution,” v *The 2007 IEEE Congress on Evolutionary Computation CEC 2007*. IEEE Press, str. 3601–3608, 2007.
- [105] V. L. Huang, A. K. Qin in P. N. Suganthan, “Self-adaptive Differential Evolution Algorithm for Constrained Real-Parameter Optimization,” v *2006 Congress on Evolutionary Computation (CEC 2006)*. IEEE Service Center, str. 17–24, 2006.
- [106] S. Kukkonen in J. Lampinen, “Performance Assessment of Generalized Differential Evolution 3 (GDE3) with a Given Set of Problems,” v *The 2007 IEEE Congress on Evolutionary Computation CEC 2007*. IEEE Press, str. 3593–3600, 2007.
- [107] S. Kukkonen in K. Deb, “A Fast and Effective Method for Pruning of Non-dominated Solutions in Many-Objective Problems,” v *Proceedings of the 9th International Conference on Parallel Problem Solving from Nature (PPSN IX)*. Springer, str. 553–562, 2006.
- [108] D. Sharma, A. Kumar, K. Deb in K. Sindhya, “Hybridization of SBX Based NSGA-II and Sequential Quadratic Programming for Solving Multi-objective Optimization Problems,” v *The 2007 IEEE Congress on Evolutionary Computation CEC 2007*. IEEE Press, str. 3003–3010, 2007.
- [109] A. Kumar, D. Sharma in K. Deb, “A Hybrid Multi-Objective Optimization Procedure Using PCX Based NSGA-II and Sequential Quadratic Programming,” v *The 2007 IEEE Congress on Evolutionary Computation CEC 2007*. IEEE Press, str. 3011–3018, 2007.
- [110] L.-Y. Tseng in C. Chen, “Multiple Trajectory Search for Multiobjective Optimization,” v *The 2007 IEEE Congress on Evolutionary Computation CEC 2007*. IEEE Press, str. 3609–3616, 2007.

- [111] K. Zielinski in R. Laur, "Differential Evolution with Adaptive Parameter Setting for Multi-Objective Optimization," v *The 2007 IEEE Congress on Evolutionary Computation CEC 2007*. IEEE Press, str. 3585–3592, 2007.
- [112] K. Zielinski, P. Weitkemper, R. Laur in K.-D. Kammeyer, "Optimization of Power Allocation for Interference Cancellation With Particle Swarm Optimization," *IEEE Transactions on Evolutionary Computation*, 2008, DOI: 10.1109/TEVC.2008.920672.
- [113] T. Stewart, R. Janssen in M. van Herwijnen, "A genetic algorithm approach to multiobjective land use planning," *Computers and Operations Research*, letn. 31, št. 14, str. 2293–2313, 2004.
- [114] K. Matthews, K. Buchan, A. Sibbald in S. Craw, "Using soft-systems methods to evaluate the outputs from multi-objective land use planning tools," v *Integrated Assessment and Decision Support: Proc. Int. Environmental Modelling and Software Society*, str. 24–27, 2002.
- [115] G. Huang, S. Cohen, Y. Yin in B. Bass, "Land resources adaptation planning under changing climate-a study for the Mackenzie Basin," 1998.
- [116] J. Bladt, "Multiobjective Land Use Optimisation using Evolutionary Algorithms," M.Sc. Thesis, Department of Systematic Botany, Institute of Biology, University of Aarhus, 2002.
- [117] P. Boggs in J. Tolle, "Sequential quadratic programming for large-scale nonlinear optimization," *Journal of Computational and Applied Mathematics*, letn. 124, št. 1-2, str. 123–137, 2000.
- [118] C. T. Lawrence, J. L. Zhou in A. L. Tits, "User's Guide for CFSQP Version 2.5: A C Code for Solving (Large Scale) Constrained Nonlinear (Minimax) Optimization Problems, Generating Iterates Satisfying All Inequality Constraints," Institute for Systems Research, University of Maryland, College Park, MD 20742, tehn. por. TR-94-16r1, 1997.
- [119] T. Okabe, Y. Jin, M. Olhofer in B. Sendhoff, "On Test Functions for Evolutionary Multi-objective Optimization," v *PPSN*, str. 792–802, 2004.
- [120] G. Rudolph, B. Naujoks in M. Preuss, "Capabilities of EMOA to Detect and Preserve Equivalent Pareto Subsets," v *Evolutionary Multi-Criterion Optimization (EMO 2007)*, letn. 4403. Springer, str. 36–50, 2007.
- [121] E. Zitzler, K. Deb in L. Thiele, "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," *Evolutionary Computation*, letn. 8, št. 2, str. 173–195, 2000.

- [122] K. Deb, L. Thiele, M. Laumanns in E. Zitzler, "Scalable Multi-Objective Optimization Test Problems," v *Congress on Evolutionary Computation (CEC 2002)*, letn. 1. Piscataway, New Jersey: IEEE Service Center, str. 825–830, 2002.
- [123] ——, "Scalable Test Problems for Evolutionary Multi-Objective Optimization," Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, tehn. por. 112, 2001.
- [124] D. A. V. Veldhuizen, "Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations," doktorska disertacija, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio, 1999.

Življenjepis

Ime in priimek: Aleš Zamuda

Rojen: 23. 4. 1982 na Ptuju

Šolanje: 1989 – 1997 Osnovna šola Velika Nedelja

1997 – 2001 Gimnazija Ptuj

2001 – danes Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru

Zaposlitev: 2006 – danes Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru

Članstvo v organizacijah: 2006 – danes IEEE

Bibliografija

ČLANKI IN DRUGI SESTAVNI DELI

1.03 Kratki znanstveni prispevek

1. BOŠKOVIĆ, Borko, ZAMUDA, Aleš, BREST, Janez, GREINER, Sašo, ŽUMER, Viljem. An Opposition-based differential evolution with adaptive mechanism, applied to the tuning of a chess evaluation function. *Journal of computational intelligence*, 2008, vol. 1, no. 1, str. 1-6. [COBISS.SI-ID 12127766]
2. BREST, Janez, ZAMUDA, Aleš, BOŠKOVIĆ, Borko, GREINER, Sašo, ŽUMER, Viljem. An analysis of the control parameters adaptation in the differential evolution algorithm. *Journal of computational intelligence*, 2008, vol. 1, no. 1, str. 7-22. [COBISS.SI-ID 12128022]

1.08 Objavljeni znanstveni prispevek na konferenci

3. ZAMUDA, Aleš, STRNAD, Damjan. Interaktivni modelirnik realističnih animiranih dreves. V: ZAJC, Baldomir (ur.). *Zbornik trinajste mednarodne elektrotehniške in računalniške konference ERK 2004, 27. - 29. september 2004, Portorož, Slovenija*. Ljubljana: IEEE Region 8, Slovenska sekcijsa IEEE, [2004], zv. B, str. 11-14. [COBISS.SI-ID 9058326]
4. ZAMUDA, Aleš, BREST, Janez, GUID, Nikola, ŽUMER, Viljem. Construction of virtual trees within ecosystems with ecomod tool. V: International Conference on Advances in the Internet, Processing, Systems and Interdisciplinary Research, Bled, November 30-December 3, 2006. *Proceedings of the VIPSI-2006 Slovenia : Multidisciplinary, Interdisciplinary, Transdisciplinary: M+I+T++, Bled Slovenia, November 30-December 3, 2006*. Belgrade: IPSI, 2006, [5] f. [COBISS.SI-ID 11005206]
5. ZAMUDA, Aleš, GUID, Nikola. Modeliranje, simulacija in upodabljanje gozdov. V: ZAJC, Baldomir (ur.), TROST, Andrej (ur.). *Zbornik petnajste mednarodne Elektrotehniške in računalniške konference ERK 2006, 25. - 27. september 2006, Portorož, Slovenija*, (Zbornik ... Elektrotehniške in računalniške konference ERK ...).

Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, 2006, zv. B, str. 391-392.
[COBISS.SI-ID 10776598]

6. ZAMUDA, Aleš, BREST, Janez, BOŠKOVIĆ, Borko, ŽUMER, Viljem. Differential evolution for multiobjective optimization with self adaptation. V: *CEC 2007 : 2007 Congress on Evolutionary Computation : 25-28 September 2007, Singapore*. Piscataway: Institute of Electrical and Electronics Engineers, 2007, str. 3617-3624. [COBISS.SI-ID 11730710]

7. ZAMUDA, Aleš, BREST, Janez, GVID, Nikola, ŽUMER, Viljem. Modelling, simulation and visualisation of forest ecosystems. V: *Region 8 Eurocon 2007 : proceedings*. Piscataway, NJ: IEEE Service Center, [2007], str. 2600-2606.
[COBISS.SI-ID 11643926]

8. BOŠKOVIĆ, Borko, BREST, Janez, ZAMUDA, Aleš, ŽUMER, Viljem. Uglaševanje šahovskega programa BBChess z uporabo algoritma diferencialne evolucije. V: ZAJC, Baldomir (ur.), TROST, Andrej (ur.). *Zbornik šestnajste mednarodne Elektrotehniške in računalniške konference ERK 2007, 24. - 26. september 2007, Portorož, Slovenija*, (Zbornik ... Elektrotehniške in računalniške konference ERK ...). Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, 2007, zv. B, str. 73-76. [COBISS.SI-ID 11760662]

9. BREST, Janez, ZAMUDA, Aleš, BOŠKOVIĆ, Borko, ŽUMER, Viljem. Večkriterijska optimizacija: primerjava algoritmov v MOjDE in DEMO. V: ZAJC, Baldomir (ur.), TROST, Andrej (ur.). *Zbornik šestnajste mednarodne Elektrotehniške in računalniške konference ERK 2007, 24. - 26. september 2007, Portorož, Slovenija*, (Zbornik ... Elektrotehniške in računalniške konference ERK ...). Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, 2007, zv. B, str. 85-88. [COBISS.SI-ID 11773718]

10. ZAMUDA, Aleš, BREST, Janez, BOŠKOVIĆ, Borko, ŽUMER, Viljem. Večkriterijska optimizacija: eksperimentalni rezultati algoritmov MOjDE in DEMO. V: ZAJC, Baldomir (ur.), TROST, Andrej (ur.). *Zbornik šestnajste mednarodne Elektrotehniške in računalniške konference ERK 2007, 24. - 26. september 2007, Portorož, Slovenija*, (Zbornik ... Elektrotehniške in računalniške konference ERK ...). Ljubljana: IEEE Region 8, Slovenska sekcija IEEE, 2007, zv. B, str. 89-92.
[COBISS.SI-ID 11773974]

11. BREST, Janez, ZAMUDA, Aleš, BOŠKOVIĆ, Borko, SEPESY MAUČEC, Mirjam, ŽUMER, Viljem. High-dimensional real-parameter optimization using self-adaptive differential evolution algorithm with population size reduction. V: *IEEE World Congress on Computational Intelligence, Hong Kong, June 1-6, 2008. WCCI 2008 Proceedings*. [S. l.]: IEEE, 2008, str. 2032-2039. [COBISS.SI-ID 12328982]

- 12.** ZAMUDA, Aleš, BREST, Janez, BOŠKOVIĆ, Borko, ŽUMER, Viljem. Large scale global optimization using differential evolution with self-adaptation and cooperative co-evolution. V: IEEE World Congress on Computational Intelligence, Hong Kong, June 1-6, 2008. *WCCI 2008 Proceedings*. [S. l.]: IEEE, 2008, str. 3719-3726. [COBISS.SI-ID 12329238]

MONOGRAFIJE IN DRUGA ZAKLJUČENA DELA

2.11 Diplomsko delo

- 13.** ZAMUDA, Aleš. *Modeliranje, simulacija in upodabljanje drevesnih ekosistemov: diplomska naloga univerzitetnega študijskega programa*, (Fakulteta za elektrotehniko, računalništvo in informatiko, Diplomska dela univerzitetnega študja). Maribor: [A. Zamuda], 2006. 98 f., ilustr. [COBISS.SI-ID 10822166]

2.21 Programska oprema

- 14.** ZAMUDA, Aleš, STRNAD, Damjan, BREST, Janez, BOŠKOVIĆ, Borko. *Drevo*. [s. l.: s. n., 2007]. 1 CD ROM. Platforma: Win32 (Windows in Linux, BSD, Mac). <http://labraj.uni-mb.si/~ales/codes/drevo.zip>. [COBISS.SI-ID 11607062]

- 15.** BOŠKOVIĆ, Borko, BREST, Janez, ZAMUDA, Aleš. *Šahovski program BBChess 1.1*. Maribor: [s. n., 2007]. 1 CD ROM. Sistemske zahteve: Windows/Linux. http://labraj.uni-mb.si/images/6/6e/BBChess1.1_src.tar.bz. [COBISS.SI-ID 11873046]

- 16.** ZAMUDA, Aleš, BREST, Janez. *FastFractal "DoubleDip" C++2Java Interface*. Maribor: [s. n., 2008]. 1 CD ROM. Javno dostopna prosta odprto kodna programska oprema (GPL v2), platformno neodvisno, dostopno v Angliji, na Kitajskem in v Sloveniji. <http://www.cs.bham.ac.uk/research/projects/ecb/data/150/cec08-f7-cpp-with-sources-1.0RC1.zip>, <http://labraj.uni-mb.si/~ales/codes/cec08-f7-cpp-with-sources-1.0RC1.zip>, <http://nical.ustc.edu.cn/conferences/cec2008/lsgo/LSGO.CEC08.Benchmark.zip>. [COBISS.SI-ID 12207894]

SEKUNDARNO AVTORSTVO

Pisec recenzij

- 17.** *IEEE Transactions on Evolutionary Computation*. Zamuda, Aleš (pisec recenzij 2007). [Print ed.]. New York: Institute of Electrical and Electronics Engineers, 1997-. ISSN 1089-778X. [COBISS.SI-ID 16940805]



UNIVERZA V MARIBORU

IZJAVA O ISTOVETNOSTI TISKANE IN ELEKTRONSKE VERZIJE DIPLOMSKE NALOGE

Ime in priimek študenta:

Aleš Zamuda

Številka indeksa:

95031458

Študijski program:

računalništvo in informatika

Naslov diplomskega dela:

Samoprilagajanje krmilnih parametrov pri algoritmu diferencialne evolucije za večkriterijsko optimizacijo

Mentor:

izr. prof. dr. Janez Brest

Somentor:

Podpisani Aleš Zamuda se strinjam, da se moja diplomska naloga objavi na portalu Digitalne knjižnice Univerze v Mariboru.

Izjavljam, da sem diplomsko nalogo izdelal sam ob pomoči mentorja in nisem kršil avtorskih pravic in intelektualne lastnine drugih. Tiskana verzija diplomske naloge je istovetna elektronski verziji, ki sem jo oddal v Digitalno knjižnico Univerze v Mariboru.

Podpis študenta:

Aleš Zamuda

V nadaljevanju izpolnite samo, če diplomska naloga ne sme biti javno dostopna.

Diplomsko delo ni javno dostopno, zaradi zagotavljanja konkurenčne prednosti varstva industrijske lastnine ali tajnosti podatkov naročnika.

Naročnik izdelave začasno nedostopne diplomske naloge je:

Naloga ne sme biti javno dostopna do _____.

Podpis mentorja:

Podpis odgovorne osebe naročnika in žig:
