

UNIVERZA V MARIBORU

FAKULTETA ZA ELEKTROTEHNIKO, RAČUNALNIŠTVO IN
INFORMATIKO

Aleš Zamuda

Operacijske raziskave logističnih, transportnih in
ekonomskih sistemov: zbrano gradivo



Univerza v Mariboru

MARIBOR, 2020

CIP — kataložni zapis o publikaciji
Univerzitetna knjižnica Maribor

Aleš Zamuda
Operacijske raziskave logističnih, transportnih in ekonomskih sistemov: zbrano gradivo / Aleš Zamuda, Maribor, Univerza v Mariboru

Naslov:	Operacijske raziskave logističnih, transportnih in ekonomskih sistemov: zbrano gradivo
Avtorja:	Aleš Zamuda
Strokovni recenzenti:	red. prof. dr. Marjan Mernik
Jezikovni recenzenti:	/
Tehnični recenzenti:	/
Računalniški prelom:	Aleš Zamuda
Oblikovanje slik:	Aleš Zamuda
Oblikovanje ovitka:	Aleš Zamuda
Tipologija/vrsta publikacije:	2.05 Drugo učno gradivo
Založnik:	Univerza v Mariboru
Kraj založbe:	Maribor
Datum izida:	februar 2020
Različica (e-pub):	R 1
URL (e-pub):	https://dk.um.si/Iskanje.php?type=napredno&niz0=operacijske&stl0=Naslov&op1=AND&niz1=Zamuda&stl1=Avtor&op2=AND&niz2=&stl2=Opis&op3=AND&niz3=2020&stl3=LetoIzida&vrsta=0&jezik=0&vir=3
Sistemske zahteve (e-pub):	računalnik združljiv z IBM modeli PC, operacijski sistem Linux, Mac ali Windows, 8 GB RAM; internetni dostop
Programske zahteve (e-pub):	internetni brskalnik, program Okular ali Adobe Reader, Bash, VirtualBox

Aktivno vsebinsko kazalo

Predgovor	1
Uvod — splošno o predmetu, vajah in nalogah	2
Vsebinski pregled	3
1 Uvod: osnovni pojmi in primeri operacijskih raziskav logističnih, transportnih in ekonomskih sistemov	6
Operacijske raziskave	6
Matematično programiranje in evolucijsko računanje	14
2 Arhitektura algoritmov in taksonomija algoritmov za operacijske raziskave logističnih, transportnih in ekonomskih sistemov	16
Algoritem diferencialne evolucije	18
3 Načrtovanje algoritmov operacijskih raziskav logističnih, transportnih in ekonomskih sistemov	30
Primer algoritma za operacijske raziskave: L-SHADE	31
Primer algoritma za operacijske raziskave: DISH	33
Primer razširitve algoritma za operacijske raziskave: DISHv	36
Združen pristop z optimizacijo za ORLTES	39
4 Kompleksnost v operacijskih raziskavah logističnih, transportnih in ekonomskih sistemov	42
Diferencialna evolucija za rekonstrukcijo parametriziranih proceduralnih drevesnih modelov	44
Kompleksnost v kibernetiko-tehnični sistemih postopkov	49

5 Planiranje kompleksnih misij in scenarijev v nepredvidljivih okoljih	52
Robotska morska sonda Slocum G2	53
Vzgonski pogon in uporabnost podmorskih sond	53
Satelitska navigacija in avtonomnost	53
Realno-časovni podatkovni tokovi o okolju in naloga optimalne morske trajektorije	55
6 Obravnava omejitev pri operacijskih raziskavah logističnih, transportnih in ekonomskih sistemov: primeri iz globokomorske navigacije in ekonomskega razporejanja s scenariji za ovrednotenje kakovosti	56
Morske trajektorije za globinsko pokrivanje vrtincev na srednjeročnem prostorskem območju z omejitvami oblike koridorja	56
7 Robustnost na napake med misijami in zanesljivost transporta	61
8 Energetska avtonomnost in algoritmi za ekonomsko planiranje v sistemih sistemov	63
9 Primerjava kakovosti algoritmov za operacijske raziskave logističnih, transportnih in ekonomskih sistemov	67
Primerjava v okolju UGPP	69
Primerjava v okolju ekonomskih sistemov	73
10 Samostojno delo – primeri za preverjanje	74
11 Zaključek	78
Literatura	79

Aktivno pojmovno (indeksno) kazalo

- ABC** optimizacija s kolonijami čebel (angl. *Artificial Bee Colony*). 17
- ACO** optimizacija s kolonijo mravelj (angl. *Ant Colony Optimization*). 17
- AI** umetna inteligenca (angl. *Artificial Intelligence*). 6
- AIS** umetni imunski sistem (angl. *Artificial Immune System*). 17
- ASOC** revija ASOC (angl. *Applied Soft Computing*). 28, 78
- AUV** avtonomna podvodna robotska sonda (angl. *Autonomous Underwater Vehicle*). xii, 53, 55, 58
- AWC** umetne travne kolonije (angl. *Artificial Weed Colonies*). 17
- BA** optimizacija s čebelami (angl. *Bees Algorithm*). 17
- BB-DOB** delavnica za ovrednotenje diskretne optimizacije s črno škatlo (angl. *Black Box Discrete Optimization Benchmarking*). 74
- BEADO** ovrednotenje evucijskih algoritmov za diskretno optimizacijo (angl. *Benchmarking of Evolutionary Algorithms for Discrete Optimization*). 74, 78
- BFO** bakterijsko preiskovanje (angl. *Bacterial Foraging Optimization*). 17
- CA** akcija **COST** (angl. *COST Action*). 71
- CEC** konferenca CEC (angl. *IEEE Congress on Evolutionary Computation*). 27, 28, 33, 36, 69, 74, 76, 78
- cHiPSet** Visoko zmogljivo modeliranje in simulacija za velepodatkovne aplikacije (angl. *High-Performance Modelling and Simulation for Big Data Applications*). 71
- CI** računska inteligenca (angl. *Computational Intelligence*). 6, 65

- CLPSO** Izboljšana metoda za celostno optimizacijo z rojem delcev (angl. *An Improved Method for Comprehensive Learning Particle Swarm Optimization*). 70, 73
- CMA-ES** evlucijska strategija s prilagoditvijo kovariančne matrike (angl. *Covariance Matrix Adaptation Evolution Strategy*). 70, 73
- CoDE** diferencialna evolucija s kompozitnimi strategijami generiranja poskusnih vektorjev in krmilnih parametrov (angl. *Differential evolution with composite trial vector generation strategies and control parameters*). 70, 71, 73
- COST** Evropsko sodelovanje v znanosti in tehnologiji (angl. *European Cooperation in Science and Technology*). iii, 71
- CS** kukavičje iskanje (angl. *Cuckoo Search*). 17
- CYB** revija [IEEE CYB](#) (angl. *IEEE Transactions on Cybernetics*). 28
- DE** diferencialna evolucija (angl. *Differential Evolution*). x, xii, 17–23, 25, 26, 28–33, 36, 39, 40, 51, 55, 58–60, 65, 66, 69, 72
- DECMOSA-SQP** diferencialna evolucija s samoprilagajanjem in lokalnim iskanjem za večkriterijsko optimizacijo z omejitvami (angl. *Differential evolution with self-adaptation and local search for constrained multi-objective optimization*). 28
- DEMO** diferencialna evolucija za večkriterijsko optimizacijo (angl. *Differential Evolution for Multi-objective Optimization*). 28
- DEMOwSA** diferencialna evolucija za večkriterijsko optimizacijo s samoprilagajanjem (angl. *Differential Evolution for Multi-objective Optimization with Self-adaptation*). 28
- DISH** na razdalji temelječe prilagajanje krmilnih parametrov za diferencialno evolucijo, temelječo na zgodovinskem uspehu (angl. **DI**stance Based Parameter Adaptation for **S**ucces-**H**istory based *Differential Evolution*). xii, 33, 36, 74, 76
- EA** evlucijski algoritem (angl. *Evolutionary Algorithm*). ix, 16, 58, 69, 71, 74
- EC** evlucijskega računanje (angl. *Evolutionary Computation*). 6

- ECJ** revija ECJ (angl. *Evolutionary Computation*). 28, 78
- EP** evoliucijsko programiranje (angl. *Evolutionary Programming*). 16
- EPSDE** algoritem diferencialne evolucije z ansamblom parametrov in mutacijskih in križnih strategij (angl. *Differential Evolution Algorithm with Ensemble of Parameters and Mutation and Crossover Strategies*). 70, 71, 73
- ES** evoliucijska strategija (angl. *Evolution Strategy*). 17
- ESWA** revija ESWA (angl. *Expert Systems With Applications*). 39, 78
- FA** algoritem kresničke (angl. *Firefly Algorithm*). 18
- GA** genetski algoritem (angl. *Genetic Algorithm*). 16
- GDE** posplošena diferencialna evolucija (angl. *Generalized Differential Evolution*). 28
- GECCO** konferenca GECCO (angl. *Genetic and Evolutionary Computation Conference*). 28, 74, 78
- GO** globalna optimizacija (angl. *Global Optimization*). 9
- GP** genetsko programiranje (angl. *Genetic Programming*). 16
- GPS** globalni sistem pozicioniranja (angl. *Global Positioning System*). 53
- HKU** Mestna univerza v Hong Kongu (angl. *City University of Hong Kong*). 72
- HS** harmonično iskanje (angl. *Harmony Search*). 18
- HTS** razporejanje proizvodnje energije med hidroelektrarnami in termoelektrarnami (angl. *Hydro-Thermal Scheduling*). xii, 64, 66, 69, 74, 76, 77
- IEEE** Inštitut inženirjev elektrotehnike in elektronike (angl. *Institute of Electrical and Electronics Engineers*). iii, iv, viii, 28
- ImAppNIO** Izboljšanje uporabnosti optimizacije po vzoru iz narave z združenjem teorije in prakse WCCI (angl. *Improving Applicability of Nature-Inspired Optimisation by Joining Theory and Practice*). 71, 74

- INS** revija INS (angl. *Information Sciences*). 28
- JADE** samoprilagodljiva diferencialna evolucija (angl. *Self-adaptive differential evolution*). 22, 31, 33, 36, 39, 70, 71, 73
- JAXA** Japonska agencija za vesoljsko raziskovanje (angl. *Japan Aerospace Exploration Agency*). 72
- L-SHADE** SHADE z linearnim zmanjševanjem velikosti populacije (angl. *Linear population size reduction – SHADE*). xii, 31–33, 36, 39, 40, 70, 71, 73, 77
- LEM** učljiv evolucijski model (angl. *Learnable Evolution Model*). 18
- LPSR** linearno zmanjševanje velikosti populacije (angl. *Linear Population Size Reduction*). 31, 33, 36
- MA** memetski algoritem (angl. *Memetic Algorithm*). 17
- MO** večkriterijska optimizacija (angl. *Multi-objective Optimization*). 9
- MO-SADE** večkriterijska optimizacija s samoprilagodljivo diferencialno evolucijo (angl. *Multi-objective Optimization based on Self-adaptive Differential Evolution*). 28
- MOEA/D** večkriterijski evolucijski algoritem z razstavitvijo (angl. *Multi-objective Evolutionary Algorithm based on Decomposition*). 27
- MOPSO** večkriterijska optimizacija z rojem delcev (angl. *Multiobjective Particle Swarm Optimization*). 27
- MP** matematično programiranje (angl. *Mathematical Programming*). 6
- MTS** večtrajektorijsko iskanje (angl. *Multiple Trajectory Search*). 27
- NPdyn ϵ JDE** samoprilagodljiva diferencialna evolucija z zmanjševanjem velikosti populacije in obravnavo omejitev s prilagajanjem ϵ nivoja (angl. *self-adaptative differential evolution with population size reduction and constraint handling using ϵ level adjustment*). xii, 64–66
- NSGA-II** genetski algoritem z elitističnim nodominiranim urejanjem (angl. *Elitist Non-Dominated Sorting Genetic Algorithm*). 26, 27

- OOM** Oceanski observatorij Madeira (angl. *Madeira Ocean Observatory*, portugalsko *Observatório Oceânico da Madeira*). 72
- ORLTES** operacijske raziskave logističnih, transportnih in ekonomskih sistemov (angl. *Operational research of logistic, transport, and economic systems*). i, x, xii, 1, 2, 39, 55, 57, 58, 62, 65, 68, 69, 71, 74–79
- PDE** diferencialna evolucija s pareto fronto (angl. *Pareto-frontier Differential Evolution*). 27
- POGA** genetski algoritem s preferenčnim urejanjem (angl. *Preference Ordering Genetic Algorithm*). 27
- PPSN** konferenca PPSN (angl. *Parallel Problem Solving from Nature*). 74
- PSADEs** paralelna samoprilagodljiva diferencialna evolucija z nadomestnim modelom (angl. *Surrogate Parallel Self-adaptive Differential Evolution*). 64
- PSO** optimizacija z rojem delcev (angl. *Particle Swarm Optimization*). 17
- PUMP** Študija navpične oceanske črpalke v srednje velikih vrtinah (angl. *Study of the Vertical Oceanic Pump in mesoscale eddies*). 72
- SaDE** diferencialna evolucija, ki se sama prilagaja (angl. *Self-adaptive Differential Evolution*). 70, 71, 73
- SEMCCO & FANCCO** konferenca SEMCCO & FANCCO (angl. *Swarm, Evolutionary and Memetic Computing (SEMCCO) & Fuzzy And Neural Computing (FANCCO)*). 28, 74, 78
- SHADE** prilagodljivi algoritem diferencialne evolucije, temelječ na zgodovini uspehov (angl. *Success-History Based Adaptive Differential Evolution Algorithm*). vi, 31
- SI** inteligenca rojev (angl. *Swarm Intelligence*). 17
- SITMA** Služba za morsko tehnologijo (angl. *The Marine Technology Service*, špansko *Servicio General de Tecnología*). 71, 72
- SOCO** revija SOCO (angl. *Soft Computing*). 28
- SPEA2** močnostni Pareto evolucijski algoritem (angl. *Strength Pareto Evolutionary Algorithm*). 27

SWEVO revija SWEVO (angl. *Swarm and Evolutionary Computaton*). [28](#), [29](#), [33](#), [36](#), [78](#)

TEVC revija [IEEE](#) TEVC (angl. *IEEE Transactions on Evolutionary Computation*). [28](#), [78](#)

TPP termoelektrarna (angl. *Thermal Power Plant*). [64–66](#)

UGPP planiranje poti podmorskih sond (angl. *Underwater Glider Path Planning*). [ix](#), [xii](#), [30](#), [39](#), [40](#), [54–60](#), [62](#), [65](#), [69–71](#), [76–78](#)

ULPGC Univerza v Las Palmas de Gran Canaria (angl. *University of Las Palmas de Gran Canaria*, špansko *Universidad de Las Palmas de Gran Canaria*). [72](#)

WCCI konferenca WCCI (angl. *IEEE World Congress on Computational Intelligence*). [28](#)

Aktivno kazalo slik

1.1	Dvokriterijska funkcija $\mathbf{f}(\mathbf{x}) = (f_1(x_1, x_2), f_2(x_1, x_2))$ in dominantnost.	10
3.1	Arhitektura načrtovanega sistema algoritmov za operacijske raziskave logističnih, transportnih in ekonomskih sistemov — na primeru načrtovanja navigacije podvodnih sond.	30
4.1	Upodobitev proceduralnih modelov drevesa, podobnega bukvi, z različnim številom žil, žični model skeleta drevesne strukture brez listov in skrajno desno, enako drevo z drugačno vegetivno strukturo.	48
5.1	Scenariji za UGPP: izris površja trajektorij globinskih vzorčenj.	54
5.2	Primer različnih trajektorij za UGPP, izrisanih le na površju globinskih vzorčenj.	54
6.1	Nabor scenarijev UGPP z omejenim koridorjem.	57
6.2	Naloga UGPP s scenarijem, kjer je koridor omejen. Prikazane so še trajektorije, izračunane z različnimi EA.	58
6.3	Prikaz morskih tokov (modre puščice) na scenariju UGPP, kjer je koridor omejen (rdeči krogi), skupaj s prikazom nedopustnih poti (vijolične tangente od koridorja do trajektorije, ki je zelene barve).	59
6.4	Plovilo v scenarijih UGPP, ki se premika v prostoru po oceanu, predstavljen v različnih koordinatnih sistemih.	59
7.1	Izboljšava robustnosti: primerjava trajektorij dveh algoritmov, prvi (rumena barva) vrača precej različne rezultate, drugi (zelena barva) robustno in bolj učinkovito vrača boljšo trajektorijo.	61

9.1	Zgled primera za primerjavo algoritmov ORLTES. Vizualizacija večkratnega rangiranja algoritmov iz primera optimizacijskih funkcij pod različnimi nastavitvami parametrov za določen algoritem (več v [209]).	68
9.2	Vizualizacija primerjave rezultatov optimizacije trajektorij [231].	69
9.3	Primerjava rezultatov ekonomskega in okoljskega planiranja v proizvodnji električne energije s pristopom z DE [71].	71

Aktivno kazalo formul

1.1	Vektorska funkcija $\mathbf{f}(\mathbf{x})$	8
1.2	Posamezna omejitev $g_i(\mathbf{x})$	8
1.3	Omejitve $\mathbf{g}(\mathbf{x}) \geq \mathbf{0}$	8
1.5	Relacija dominantnosti $\mathbf{x} \preceq \mathbf{y}$	9
1.6	Odvod za funkcijo $f'(x)$	11
1.7	Lagrangeva optimizacijska metoda	11
1.8	Optimizacijski problem z več kriteriji	11
1.9	Metoda utežene vsote	11
1.10	Uteži optimizacijskih kriterijev	12
1.11	Enokriterijski problem z omejitvami	12
1.13	Enakostne in neenakostne omejitve	14
1.14	Povprečna vrednost kršenja vseh omejitev	14
1.16	Povprečna vrednost kršenja enakostnih ali neenakostnih omejitev	14
2.1	Izračun mutiranega vektorja	19
2.2	Strategija best/1	19
2.3	Strategija rand-to-best/1	19
2.4	Strategija rand/2	19
2.5	Strategija best/2	19
2.6	Binarno križanje	19
2.7	Eksponentno križanje	20
2.8	Selekcijski operator	20
2.9	Samoprilagajanje krmilnega parametra mutacije z naključjem	24
2.10	Samoprilagajanje krmilnega parametra križanja z naključjem	24
2.11	Obravnava omejitev s primerjavo med selekcijo	25
2.15	Metoda obravnave omejitev s prilagajanjem nivoja	25
4.1	Ovrednotenje fenotipa rekonstruiranega proceduralnega modela drevesa	49
4.2	Izračun razlik med slikami dreves	49
8.1	Izračun nadomestne matrike vrednosti za vsako od elektrarn	64
8.2	Kodiranje moči prve termoelektrarne	65
8.3	Kodiranje moči druge termoelektrarne	65
8.4	Kodiranje moči tretje termoelektrarne	65

Seznam postopkov

1	Osnovni algoritem DE [171] s strategijo rand/1/bin.	21
2	Algoritem jDE [21] s strategijo rand/1/bin.	24
3	Optimizacijski algoritem L-SHADE.	31
4	Optimizacijski algoritem DISH.	33
5	Optimizacijski algoritem DISHv.	36
6	Algoritem za ORLTES: L-SHADE@UGPP.	39
7	Izračun geometrijske strukture s proceduralnim drevesnim modelom. Rekurzivni algoritem pričnemo s klicem <code>odsekveje(0, 0, S, 1, l₀^{0,0}, I, I)</code> , kjer I označuje matriko identitete.	50
8	Morfološka rekonstrukcija kompleksnega drevesa iz narave(z *).	51
9	Ovrednotenje ustreznosti poti AUV (simulacija UGPP).	55
10	Postopek za avtonomno navigacijo pri UGPP z omejitvami koridorja: CUGPP@DE	58
11	Postopek HTS NPdynεjDE za izračun nadomestne matrike v dvonivojskem ekonomskem planiranju sistema sistemov	66

Predgovor

Predmet Operacijske raziskave logističnih, transportnih in ekonomskih sistemov ([ORLTES](#)) je uvrščen v tretjestopenjski bolonjski študijski program Računalništvo in informatika kot izbirna učna enota. Ta dokument predstavlja študijsko gradivo pri tem predmetu. Na naslednji strani sledi uvod, nato je v naslednjih poglavjih podano zbrano gradivo.

Zahvala

Zahvala dodeljenemu recenzentu, komisiji za e-založništvo in založniku.

Uvod — splošno o predmetu, vajah in nalogah

Študijska enota predmeta **Operacijske raziskave logističnih, transportnih in ekonomskih sistemov (ORLTES)** je sestavljena iz:

- predavanj in
- samostojnega dela.

V nadaljevanju je spodaj navedenih nekaj osnovnih informacij o predmetu, nato sledi še vsebinski pregled.

- **Opis predmeta v katalogu UM:** objavljen na <https://www.um.si/studij/podiplomski-studij/Lists/Studjiski%20programi%203stopnja/Attachments/38/U%C4%8Dni%20na%C4%8Drti.zip>.
- **Predavanja:** 60 ur (3 ure tedensko).
- **Samostojno delo:** 210 ur (14 ur tedensko).

Vsebinski pregled

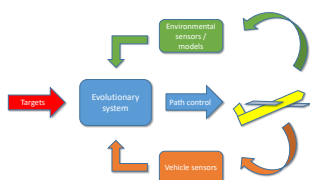
Za samostojno delo vsebinsko najprej potekajo predavanja, za katera so na voljo tudi gradiva iz literature oz. je obravnavana snov najprej predavana. Skozi samostojno delo se nato z uporabo praktičnih primerov podrobneje osvoji snov iz predavanj.



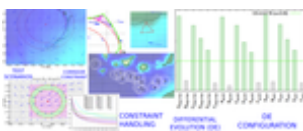
Primer prosojnic za predavanja

http://videlectures.net/solomon_zamuda_optimizacija/

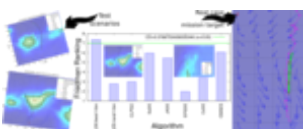
Predavanja zajemajo predvsem zbrano gradivo iz naslednje literature:



Aleš Zamuda and José Daniel Hernández Sosa. **Success history applied to expert system for underwater glider path planning using differential evolution.** *Expert Systems with Applications*, 2019, vol. 119, pp. 155–170. <http://dx.doi.org/10.1016/j.eswa.2018.10.048>



A. Zamuda, J. D. Hernández Sosa, L. Adler. **Constrained Differential Evolution Optimization for Underwater Glider Path Planning in Sub-mesoscale Eddy Sampling.** *Applied Soft Computing*, 2016, vol. 42, pp. 93-118. <http://dx.doi.org/10.1016/j.asoc.2016.01.038>



A. Zamuda, J. D. Hernández Sosa. **Differential Evolution and Underwater Glider Path Planning Applied to the Short-Term Opportunistic Sampling of Dynamic Mesoscale Ocean Structures.** *Applied Soft Computing*, vol. 24, November 2014, pp. 95-108. <http://dx.doi.org/10.1016/j.asoc.2014.06.048>

```

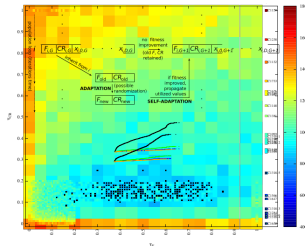
if  $S_F \neq \emptyset$  and  $S_{CR} \neq \emptyset$  then
  Update Update  $M_{F,k}$  (9) and  $M_{CR,k}$ 
  (10) with Lehmer mean computed
  by (11) with distance based weights
  from (17),  $k++$ ;
  if  $k > H$  then
     $k = 1$ ;
  end if
end if

```

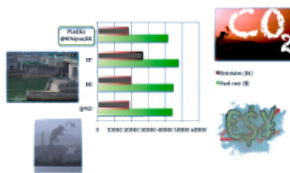
DISH

$$w_k = \frac{\sqrt{\sum_{j=1}^D (u_{k,j,G} - x_{k,j,G})^2}}{\sum_{m=1}^{S_{CR}} \sqrt{\sum_{j=1}^D (u_{m,j,G} - x_{m,j,G})^2}}$$

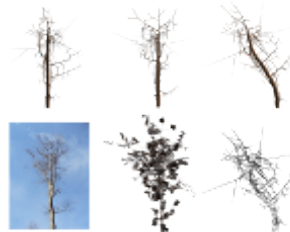
A. Viktorin, R. Senkerik, M. Pluhacek, T. Kadavy, A. Zamuda. **Distance Based Parameter Adaptation for Success-History based Differential Evolution**. Swarm and Evolutionary Computation, Available online 12 November 2018. <http://dx.doi.org/10.1016/j.swevo.2018.10.013>



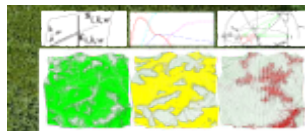
A. Zamuda, J. Brest. **Self-adaptive control parameters' randomization frequency and propagations in differential evolution**. Swarm and Evolutionary Computation, 2015, vol. 25C, pp. 72-99. <http://dx.doi.org/10.1016/j.swevo.2015.10.007>



A. Glotić, A. Zamuda. **Short-term combined economic and emission hydrothermal optimization by surrogate differential evolution**. Applied Energy, 1 March 2015, vol. 141, pp. 42-56. <http://dx.doi.org/10.1016/j.apenergy.2014.12.020>



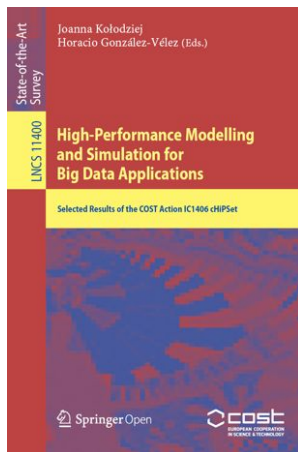
A. Zamuda, J. Brest. **Vectorized Procedural Models for Animated Trees Reconstruction using Differential Evolution**. Information Sciences, vol. 278, pp. 1-21, 2014. <http://dx.doi.org/10.1016/j.ins.2014.04.037>



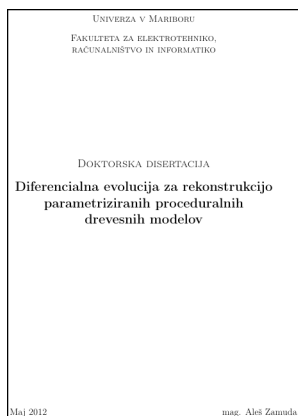
A. Zamuda, J. Brest. **Environmental Framework to Visualize Emergent Artificial Forest Ecosystems**. Information Sciences, vol. 220, pp. 522-540, 2013. <http://dx.doi.org/10.1016/j.ins.2012.07.031>



H. Hamann, Y. Khaluf, J. Botev, M. Divband Soorati, E. Ferrante, O. Kosak, J.-M. Montanier, S. Mostaghim, R. Redpath, J. Timmis, F. Veenstra, M. Wahby and A. Zamuda. **Hybrid Societies: Challenges and Perspectives in the Design of Collective Behavior in Self-organizing Systems**. Frontiers in Robotics and AI, 2016, vol. 3, no. 14. <http://dx.doi.org/10.3389/frobt.2016.00014>



A. Zamuda, V. Crescimanna, J. C. Burguillo, J. Matos Dias, K. Wegrzyn-Wolska, I. Rached, H. González-Vélez, R. Senkerik, C. Pop, T. Cioara, I. Salomie, A. Bracciali. **Forecasting Cryptocurrency Value by Sentiment Analysis: An HPC-Oriented Survey of the State-of-the-Art in the Cloud Era**. Kołodziej J., González-Vélez H. (eds) High-Performance Modelling and Simulation for Big Data Applications. Lecture Notes in Computer Science, vol 11400, 2019, pp. 325-349. http://dx.doi.org/10.1007/978-3-030-16272-6_12



A. Zamuda. Diferencialna evolucija za rekonstrukcijo parametriziranih proceduralnih drevesnih modelov: doktorska disertacija. Fakulteta za elektrotehniko, računalništvo in informatiko, 2012. <https://dk.um.si/Dokument.php?id=30084>

V nadaljevanju tega dokumenta sledijo poglavja z zbranimi gradivi, s katerimi študent(ka) nadalje gradi, analizira in vrednoti algoritme pri samostojnem delu.

Poglavje 1

Uvod: osnovni pojmi in primeri operacijskih raziskav logističnih, transportnih in ekonomskih sistemov

To poglavje podaja osnovne pojme, pri vsakem pa tudi primere, ki se pojavljajo pri logističnih, transportnih in ekonomskih sistemih. Pretežni del zbranega gradiva v tem (in drugem) poglavju je zajetega iz doktorske disertacije, h kateremu so dodane manjše osvežitve oz. še nekaj nedavnih člankov:



A. Zamuda. Diferencialna evolucija za rekonstrukcijo parametriziranih proceduralnih drevesnih modelov: doktorska disertacija. Fakulteta za elektrotehniko, računalništvo in informatiko, 2012. <https://dk.um.si/Dokument.php?id=30084>

Operacijske raziskave

Operacijske raziskave spadajo na področje računalništva in z njim povezane inteligence [179], kar bi lahko poimenovali tudi **umetna inteligenca** (angl. *Artificial Intelligence*) [126]. Še natančneje lahko področje opredelimo kot **računska inteligenca** (angl. *Computational Intelligence*) [56], ožje najdemo primere za **evolucijskega računanje** (angl. *Evolutionary Computation*) [180, 66, 10, 65, 157, 84] in prvotno **matematično programiranje** (angl. *Mathematical Programming*), t.j. matematične optimizacije [145].

V tem podpoglavju podajamo sorodna dela in ozadje operacijskih raziskav, od optimizacije in matematičnega programiranja, preko evolucijskih optimizacijskih algoritmov in diferencialne evolucije¹ do našihboljšav algoritmov diferencialne evolucije. Podamo še načrt svojega novega algoritma

¹Algoritem — postopek; daljša razlaga v [143]; beseda "diferencialna" se je ponekod ustalila kot prevod, nekateri pa uporabljajo tudi "diferenčna" — ime izhaja iz razlike med parametri, ki jo upoštevamo pri izračunu smeri spremembe, lahko bi prevedli tudi kot "razlikovalna" ali skupno kar "evolucija razlik".

za optimizacijo [213], predstavljenega v magistrski nalogi [204], in podamo njegov psevdokod.

Optimizacija in matematično programiranje

Optimizacija (angl. *optimization*) je postopek povečevanja ustreznosti izraženosti pozitivnih lastnosti (kakovosti) rešitve zadanega problema, ki ga s tem imenujemo **optimizacijski problem** (angl. *optimization problem*). Optimizacijske probleme najdemo v najrazličnejših domenah stvarstva in nestvarstva z različnim številom lastnosti (pri merjenju, dimenzij), od obstajanja, trajanja, velikosti, števnosti, razporejenosti, relativnosti, omejitve, porazdelitve, usklajenosti, spremenljivosti, tendence, verjemljivosti, cene, zanesljivosti in drugih domenskih metrik ter pojmov².

Najstarejši optimizacijski problem, glede na samo definicijo časa [60, 82], je porazdelitev stvari in protistvari v sistemu obstoja časa, prostora in drugih dimenzij, ki poteka glede na stvarne in nestvarne vplive (fizikalne materijske in antimaterijske sile). Tisti del materije, ki ga trenutno sprejemamo kot živi [142] in je optimiziran znotraj tega sistema, pogosto definiramo s pod sistemi atomov in molekul, ki se pojavljajo in z drugimi združujejo na tak način v organske sisteme, da opažen pojav pojmuje kot **življenje** [57]. Primeri teh podsistemov smo živa bitja na Zemlji in primeri optimizacijskih problemov naše življenje kot celota, kot skupine ali kot posamezniki. Kot primer živih bitij na Zemlji smo človeška bitja (lat. *Homo sapiens sapiens* [119]) [41, 156], ljudje združeni v civilizacije [32].

Kot tudi druga živa bitja, je življenje ljudi spremljala množica izzivov vplivanja na dogodke in s tem spreminjanja v začetku omenjene ustreznosti, v tem primeru kakovosti življenja ljudi. Kot podmnožico teh izzivov lahko prepoznamo npr. vpliv na skupke stvari in interakcijo z njimi, kot je npr. uporaba materije za prehranjevanje, gibanje po prostoru, razmnoževanje in uporaba predmetnih orodij³. Skozi čas in prostor so ljudje⁴ tako optimirali različna orodja za prehranjevanje, bivanje, umetnost in druge dejavnosti. Optimirani so bili postopki določitve dimenzij in izdelave orodij (npr. orožij, tudi za operacijske raziskave), razporeditve predmetov (npr. gradnja bivalnih prostorov ali velikih monumentov, kot so piramide in zigurati [8] ter stolpnice [96]), dolžin, razmerij in površin prostora (npr. štiri tisočletja optimizacije točnega razmerja π [149, 144] ali zlato razmerje samopodobnosti ϕ [121];

²Epistemološki opis teh pojmov najdemo v enciklopedijah [203, 192].

³Za definicijo orodja kot sredstva za preživetje oz. splošneje, povečanja ustreznosti, beri knjigo [46].

⁴Dosti krat lahko navedemo le določeno izvorno civilizacijo nekega izuma, saj nimamo beležk avtorstev in je izum toliko težje točno pripisati določenemu človeku.

oba uporabljana že tisočletja tudi v gradnji in slikarstvu [170]) ter merjenja časa (npr. s položaji nebesnih teles [67]).

S povečevanjem zahtevnosti rešenih nekaterih zgoraj navedenih izzivov se je povečevala tudi kompleksnost postopkov za določitev optimalnosti. Znanstveno področje optimizacije, ki obsega pripravo postopka za koračno izboljševanje optimalnosti rešitve, je matematično programiranje⁵. Enega prvih zapisanih optimizacijskih postopkov matematičnega programiranja je podal Kepler⁶. Newton čez nekaj desetletij kasneje predstavi račun, kalkulus [141], in z njegovo pomočjo poda gradientno metodo za pomikanje proti optimumu.

Ker jo bomo uporabljali tudi v nadaljevanju tega dela, z računsko psevdokodo [141] definirajmo optimizacijski postopek in podajmo definicije nekaterih pojmov, kot so jih videli uporabniki tega, za računalniško programiranje [104], zgodnjega postopkovnega kodnega zapisa. Za razumevanje definicije optimizacijskega postopka najprej definirajmo problem, ki ga postopek rešuje. Optimizacijski problem definiramo kot iskanje dopustnega vektorja spremenljivk oziroma iskalnih parametrov $\mathbf{x} = (x_1, x_2, \dots, x_D)$, ki optimira (minimira) vektorsko funkcijo $\mathbf{f}(\mathbf{x})$ z M kriteriji:

$$\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})). \quad (1.1)$$

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. Vektor \mathbf{x} pri tem mora zadoščati m omejitvam:

$$g_i(\mathbf{x}) \geq 0; \quad i = 1, 2, \dots, m, \quad (1.2)$$

s katerimi podamo omejitve v obliki več enokriterijskih funkcij. Tak vektor imenujemo dopustni vektor in vektor, ki krši omejitve, nedopustni vektor. Omejitve $g_i(\mathbf{x})$ lahko določajo npr. le definicijsko območje in zalogo vrednosti optimizirane funkcije:

$$\mathbf{g}(\mathbf{x}) \geq \mathbf{0} \Leftrightarrow (x_{\min,1}, x_{\max,1}, \dots, x_{\min,D}, x_{\max,D}, f_{\min,1}, f_{\max,1}, \dots, f_{\min,M}, f_{\max,M}). \quad (1.3)$$

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.

⁵Pomen besede programiranje se v terminu navezuje na načrtovanje postopka, upravljanje, in ne na kodiranje s programskim jezikom, kot bi lahko morda pomislili [145].

⁶V svojem predlogu rešitve problema izbire soproge, ali, kot je kasneje formulacijo spremenil, osebne tajnice [102].

V kolikor želimo najti optimalni vektor iz celotnega dopustnega območja, tak postopek imenujemo **globalna optimizacija** (angl. *Global Optimization*). Področje matematičnega programiranja in posebej globalna optimizacija obsežnejši razmah doživi v času po drugi svetovni vojni, po predstavitvi digitalnih računalnikov, ki med drugim omogočijo avtomatsko izvajanje postopkov matematičnega programiranja [66]. Tukaj omenimo le algoritem preproste minimizacije s spuščanjem po hribu navzdol, Nelder-Mead [136], nadzorovano naključno preiskovanje [152], simulirano ohlajanje [101] in iskanje s tabuji [72].

Kasneje se pojavi še numerična **večkriterijska optimizacija** (angl. *Multi-objective Optimization*) [50]. Pri tej imamo na izbiro več kriterijev za ovrednotenje iste rešitve. Ker so si ti kriteriji pogosto v nasprotju, to pomeni, da izboljšanje neke odločitve po enem kriteriju lahko povzroči njeno poslabšanje po drugih kriterijih. Takrat nimamo opravka samo z eno optimalno možno rešitvijo oz. odločitvijo, temveč z množico optimalnih rešitev. Ta pristop večkriterijske optimizacije, t.j. sočasne optimizacije medsebojno konfliktnih kriterijev, lahko uporabimo pri boljši podpori odločanju, saj odločanje zahteva izbiro ene izmed različnih danih možnosti, pogosto pa si ne želimo že v začetku postaviti prioriteta za kriterije.

Definirajmo še relacijo **dominantnosti** (angl. *dominance*), označimo jo z $\mathbf{x} \preceq \mathbf{y}$. Vektor \mathbf{x} dominira vektor \mathbf{y} , če velja:

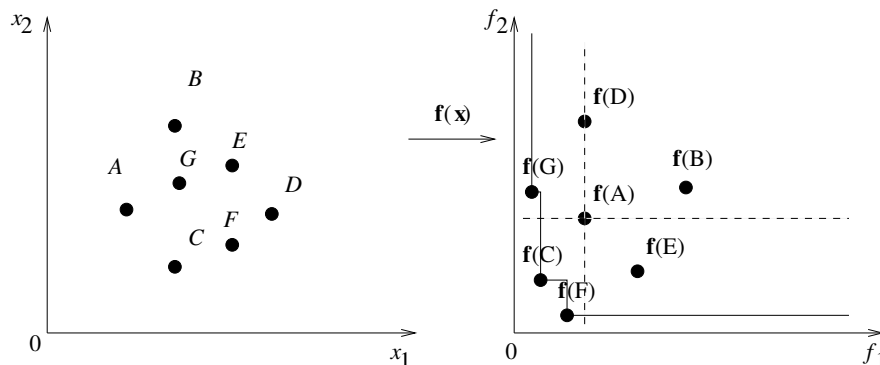
$$f_i(\mathbf{x}) \leq f_i(\mathbf{y}), \quad \forall \{i = 1, \dots, M\}, \quad (1.4)$$

$$\exists j \in \{1, \dots, M\} : f_j(\mathbf{x}) < f_j(\mathbf{y}), \quad (1.5)$$

torej da glede na izjavo 1.4 vektor \mathbf{x} ni slabše ovrednoten od vektorja \mathbf{y} po vseh kriterijih in glede na izjavo 1.5 je vektor \mathbf{x} boljše ovrednoten od vektorja \mathbf{y} po vsaj enem kriteriju.

Prostor kriterijev \mathfrak{R}^M je z relacijo \preceq delno urejen, saj nekateri vektorji medsebojno niso primerljivi. Slikovna razjasnitev pojma dominantnosti je vidna na sliki 1.1. Kot je videti iz slike, je vektor A boljše ovrednoten (boljši) od vektorjev B in D ter neprimerljiv (glede na dane kriterije) z E . Pri tem se neprimerljivost vektorja A nanaša na nedominantnost pripadajočega kriterijskega vektorja $\mathbf{f}(A)$, saj primerjavi po kriterijih f_1 oz. f_2 za $\mathbf{f}(A)$ z $\mathbf{f}(E)$ nista istoznačni.

Pareto optimalni vektor je tak vektor \mathbf{x} , ki ga ne dominira noben drug dopustni vektor \mathbf{z} in velja $\mathbf{f}(\mathbf{z}) \preceq \mathbf{f}(\mathbf{x})$. **Množica nedominiranih vektorjev** v neki množici vektorjev je množica vseh tistih vektorjev, ki jih ne dominira noben vektor iz te neke množice. Kriterijski vektorji teh vektorjev spremenljivk iz množice nedominiranih vektorjev tvorijo **nedominirano**



Slika 1.1: Dvokriterijska funkcija $\mathbf{f}(\mathbf{x}) = (f_1(x_1, x_2), f_2(x_1, x_2))$ in dominantnost.

fronto. Množico nedominiranih vektorjev celotnega prostora dopustnih vektorjev imenujemo **Pareto optimalna množica**, njene elemente **Pareto optimalni vektorji** in pripadajočo nedominirano fronto v prostoru kriterijev imenujemo **Pareto optimalna fronta**. Na sliki 1.1 tvorijo množico nedominiranih vektorjev vektorji G , C in F , saj niso dominirani od nobenega drugega vektorja, t.j. njihovi kriterijski vektorji $\mathbf{f}(G)$, $\mathbf{f}(C)$ in $\mathbf{f}(F)$ imajo manjše vrednosti od ostalih po vsaj enem kriteriju.

Pristope večkriterijskega optimiranja lahko, predvsem iz zgodovinskih razlogov [102, 39, 40], 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). Če vnaprej poznamo ali določimo informacijo o pomembnosti kriterijev (metakriterij), nam to omogoči uporabo prednostnega pristopa. Sicer uporabimo drugi, t.i. **idealni pristop**, ki najprej poišče Pareto optimalno množico vektorjev. Nato iz nje sami izberemo vektor, ki nam najbolj ustreza. Izbiro najustreznejšega vektorja naredimo na podlagi dodatne informacije o pomembnosti kriterijev, ki navadno ni znana vnaprej in jo lahko izvemo šele po predstavitvi rezultatov optimizacije. Zato je idealni pristop bolj pregleden in manj subjektiven od prednostnega pristopa.

Pri optimizaciji z idealnim pristopom želimo, da večkriterijska optimizacijska metoda najde čim več Pareto optimalnih vektorjev. Ker izmed teh vektorjev (s pomočjo informacije o pomembnosti kriterijev) izbiramo najboljšega, si želimo, da so dobljeni vektorji kar se da enakomerno razporejeni po prostoru kriterijev (t.j. odločitvenem prostoru). Nalogo večkriterijskega optimiranja lahko tako pretvorimo v nalogo iskanja množice nedominiranih vektorjev, ki jo imenujemo **aproksimacijska množica** (angl. *approximation set*). Za elemente aproksimacijske množice želimo, da so čim bližje Pareto

optimalni fronti in da so enakomerno razporejeni vzdolž Pareto optimalne fronte. Ti dve željeni lastnosti algoritmov za večkriterijsko optimizacijo si pogosto nasprotujeta. Zato je že sama zadostitev željenih lastnosti večkriterijske optimizacije lahko večkriterijska. Pogosto zanemarjena, tretja zelena lastnost algoritmov večkriterijske optimizacije je čim krajši čas računanja za doseg prvih dveh ciljev večkriterijske optimizacije [49]. Primer zmanjšanja časa računanja je uporaba superskalarnih arhitektur, kot je CUDA, na kateri smo predstavili primer izračunavanja enega od realnih industrijskih problemov [225]. Z algoritmom večkriterijske optimizacije želimo doseči tudi čim večjo robustnost ob različnih zagonih algoritma [48, 59], t.j. da algoritem s čim večjo verjetnostjo uspešno optimizira dan problem.

Oglejmo si osnovno Newtonovo metodo in dve najbolj priljubljeni osnovni metodi za večkriterijsko optimizacijo: metodo utežene vsote in metodo ϵ -omejitev. Več o osnovnih metodah najdemo v [50].

Newton je za funkcijo podal izračun gradienta, s pomočjo simboličnega ali numeričnega odvoda funkcije:

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}, \quad (1.6)$$

kasneje Newtonovo metodo dopolni Lagrange, v postopku za določitev optimuma, tako da gradient prišteje, integrira, k izhodiščni točki:

$$f^*(x) = f(x) + \Delta x f'(x), \quad (1.7)$$

pri čemer Δx označuje spremembo po spremenljivki x in $\Delta f(x)$ pripadajočo spremembo funkcijske vrednosti ter $f^*(x)$ označuje optimiran približek in $f'(x)$ diferencial oz. spremembo funkcije $f(x)$ za interval Δx .

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

$$\text{optimiraj } \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_M(\mathbf{x})) \quad (1.8)$$

dobimo nalogo:

$$\text{optimiraj } \mathbf{g}(\mathbf{x}) = (w_1 f_1(\mathbf{x}) + \dots + w_M f_M(\mathbf{x})), \quad (1.9)$$

ki jo rešimo z eno od metod za enokriterijsko optimiziranje. Ker se optimum te naloge ne spremeni, če uteži pomnožimo s konstanto, navadno predpostavljamo, da zanje velja

$$w_i \in [0, 1] \text{ in } \sum_{i=1}^M w_i = 1. \quad (1.10)$$

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 pa ima več slabosti. Prva očitna slabost te metode je lahko, da zahteva vektor uteži. Če ne poznamo prednosti kriterijev, je določitev takšnega vektorja zahtevna naloga. Poleg tega moramo, da bi dobili več Pareto optimalnih vektorjev, metodo uporabiti večkrat – vsakič z drugačno nastavitvijo uteži. Vendar je pri tem treba paziti, saj z enakomerno porazdelitvijo vektorjev uteži ne najdemo nujno enakomerno porazdeljenih kriterijskih vektorjev na Pareto optimalni fronti.

Ker nam odvisnost med vektorjem uteži in vektorji na Pareto optimalni fronti navadno ni poznana, težko postavimo nastavitve uteži, ki bo našla željen kriterijski vektor Pareto optimalne fronte. Poleg tega različni vektorji uteži ne najdejo nujno različnih kriterijskih vektorjev na Pareto optimalni fronti. Podobno tudi en vektor uteži lahko vrne različne kriterijske vektorje Pareto optimalne fronte. Če je večkriterijski optimizacijski problem konveksen, potem 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 te vrednosti predhodno normirati.

Če se želimo izogniti težavam, ki so značilne za metodo utežene vsote pri nekonveksnih Pareto optimalnih frontah, lahko uporabimo **metodo ϵ -omejitve** (*ϵ -constraint method*). Predlagali so jo Haimes in sodelavci [76] in rešuje večkriterijski optimizacijski problem tako, da optimira samo en kriterij, ostale pa doda med omejitve. Preoblikovana optimizacijska naloga se tako glasi:

$$\text{optimiraj } f_j(\mathbf{x}) \text{ pri pogojih } f_i(\mathbf{x}) \leq \epsilon_i, \quad i = 1, \dots, M \text{ in } i \neq j. \quad (1.11)$$

V tej obliki parameter ϵ_i označuje zgornjo mejo za vrednost f_i in ne pomeni nujno majhne vrednosti blizu nič. Kot zanimivost lahko povemo še, da

se metoda ϵ -omejitev uspešno uporablja tudi na področju enokriterijske optimizacije z omejitvami [172], 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 pomembnosti kriterijev. Tokrat je ta podana v obliki vektorja omejitev. Če izberemo drugačen vektor omejitev ali če za optimiziranje izberemo drug kriterij, kot rezultat dobimo drugačno množico kriterijskih vektorjev. Ker omejujemo kriterije (in ne spremenljivk), nam ta metoda omogoča lažje lokaliziranje rešitev v željenih 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 vrednosti ovrednotenih kriterijev, saj razlike v vrednosti kriterijev upoštevamo v vektorju omejitev.

Če želimo v optimizaciji posebej obravnavati še kriterijske omejitve, imenujemo takšno optimizacijo, **optimizacija z omejitvami** (angl. *constraint optimization*). Skozi zgodovino najdemo definicijo številnih metod za obravnavo omejitev v optimizacijskih algoritmih. Michalewicz in sod. [132, 108] so te metode uvrstili v štiri skupine:

1. *Metode z ohranjanjem dopustnih rešitev.* Ta skupina metod zajema posebne operatorje, ki dopustne vektorje staršev vedno preslika v dopustne nove vektorje potomcev. Začetna populacija mora vsebovati le dopustne vektorje in omejitve so lahko le linearne.
2. *Metode s kazensko funkcijo.* Pri tej skupini metod z zunanjo kazensko funkcijo ocenimo kazen nedopustnih rešitev. Metode znotraj te skupine se razlikujejo v podrobnostih, kako pripraviti kazensko funkcijo in kako jo uporabiti pri kaznovanju nedopustnih posameznikov.
3. *Metode s strogim razlikovanjem med dopustnimi in nedopustnimi vektorji.* Metode v tej skupini strogo ločijo med dopustnimi in nedopustnimi vektorji. Primer takšne metode je ločitev vsakega dopustnega vektorja \mathbf{x} in vsakega nedopustnega vektorja \mathbf{y} s strogo urejenostjo, t.j. vsak dopustni vektor je boljši od vsakega nedopustnega vektorja.
4. *Druge mešane metode.* Tu združimo metode iz več področij.

Večino problemov z omejitvami je moč obravnavati z metodami s kazensko funkcijo. Funkcija meri oceno kršitve omejitev, ki so lahko enakostne ali neenakostne. Vektor \mathbf{x} je *dopusten* za neenakostne omejitve $g_i(\mathbf{x}) \leq 0$ in enakostne omejitve $h_j(\mathbf{x}) = 0$, če:

$$g_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, q, \quad (1.12)$$

$$|h_j(\mathbf{x})| - \epsilon \leq 0, \quad j = q + 1, \dots, m, \quad (1.13)$$

kjer enakostne omejitve pred oceno dopustnosti pretvorimo v neenakostne omejitve za nek interval ϵ , npr. $\epsilon = 0,0001$ za funkcije CEC 2016 [116]. Povprečno vrednost kršenja vseh omejitev \bar{v} lahko definiramo kot [116]:

$$\bar{v} = \frac{(\sum_{i=1}^q G_i(\mathbf{x}) + \sum_{j=q+1}^m H_j(\mathbf{x}))}{m}, \quad (1.14)$$

kjer

$$G_i(\mathbf{x}) = \begin{cases} g_i(\mathbf{x}), & g_i(\mathbf{x}) > 0, \\ 0, & g_i(\mathbf{x}) \leq 0, \end{cases} \quad (1.15)$$

$$H_j(\mathbf{x}) = \begin{cases} |h_j(\mathbf{x})|, & |h_j(\mathbf{x})| - \epsilon > 0, \\ 0, & |h_j(\mathbf{x})| - \epsilon \leq 0. \end{cases} \quad (1.16)$$

Ta kriterij pravi, da je vsota kršitev vseh omejitev enaka nič za dopustne vektorje in večja od nič, ko je kršena vsaj ena omejitev. Da bi našli dopustne vektorje, ta kriterij uporabimo za vodenje iskanja proti območjem dopustnih vektorjev. Implementacije tega vodenja v optimizacijskih algoritmih zajemajo obsežen nabor idej, povzetek pristopov najdemo v [131, 38].

Matematično programiranje in evolucijsko računanje

V 19. stoletju našega štetja se v raziskavah iz bioloških znanosti pojavi vseprisotna evolucijska teorija darvinizma [93] in nato z njo vezan neodarvinizem [162], ki poleg Darwinovega zakona naravne selekcije [41] vsebuje še Weismannovo teorijo dedne plazme [191] in Mendlove zakone dedovanja [127]. Izraz neodarvinizem je bil tudi še kasneje populariziran [46] v namene sklicevanja na moderno evolucijsko sintezo [91]. Teorije o evoluciji oz. vsaj potomstvu so se dopolnjevale tisočletja in pravkar citirani pristopi zajemajo zgolj bolj citirana dela, medtem ko obstajajo še številna druga dela (npr. predhodna E. Darwina [43] in tudi analogna A. R. Wallacea [188, 42] leto pred objavo [41]). Obstajajo tudi naravne teorije iz antike in še starejše, ki nastanek življenja omejujejo zgolj na kreacionizem, najpogosteje iz vode kot generativnega elementa, z načrtovano pojavitvijo po vodni poplavi [176, 159, 81, 135, 36], včasih ex nihilo [34, 159, 81], ki pa so precej manj podrobne, bolj skope z empiričnimi dokazi in pogosto v nasprotju s prej omenjenimi. Pojem neodarvinizma z modelom odmisli (t.j. metodično abstrahira [83]) dinamiko življenja zgolj na naslednje mehanizme [58, 128]:

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

Evolucijska ideologija v 20. stoletju vpliva na področje matematičnega programiranja tako, da znotraj nastajajočega skupka algoritmov umetne inteligence [179, 126], nastane skupno področje evolucijskega računanja [180, 66, 10, 65, 157, 84]. Evolucijsko računanje je tako področje algoritmov, ki po evolucijskem vzoru simulirajo nek naravni proces. Ta naravni proces je abstrahiran v model, vsebovan v posameznem algoritmu [64]. Večina teh modelov temelji na pojmu neodarvinizma [92], ki je tako njihova biološka podlaga. Modeli so najpogosteje definirani na treh stopnjah podrobnosti: najprej opisani domensko prozaično (t.j. za opis so uporabljeni pojmi določene domene iz narave), nato zapisani z računskimi postopki (t.j. definirane so še računske operacije med pojmi) in še kodirani v programskem jeziku (t.j. definirani s tipiziranimi programskimi stavki in strukturami). Te tri stopnje podrobnosti nam dajo vsaj tri izhodišča, kako narediti primerjavo podobnosti modelov, kje med njimi ločiti in v kakšne vrste jih lahko razdelimo.

Poglavje 2

Arhitektura algoritmov in taksonomija algoritmov za operacijske raziskave logističnih, transportnih in ekonomskih sistemov

Glede na domensko prozaične definicije njihovih modelov (danes obstaja primeren članek na to temo, ki ga bralec lahko prebere v [168]), ločimo več vrst algoritmov iz nabora evolucijskega računanja in ti pogosto služijo za optimizacijo. Nekatere posamezne optimizacijske algoritme znotraj evolucijskega računanja lahko po dogovorjenem pojmovnem izboru označimo tudi kot posamezni **evolucijski algoritem** (angl. *Evolutionary Algorithm*). Primeri nekaterih algoritmov iz tega nabora so:

- **genetski algoritem** (angl. *Genetic Algorithm*) [75] posameznike izdeluje tako, da realna števila v posameznem vektorju kodira z binarnim nizom, nato z operatorji mutacije in križanja spreminja posamezne bite v binarni predstavitvi. Operator mutacije navadno deluje uniformno naključno nad vsakim bitom binarno kodiranega posameznika. Operator križanja združi posamezne bite iz več posameznikov (staršev) v novega posameznika, tako da jemlje nekaj bitov iz vsakega od staršev,
- **genetsko programiranje** (angl. *Genetic Programming*) [107, 113] gradi drevesno strukturo, kot jo pozna programski jezik lisp [125], uporabljeni genetski operatorji so posebej prirejani za operacije nad drevesom,
- **evolucijsko programiranje** (angl. *Evolutionary Programming*) uporablja končni avtomat za predstavitev in je bilo predstavljeno kot pristop v umetni inteligenci [65], enega od kasnejših algoritmov pa predstavlja delo [199], ki daje tudi primerjalno osnovo za številne medsebojne primerjave ostalih algoritmov nad standardnim izbranim naborom testnih funkcij za enokriterijsko optimizacijo,

- **memetski algoritem** (angl. *Memetic Algorithm*) združuje evolucijske pristope s pristopi iskustveno prilagojenih (hevrističnih) algoritmov za reševanje ožje določenih optimizacijskih problemov; primer pregleda področja zajema [139],
- **evolucijska strategija** (angl. *Evolution Strategy*), ki iskalne parametre kodira z realnimi vektorji, v posameznem vektorju pa kodira tudi krmilne parametre iskalnega algoritma (pregled podan v [13, 7, 165], močnejši predstavniki izpeljav so [80, 94, 70]),
- **umetni imunski sistem** (angl. *Artificial Immune System*), ki uporablja vzor imunskih sistemov za iskanje rešitve proti sovražnim telesom [100],
- **inteligenca rojev** (angl. *Swarm Intelligence*), ki uporablja nek altruistični vzor iz rojev ali kolonij s skupno inteligenco, ki se pojavi ob sodelovanju in izmenjavi med posamezniki [12];
- **optimizacija z rojem delcev** (angl. *Particle Swarm Optimization*) poleg vsakega vektorja spremenljivk (delca) hrani še njegov diferenčni vektor, ki modelira hitrost premikanja delca v geografskem (kriterijskem) prostoru [99, 240, 238],
- **optimizacija s kolonijo mravelj** (angl. *Ant Colony Optimization*) modelira pomnjenje zgodovine dobrih iskalnih parametrov po vzoru mravelj, ki v primeru najdene hrane za seboj puščajo feromonske sledi [55, 106, 22],
- **optimizacija s čebelami** (angl. *Bees Algorithm*), ki posnema delovanje čebel pri nabiranju hrane [148]; daljšo razlago in preskus algoritma najdemo v diplomski nalogi [17],
- **optimizacija s kolonijami čebel** (angl. *Artificial Bee Colony*), ki temelji na vzorih algoritma za optimizacijo s čebelami in dodaja razširitev s kolonijami [98],
- **kukavičje iskanje** (angl. *Cuckoo Search*), ki posnema parazitstvo valjenja kukavičjih zarodkov v tujih gnezdih [196],
- **bakterijsko preiskovanje** (angl. *Bacterial Foraging Optimization*), ki posnema iskanje hrane po vzoru bakterij *E. coli* [147],
- **umetne travne kolonije** (angl. *Artificial Weed Colonies*), ki posnema rast trave; algoritem so primerjali tudi z enim od DE v prispevku [111],

- **diferencialna evolucija** (angl. *Differential Evolution*), ki združuje simpleks metodo Nelder-Mead [136], nadzorovano naključno preiskovanje [152] in nekatere evolucijske mehanizme ter je opisana v naslednjem razdelku in še
- nabor drugih algoritmov znotraj evolucijskega računanja, kot so **harmonično iskanje** (angl. *Harmony Search*) [69], **učljiv evolucijski model** (angl. *Learnable Evolution Model*) [193] in **algoritem kresničke** (angl. *Firefly Algorithm*) [195], in ti algoritmi povečini vsebujejo vsaj nekaj gradnikov EA.

Algoritem diferencialne evolucije

DE [171] je **populacijski algoritem**, ki se uspešno uporablja za globalno optimizacijo **realno kodiranih** [74] numeričnih funkcij. Algoritem zaradi svoje narave prilagajanja optimizacijskega koraka (pri tvorbi potomcev se upoštevajo difference med starši) in **stabilnega iskanja** (angl. *steady-state search*) z elitističnim selekcijskim mehanizmom navadno daje boljše rezultate od drugih evolucijskih algoritmov [21, 129, 138, 45]. Algoritem diferencialne evolucije [171] sestoji iz glavne evolucijske zanke, v kateri z evolucijskimi operatorji mutacije, križanja in selekcije postopno in vzporedno po generacijah izboljšuje približek iskane rešitve. Evolucijski operatorji vplivajo na vsak primerok \mathbf{x}_i , $\forall i \in \{0, 1, \dots, NP - 1\}$ v populaciji rešitev, iz katerih se zgradi nova populacija. Posameznike po tem ovrednotimo, skupno število ovrednotenih posameznikov pa označimo s FEs. Algoritem DE je tudi preprost in ima malo parametrov, F – **faktor skaliranja** diferenčnega vektorja za mutacijo, CR – **stopnja križanja**, NP – **velikost populacije** ter včasih tudi s – **strategija** mutacije in križanja.

Natančneje razložimo osnovni algoritem diferencialne evolucije, ki deluje za enokriterijsko optimizacijo [171]. DE sestoji iz glavne evolucijske zanke, v kateri 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 [95]. Evolucijski operatorji vplivajo na vsak primerok \mathbf{x}_i , $\forall i \in \{0, 1, \dots, NP - 1\}$ v populaciji rešitev, iz katerih se zgradi nova populacija za naslednjo **generacijo** (angl. *generation*). Eno izdelavo novega posameznika poimenujemo **iteracija** (angl. *iteration*); tega pojma ne smemo zamenjevati s pojmom generacije. V vsaki iteraciji operator mutacije izračuna **mutiran vektor** (angl. *donor vec-*

tor) $\mathbf{v}_{i,g+1}$:

$$\mathbf{v}_{i,g+1} = \mathbf{x}_{r_1,g} + F \times (\mathbf{x}_{r_2,g} - \mathbf{x}_{r_3,g}), \quad (2.1)$$

kjer so $r_1, r_2, r_3 \in \{0, 1, \dots, NP - 1\}$ paroma in od i različni indeksi primerkov iz populacije v generaciji $g \in \{0, 1, \dots, G - 1\}$, $i \in \{0, 1, \dots, NP - 1\}$ in $F \in [0, 2]$. Vektor r_1 imenujemo **osnovni vektor** (angl. *base vector*). Izraz $\mathbf{x}_{r_2,g} - \mathbf{x}_{r_3,g}$ imenujemo **diferenčni vektor** (angl. *difference vector*) in po množenju s faktorjem ojačanja F , **utežen diferenčni vektor** (angl. *weighted difference vector*). Pravkar opisano strategijo mutacije imenujemo rand/1, obstajajo še druge [21], kot je strategija best/1:

$$\mathbf{v}_{i,g+1} = \mathbf{x}_{\text{best},g} + F \times (\mathbf{x}_{r_1,g} - \mathbf{x}_{r_2,g}), \quad (2.2)$$

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

$$\mathbf{v}_{i,g+1} = \mathbf{x}_{i,g} + F \times (\mathbf{x}_{\text{best},g} - \mathbf{x}_{i,g}) + F \times (\mathbf{x}_{r_1,g} - \mathbf{x}_{r_2,g}). \quad (2.3)$$

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

$$\mathbf{v}_{i,g+1} = \mathbf{x}_{r_1,g} + F \times (\mathbf{x}_{r_2,g} - \mathbf{x}_{r_3,g}) + F \times (\mathbf{x}_{r_4,g} - \mathbf{x}_{r_5,g}) \quad (2.4)$$

in strategijo mutacije z najboljšim vektorjem, best/2:

$$\mathbf{v}_{i,g+1} = \mathbf{x}_{\text{best},g} + F \times (\mathbf{x}_{r_1,g} - \mathbf{x}_{r_2,g}) + F \times (\mathbf{x}_{r_3,g} - \mathbf{x}_{r_4,g}). \quad (2.5)$$

Po mutaciji dobljeni mutiran vektor $\mathbf{v}_{i,g+1}$ križamo s **ciljnim vektorjem** (angl. *target vector*) $\mathbf{x}_{i,g}$ in tako dobimo **poskusni vektor** (angl. *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:

$$u_{i,j,g+1} = \begin{cases} v_{i,j,g+1} & \text{če } \text{rand}(0, 1) \leq CR \text{ ali } j = j_{\text{rand}} \\ x_{i,j,g} & \text{sicer} \end{cases}, \quad (2.6)$$

kjer $j \in \{1, \dots, D\}$ označuje j -to komponento vektorja v prostoru z D dimenzijami, funkcija $\text{rand}(0, 1) \in [0, 1]$ označuje vzorčenje uniformno (psevdo)

naključno porazdeljenega števila in j_{rand} izbiro uniformno naključnega indeksa 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:

$$u_{i,j,g+1} = \begin{cases} v_{i,j,g+1} & \text{če } j = \langle n \rangle_D, \langle n+1 \rangle_D, \dots, \langle n+l-1 \rangle_D, \\ x_{i,j,g} & \text{sicer} \end{cases}, \quad (2.7)$$

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 ustrežnejša (oz. boljša, saj boljše pomeni manjšo vrednost) od ocene ciljnega vektorja $f(\mathbf{x}_{i,g})$, kjer algoritem **DE** privzema elitističen princip preživetja najustrežnejših [41]:

$$\mathbf{x}_{i,g+1} = \begin{cases} \mathbf{u}_{i,g+1} & \text{če } f(\mathbf{u}_{i,g+1}) < f(\mathbf{x}_{i,g}) \\ \mathbf{x}_{i,g} & \text{sicer} \end{cases}. \quad (2.8)$$

Tabela 2.1: Diferencialna evolucija – kombinacije operatorjev križanja in mutacije ter njihovo poimenovanje v implementaciji [150].

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

S kombinacijo naštetih operatorjev mutacije in križanja se je z objavo implementacije algoritma **DE** [150] pojavilo oštevilčenje imen teh kombinacij v algoritmu **DE**, ki jih vidimo v tabeli 2.1. V kolikor 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, ki jo prikazuje postopek 1. Strategije 2, 5 in 10 lahko uporabimo, če ostale odpovejo, obstaja pa tudi veliko število izboljšav in dopolnitev osnovnega algoritma diferencialne evolucije, ki jih navajamo v nadaljevanju.

Postopek 1 Osnovni algoritem DE [171] s strategijo rand/1/bin.

Vhod: $f(\mathbf{x})$ — kriterijska funkcija; D , NP , G , F , CR — krmilni parametri DE.

Izhod: \mathbf{x}_{best} — optimirani parametri za dano kriterijsko funkcijo.

- 1: Uniformno naključno generiraj začetno populacijo DE $\mathbf{x}_{i,0}$ za $i = 1..NP$;
 - 2: **for** DE generacijska zanka g (dokler $g < G$) **do**
 - 3: **for** DE iteracijska zanka i (za vse vektorje $\mathbf{x}_{i,g}$ v trenutni populaciji) **do**
 - 4: DE izračun poskusnega vektorja $\mathbf{x}_{i,g}$ (mutacija, križanje):
 - 5: $\mathbf{v}_{i,g+1} = \mathbf{x}_{r_1,g} + F \times (\mathbf{x}_{r_2,g} - \mathbf{x}_{r_3,g})$;
 - 6: $u_{i,j,g+1} = \begin{cases} v_{i,j,g+1} & \text{če } rand(0, 1) \leq CR \text{ ali } j = j_{\text{rand}}; \\ x_{i,j,g} & \text{sicer} \end{cases}$;
 - 7: DE selekcija z oceno ustreznosti $f(\mathbf{u}_{i,G+1})$:
 - 8: $\mathbf{x}_{i,g+1} = \begin{cases} \mathbf{u}_{i,g+1} & \text{če } f(\mathbf{u}_{i,g+1}) < f(\mathbf{x}_{i,g}); \\ \mathbf{x}_{i,g} & \text{sicer} \end{cases}$;
 - 9: **end for**
 - 10: **end for**
 - 11: **vrni** najboljši najden posameznik in ga poimenuj \mathbf{x}_{best} ;
-

Osnovni algoritem diferencialne evolucije ima številne razširitve. Že Lampinen je zbral tedaj obsežno zbirko izboljšav algoritma DE [112]. Sedaj obstaja precej izboljšav osnovnega algoritma DE, omenimo le nekaj pomembnejših:

- DE s trigonometrično mutacijo avtorjev Fan in Lampinen [62], ki je posebej primeren za reševanje rotiranih problemov,
- samoprilagodljivi DE avtorjev Liang, Yao in Newton [117],
- večkriterijski algoritem DE avtorjev Xue, Sanderson in Graves [194],
- algoritem DE z mehko mutacijo avtorjev Liu in Lampinen [120],
- samoprilagodljivi DE avtorjev Omran, Salman in Engelbrecht [146],

- algoritem [DE](#) avtorjev Qin in Suganthan [154],
- algoritem [DE](#) s prilagajanjem dopustnosti za optimizacijo z omejitvami avtorjev Takahama, Sakai in Iwane [172],
- algoritem [DE](#) s prilagajanjem velikosti populacije avtorja J. Teo [175],
- algoritem [DE](#) s prilagajanjem krmilnega parametra *CR* avtorja M. M. Ali [3],
- algoritem [DE](#) s samoprilagajanjem krmilnih parametrov *F* in *CR* avtorjev Huang, Qin in Suganthan [87],
- algoritem [DE](#) z logističnim samoprilagajanjem krmilnih parametrov *F* in *CR* (jDE) avtorjev Brest s sod. [21, 24], ki je skupaj s svojimi izboljšavami opisan kasneje v nadaljevanju tega poglavja,
- algoritem [DE](#) s prilagajanjem izbrane mutacijske strategije avtorjev Qin, Huang in Suganthan [153],
- algoritem [DE](#) za optimizacijo problemov z velikim številom dimenzij avtorjev Yang, Tang in Yao [198],
- algoritem [DE](#) z novim logističnim samoprilagajanjem krmilnih parametrov *F* in *CR* in spremenjeno mutacijsko strategijo ([samoprilagodljiva diferencialna evolucija \(angl. *Self-adaptive differential evolution*\) \(JADE\)](#)) avtorjev Zhang in Sanderson [233, 232],
- samoprilagodljivi [DE](#) s preiskovanjem okolice avtorjev Yang, He in Yao [197],
- [DE](#) z normalno porazdeljenim samoprilagajanjem krmilnih parametrov *F* in *CR* po vzoru [ES \(samoprilagodljiva diferencialna evolucija \(angl. *Self-Adaptive Differential Evolution*\) \(SA-DE\)\)](#) avtorjev Brest, Zamuda, Boškovič in Žumer [29], in ta [DE](#) s koevolucijo ([diferencialna evolucija s samoprilagajanjem in kooperativno koevolucijo \(angl. *Differential Evolution with Self-adaptation and Cooperative Co-evolution*\) \(DEwSAcc\)](#)) avtorjev Zamuda, Brest, Boškovič in Žumer [212],
- memetski algoritem [DE](#) avtorjev Caponio, Neri in Tirronen [35],
- algoritem [DE](#) s podpopulacijami z indeksnim sosedstvom avtorjev Das, Abraham, Chakraborty in Konar [44],

- algoritem DE z ansambli strategij avtorjev Yu in Suganthan [201] in ponovno Mallipeddi, Suganthan, Pan ter Tasgetiren [123],
- algoritem DE za manjše kompaktne sisteme avtorjev Mininno, Neri, Cupertino in Naso [133],
- ter študije posebej parametriziranih DE, avtorjev D. Zaharie [202], J. Tvrđík [183] in zbrani nabori mutacijskih strategij DE avtorjev Price in Storn [151], knjiga V. Feoktistov [63], knjiga D. B. Fogel [64], pregledih področja avtorjev Montes s sod. [130, 129], Neri in Tirronen [138] in predvsem Das in Suganthan [45].

Pri algoritmu DE je bila potrjena tudi potreba po samoprilagajanju krmilnih parametrov [117, 120, 146] in uporabljena v [154, 175] in kasneje npr. [21, 24, 20, 25, 19]. Samoprilagajanje krmilnih parametrov v evolucionih algoritmih v splošnem izhaja iz evolucionih strategij [13, 7, 165], kjer so krmilni parametri vključeni v posameznike skupaj z iskalnimi parametri. Notacija splošne evolucionih 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 $\mathbf{p} = (\mathbf{x}, \mathbf{s}, F(\mathbf{x}))$, kjer so \mathbf{x} iskalni parametri, \mathbf{s} krmilni parametri in $F(\mathbf{x})$ ocenitev posameznika. Čeprav obstajajo druge notacije in variante ES [80, 79, 94], omenimo zaradi umestitve DE zaenkrat zgolj osnovno varianto ES. Ta uporablja logaritmčno normalno porazdeljeno [118] prilagajanje krmilnih parametrov, ki se skozi selekcijo samoprilagajajo. Preživijo le posamezniki, 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 [59, 58].

Poznamo več različic algoritma DE s samoprilagajanjem krmilnih parametrov, izmed katerih natančneje opišimo algoritem jDE [21], ki ga prikazuje postopek 2. Algoritem jDE vključuje mehanizem samoprilagajanja krmilnih parametrov F in CR , ki so ga uvedli Brest s sod. [21]. Slednji mehanizem uporablja zgoraj predstavljeno strategijo rand/1/bin. Mehanizem iz [21] samoprilagaja krmilna parametra F in CR med evolucionim procesom; NP ostaja nespremenjen. Vsak posameznik v populaciji je bil razširjen z vrednostmi teh dveh samoprilagodljivih krmilnih parametrov. Njuni vrednosti se spreminjata:

$$F_{i,g+1} = \begin{cases} F_1 + rand_1 \times F_u & \text{če } rand_2 < \tau_1, \\ F_{i,g} & \text{sicer,} \end{cases} \quad (2.9)$$

Postopek 2 Algoritem jDE [21] s strategijo rand/1/bin.

Vhod: $f(\mathbf{x})$ — kriterijska funkcija; D , NP , G — krmilni parametri DE.

Izhod: \mathbf{x}_{best} — optimirani parametri za dano kriterijsko funkcijo.

1: Uniformno naključno generiraj začetno populacijo DE $\mathbf{x}_{i,0}$ za $i = 1..NP$;

2: **for** DE generacijska zanka g (dokler $g < G$) **do**

3: **for** DE iteracijska zanka i (za vse vektorje $\mathbf{x}_{i,g}$ v trenutni populaciji) **do**

4: DE izračun poskusnega vektorja $\mathbf{x}_{i,g}$ (samoprilagajanje, mutacija, križanje):

$$5: \quad F_{i,g+1} = \begin{cases} F_1 + rand_1 \times F_u & \text{če } rand_2 < \tau_1, \\ F_{i,g} & \text{sicer} \end{cases};$$

$$6: \quad CR_{i,g+1} = \begin{cases} rand_3 & \text{če } rand_4 < \tau_2, \\ CR_{i,g} & \text{sicer} \end{cases};$$

$$7: \quad \mathbf{v}_{i,g+1} = \mathbf{x}_{r_1,g} + F_{i,g+1}(\mathbf{x}_{r_2,g} - \mathbf{x}_{r_3,g});$$

$$8: \quad u_{i,j,G+1} = \begin{cases} v_{i,j,g+1} & \text{če } rand(0,1) \leq CR_{i,g+1} \text{ ali } j = j_{\text{rand}}, \\ x_{i,j,g} & \text{sicer} \end{cases};$$

9: DE selekcija z oceno ustreznosti $f(\mathbf{u}_{i,g+1})$:

$$10: \quad \mathbf{x}_{i,G+1} = \begin{cases} \mathbf{u}_{i,g+1} & \text{če } f(\mathbf{u}_{i,g+1}) < f(\mathbf{x}_{i,g}), \\ \mathbf{x}_{i,g} & \text{sicer} \end{cases};$$

11: **end for**

12: **end for**

13: **vrni** najboljši najden posameznik in ga poimenuj \mathbf{x}_{best} ;

$$CR_{i,g+1} = \begin{cases} rand_3 & \text{če } rand_4 < \tau_2, \\ CR_{i,g} & \text{sicer.} \end{cases} \quad (2.10)$$

Novi vrednosti za F in CR sta shranjeni v nov vektor. $rand_j, j \in \{1, 2, 3, 4\}$ so uniformno naključna števila $\in [0, 1]$. τ_1 in τ_2 predstavlja verjetnost prilagajanja obeh krmilnih parametrov F in CR . τ_1, τ_2, F_1, F_u imajo nespremenljive vrednosti 0,1, 0,1, 0,1, 0,9. Nov F zavzema naključne vrednosti med $[0,1, 1,0]$ in nov CR med $[0, 1]$. $F_{i,g+1}$ in $CR_{i,g+1}$ sta izračunana pred mutacijo, po vzoru iz izkušenj ob razvoju ES. Nova parametra tako vplivata na mutacijo, križanje in selekcijo novega vektorja $\mathbf{x}_{i,g+1}$.

Ena od razširitev algoritma jDE je tudi obravnava omejitev, algoritem ϵ -jDE [18], ki sloni na osnovnem algoritmu jDE [21] in prvem dopolnjenem

algoritmu za obravnavo omejitev z jDE, jDE-2 [24]. Algoritem ϵ -jDE omejitve obravnava v selekciji, tako da prilagojeno izbira med vektorjema z indeksi i in j , glede na dopustnost \bar{v} in ovrednotenje f :

$$\mathbf{x}_{i,g+1} = \begin{cases} \mathbf{x}_{j,g} & \text{če } (\bar{v}_{i,g} > \bar{v}_{j,g}), \\ \mathbf{x}_{j,g} & \text{sicer če } (\bar{v}_{j,g} = 0) \wedge (f(\mathbf{x}_{j,g}) < f(\mathbf{x}_{i,g})), \\ \mathbf{x}_{i,g} & \text{sicer.} \end{cases} \quad (2.11)$$

Algoritem razlikuje med dopustnimi ($\bar{v} = 0$) in nedopustnimi vektorji: vsak dopustni vektor je boljši od vsakega nedopustnega. Takahama in Sakai v [172] sta izpostavila, da je za izboljšanje optimizacije problemov z enakostnimi omejitvami koristno nadzorovati interval ϵ , tako da prilagajamo njegov nivo vpliva. Algoritem ϵ -jDE uporablja metodo krmiljenje nivoja ϵ , tako da vektorje s kršitvami omejitev do nivoja ϵ obravnavamo enakovredno dopustnim vektorjem in primerjamo le njihove kriterijske ocene. Nivo ϵ prilagajamo, dokler število generacij g ne doseže nadzorne generacije G_c . Ko število generacij preseže G_c , nivo ϵ postavimo na 0, da najdemo le vektorje z najmanjšo še dovoljeno kršitvijo omejitev. Metoda je definirana kot:

$$\epsilon_0 = \epsilon \quad (2.12)$$

$$v_0 = \bar{v}(\mathbf{x}_\theta) \quad (2.13)$$

$$\bar{v}_g = \begin{cases} \alpha_1 v_{g-1}, & \alpha_2 \bar{v}(\mathbf{x}_\beta) < v_{g-1}, \quad 0 < g < G_c \\ v_{g-1}, & \text{sicer} \end{cases} \quad (2.14)$$

$$\epsilon_G = \begin{cases} \max\{v_g(1 - \frac{g}{G_c})^{c_p}, \epsilon\}, & 0 < g < G_c \\ 0, & \text{sicer} \end{cases}, \quad (2.15)$$

kjer predstavlja \mathbf{x}_θ najboljših θ vektorjev in $\theta = 0,3NP$. \mathbf{x}_β označuje najboljših β vektorjev in $\beta = 0,7NP$. Pri izračunu $\bar{v}(\mathbf{x}_\theta)$ privzamemo $\epsilon(0) = 0$. c_p je krmilni parameter hitrosti rahljanja dopustnosti omejitev in $\alpha_1 < 1$ ter $\alpha_2 > 1$ krmilita prilagajanje vrednosti v_g , ki prav tako krmili hitrost rahljanja dopustnosti omejitev. Parametra α_1 in α_2 lahko le zmanjšata vrednost v_g za majhno vrednost, ko je povprečna kršitev omejitev $\bar{v}(\mathbf{x}_\beta)$ za najboljših β vektorjev, pomnožena z α_1 , manjša od v_g . S tem prilagajanjem lahko nivo ϵ doseže vrednost 0 že pred $g \geq G_c$.

Algoritem jDE pozna še številne druge razširitve evlucijskih operatorjev, kot so dela [20, 18, 29, 23, 26, 19, 27, 31, 22, 30]. Algoritem jDE [21, 27] smo nadgradili tudi z večkriterijskim optimiranjem [28, 211], kar je obširneje razloženo v nadaljevanju. DE smo s soavtorji pomagali testirati tudi na dinamičnih problemih [105], kjer je algoritem zmagal na svetovnem tekmovanju.

Za ta algoritem smo opravili še primerjavo z algoritmom DASA v revijalni objavi [22].

Zaradi svoje uspešnosti je algoritem DE bil že večkrat uporabljen tudi za večkriterijsko optimizacijo [1, 122, 5, 6, 89, 33, 161, 181, 210, 163, 173, 2, 237, 214, 213, 110]. Na začetku raziskav na področju večkriterijske optimizacije so bile v uporabi klasične enokriterijske optimizacijske metode (npr. Newtonova metoda), pri katerih večkriterijski problem z utežno funkcijo pretvorimo v enokriterijski problem. Prvi poskus reševanja večkriterijskega problema z iskanjem več nedominiranih rešitev je bil predstavljen v [164]. Od takrat naprej [39, 40] se je nato pri večkriterijski optimizaciji precej povečal interes za uporabo populacijskih naključnih optimizacijskih algoritmov [50, 51, 241, 1, 242, 89, 161, 160, 54], kot so evolucijski algoritmi, simulirano ohlajanje [101], iskanje s tabuji [72] in drugi že omenjeni. S temi želimo najti čim več in čim ustreznejše porazdeljenih Pareto optimalnih rešitev. 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. Pristop izdelave populacijskih vektorjev z diferencialno evolucijo so izbrali tudi že algoritmi za večkriterijsko optimizacijo, izmed katerih so prvi znani algoritmi [1, 5, 89, 161, 109]. Omenimo nekaj primerkov posameznih evolucijskih algoritmov:

NSGA-II, genetski algoritem z elitističnim nedominiranim urejanjem (angl. *Elitist Non-Dominated Sorting Genetic Algorithm*) je algoritem Deba in sodelavcev iz leta 2000 [52] in je zelo uspešen genetski algoritem (algoritem je nadgradnja predhodnega algoritma [169]), ki se je v svojem času odrezal najbolje od vseh (takrat obstoječih) evolucijskih algoritmov za večkriterijsko optimizacijo. Algoritem 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, t.j. čim bolj oddaljen od rešitev v isti fronti,

SPEA2, močnostni Pareto evolucijski algoritem (angl. *Strength Pareto Evolutionary Algorithm*) je algoritem Zitzlerja in sod. iz leta 2001 [243, 241] in je prav tako popularen genetski algoritem za večkriterijsko optimizacijo, ki glede na **NSGA-II** najde enakomernejše porazdeljene (manj nakopičene) rešitve. Vrednotenje posameznikov ne poteka po frontah, pač pa enovito z merjenjem **moči** (*strength*), **grobe uspešnosti** (*raw fitness*) in **gostote** (*density*) posameznikov. Moč posameznika je enaka številu posameznikov iz populacije, ki jih izbrani posameznik dominira.

Kot zanimivost povejmo, da Zitzler in sod. označujejo relacijo dominiranosti v obratni smeri, s simbolom \succ . Grobo spešnost posameznika izračunamo kot vsoto moči vseh posameznikov iz populacije, ki ta posameznik dominirajo. Ker z evolucijo številni posamezniki postanejo nedominirani (in imajo zato grobo uspešnost enako nič), potrebujemo dodatno informacijo o nakopičenosti rešitev. Zato za vse posameznike z enako grobo uspešnostjo izračunamo še gostoto. Algoritem za njen izračun je prilagojena različica algoritma k -tega najbližjega soseda [167]. Metriko nakopičenosti posameznika dobimo tako, da seštejemo grobo uspešnost in gostoto,

PDE, diferencialna evolucija s pareto fronto (angl. *Pareto-frontier Differential Evolution*) kot prvi predstavnik algoritmov diferencialne evolucije za večkriterijsko optimizacijo [1, 122], kjer je v algoritmu dopolnjen selekcijski operator, tako da upošteva večkriterijsko selekcijo,

POGA, genetski algoritem s preferenčnim urejanjem (angl. *Preference Ordering Genetic Algorithm*) [54] je 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 dva posameznika neprimerljiva oz. nedominantna, se ohrani posameznik, ki po čim več kriterijih dominira druge,

MTS, večtrajektorijsko iskanje (angl. *Multiple Trajectory Search*) [178], ki z več trajektorijami neodvisno preiskuje kriterijski prostor,

MOPSO, večkriterijska optimizacija z rojem delcev (angl. *Multiobjective Particle Swarm Optimization*) [240], ki je razširitev algoritma optimizacije z rojem delcev na večkriterijsko optimizacijo,

MOEA/D, večkriterijski evolucijski algoritem z razstavitvijo (angl. *Multiobjective Evolutionary Algorithm based on Decomposition*) [234], ki razstavi večkriterijsko funkcijo in dosega dobre rezultate, vidne na primerjavi rezultatov iz tekmovanja CEC 2009 [235],

DEMO, diferencialna evolucija za večkriterijsko optimizacijo (angl. *Differential Evolution for Multi-objective Optimization*), ki je ena prvih razširitev algoritma DE na večkriterijsko optimizacijo [161],

GDE, posplošena diferencialna evolucija (angl. *Generalized Differential Evolution*) [110], ki je podoben algoritmu DEMO, a ima drugače nastavljene nespremenljive vrednosti krmilnih parametrov,

MO-SADE, večkriterijska optimizacija s samoprilagodljivo diferencialno evolucijo (angl. *Multi-objective Optimization based on Self-adaptive Differential Evolution*) [88], ki je razširitev enokriterijskega algoritma SADE [153] in daje povprečne rezultate [86],

MO-SADE, DEMOWSA, DECMOSA-SQP, to so algoritmi: večkriterijska optimizacija s samoprilagodljivo diferencialno evolucijo (angl. *Multi-objective Optimization based on Self-adaptive Differential Evolution*) [211], diferencialna evolucija za večkriterijsko optimizacijo s samoprilagajanjem (angl. *Differential Evolution for Multi-objective Optimization with Self-adaptation*) [210] in diferencialna evolucija s samoprilagajanjem in lokalnim iskanjem za večkriterijsko optimizacijo z omejitvami (angl. *Differential evolution with self-adaptation and local search for constrained multiobjective optimization*) [214], ki so opisani v nadaljevanju in

tisoče drugih večkriterijskih algoritmov za optimizacijo in njihovih izpeljank, ki so po večini bile predstavljene v revijah, kot so:

- revija IEEE TEVC (angl. *IEEE Transactions on Evolutionary Computation*),
- revija IEEE CYB (angl. *IEEE Transactions on Cybernetics*),
- revija SWEVO (angl. *Swarm and Evolutionary Computaton*),
- revija INS (angl. *Information Sciences*),
- revija ASOC (angl. *Applied Soft Computing*),
- revija ECJ (angl. *Evolutionary Computation*) in
- revija SOCO (angl. *Soft Computing*)

ter na znanstvenih konferencah, kot so: konferenca CEC (angl. *IEEE Congress on Evolutionary Computation*), konferenca WCCI (angl. *IEEE World Congress on Computational Intelligence*), konferenca GECCO (angl. *Genetic and Evolutionary Computation Conference*) in konferenca SEMCCO & FANCCO (angl. *Swarm, Evolutionary and Memetic Computing (SEMCCO) & Fuzzy And Neural Computing (FANCCO)*). Za enega od novejših pregledov večkriterijskih algoritmov za optimizacijo z literaturo v obsegu 310 člankov usmerimo bralca na aktualno objavo avtorjev Zhou s sod. [236] v reviji SWEVO.

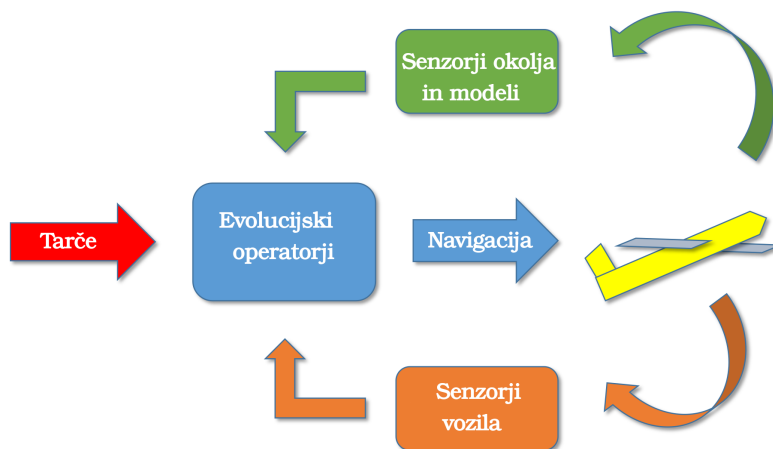
Izboljšave diferencialne evolucije zajemajo tudi robustnost pri velikih dimenzijah iskalnega prostora [198, 212]. Križanje algoritma DE in evolucijskega programiranja [199] zasledimo v delu [197]. Algoritmu DE je

soroden tudi algoritem optimizacije s kolonijami mravelj [105], zato smo izvedli tudi primerjalno študijo optimizacijskih mehanizmov in kakovosti optimiranja med algoritmoma [22]. Diferencialna evolucija je bila uporabljena tudi za ugaševanje parametrov iger s popolno informacijo in ničelno vsoto [16, 15]. Algoritem DE je prav tako rotacijsko invarianten v prostoru spremenljivk [151]. Posledica tega je, da lahko dobro rešuje probleme z neločljivimi iskalnimi parametri. Takšno lastnost izkazuje precej realnih oz. industrijskih problemov [151, 63, 97, 177, 182, 239, 124, 190, 137]. Nad nekaterimi realnimi problemi iz dejanskega sveta smo preskusili tudi algoritem jDE [216, 205]. Tudi uporabnost večkriterijskih optimizacijskih algoritmov je bila prikazana že v različnih domenah [50]. Evolucijski algoritmi so se splošneje pokazali kot zelo uporabni pri reševanju kompleksnih večkriterijskih optimizacijskih problemov, vključno s številnimi problemi iz dejanskega sveta [216, 205]. Nekateri novejši evolucijski algoritmi za večkriterijsko optimizacijo rešujejo tudi probleme šuma v vhodnih podatkih [73]. Laumanns in sodelavci so enokriterijsko optimizacijo z omejitvami reševali z večkriterijsko optimizacijo [114]. Tan in sodelavci [173] so v večkriterijskem optimizacijskem algoritmu uporabili koevolucijo [37] in optimizacijo izvajali paralelno. Primer uporabe diferencialne evolucije za večkriterijsko optimizacijo realnih problemov vključuje tudi podatkovno rudarjenje [2]. Primeri večkriterijskih optimizacijskih problemov iz dejanskega sveta vključujejo še optimizacijo stacionarne plinske gorilne turbine [33], oblikovanje drobilca kamnov [9], problem distribucije naftnih derivatov [47], upravljanje z jedrskim gorivom [61], planiranje [166], načrtovanje telekomunikacijskih omrežij [189], obrambne aplikacije [90], načrtovanje elektromotorjev [182], rekonstrukcije prostorskih proceduralnih modelov naravnih dreves [208], optimizacija razporejanja proizvodnje električne energije med hidroelektrarnami in termoelektrarnami [71] in planiranje avtonomne vožnje [231]. Nekateri od teh pristopov bodo obravnavani tudi v naslednjih poglavjih.

Poglavje 3

Načrtovanje algoritmov operacijskih raziskav logističnih, transportnih in ekonomskih sistemov

Primer predstavljenega načrta temeljnega operacijskega algoritma (za optimizacijo) iz prejšnjega poglavja je v tem poglavju vstavljen kot aplikacija pri operacijskih raziskavah logističnih, transportnih in ekonomskih sistemov. S tem dobimo aplikacijo takega temeljnega algoritma. Primer te aplikacije je razložen v nadaljevanju skozi logistični sistem za [planiranje poti podmorskih sond](#) (angl. *Underwater Glider Path Planning*) (UGPP) — slika 3.1.

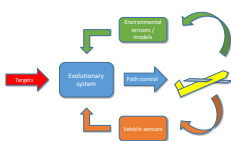


Slika 3.1: Arhitektura načrtovanega sistema algoritmov za operacijske raziskave logističnih, transportnih in ekonomskih sistemov — na primeru načrtovanja navigacije podvodnih sond.

Zbrano gradivo v tem poglavju, ki je zajeto in prevedeno (prevedeni so tudi nekateri drugi vključeni psevdokodi) iz revijalnega prispevka na koncu prikazuje postopek 6. Najprej pa so predstavljeni primeri osnovnega dela tega pristopa, ki predstavljajo optimizacijske algoritme tipa DE (postopki 3, 4 in 5).

Primer algoritma za operacijske raziskave: L-SHADE

V tej sekciji je podan primer novejšega temeljnega algoritma, SHADE z linearnim zmanjševanjem velikosti populacije (angl. *Linear population size reduction – SHADE*) (L-SHADE) [174]. Za osnovo tega algoritma je uporabljen prilagodljivi algoritem diferencialne evolucije, temelječ na zgodovini uspehov (angl. *Success-History Based Adaptive Differential Evolution Algorithm*) (SHADE), ki vključuje še mehanizem za linearno zmanjševanje velikosti populacije (angl. *Linear Population Size Reduction*) (LPSR). Zbran in v slovenščino preveden psevdokod algoritma prikazuje postopek 3. Algoritem v vrsticah od 5 do 13 izračuna poskusni vektor na podlagi arhiva in trenutne populacije, nato pa v vrstici 18 oz. 19 še osveži arhiv oz. krmilne parametre za generiranje poskusnih vektorjev. Krmilni parametri imajo svoj manjši pomnilniški vmesnik, ki je osvežen v vrstici 22, zmanjševanje populacije LPSR in arhiv pa sta obravnavana v vrsticah od 23 do 26. Psevdokod postopka (algoritma) je zbran in preveden iz nedavnega prispevka, ki obravnava tematike logističnih, transportnih in ekonomskih sistemov in uporablja L-SHADE:



Aleš Zamuda and José Daniel Hernández Sosa. **Success history applied to expert system for underwater glider path planning using differential evolution.** *Expert Systems with Applications*, 2019, vol. 119, pp. 155–170. <http://dx.doi.org/10.1016/j.eswa.2018.10.048>

Postopek 3 Optimizacijski algoritem L-SHADE.

Vhod: Spremenljivi parametri — MAX_FES (največje število dodeljenih ovrednotenj kriterijske funkcije), konstante — p in $|A|$ (konstante L-SHADE za JADE [232] parametre: pohlepnost mutacije $p = 0.11$, velikost arhiva $|A|$ enaka NP), $NP_{\min} = 4$ in $NP_{\text{init}} = 5D = 60$ (konstante L-SHADE za največjo in najmanjšo velikost populacije DE pri mehanizmu spreminjanja velikosti populacije, LPSR), H (konstante L-SHADE za nastavljanje parametra za velikost pomnilnika hrambe zgodovinsko uspešnih krmilnih parametrov F_i in CR_i).

Izhod: \mathbf{x} – iskani parametri (velikosti D).

- 1: uniformno naključno generiramo začetno populacijo **DE** $\mathbf{x}_{i,0}$,
 $\forall i \in \{1, 2, \dots, NP_{\text{init}}\}$; $NP_0 = NP_{\text{init}}$;
- 2: inicializiramo shrambe **L-SHADE**: $A = \emptyset$, $\forall k \in \{1, 2, \dots, H\}$: $M_{\text{CR},k} = M_{\text{F},k} = 0,5$;
- 3: **for** Generacijska zanka **DE** $g = 1$ do G , $NFE < MAX_FES$ **do**
- 4: **for** Iteracijska zanka **DE**, za i (vsi posamezniki $\mathbf{x}_{i,g}$ v populaciji) **do**
- 5: Izračun novega poskusnega vektorja **DE**, $\mathbf{x}_{i,g}$:
 - 6: uniformno naključno generiramo $r_i \in [1, H]$;
 - 7: $F_i = \text{randc}_i(M_{\text{F},r_i}, 0,1)$;
 - 8: $CR_i = \text{randn}_i(M_{\text{CR},r_i}, 0,1)$;
 - 9: $\mathbf{P} = \{\forall k \in [1, NP_g] : \mathbf{x}_{k,g}\} \cup \mathbf{A}$;
 - 10: $p_{\text{best}} = \lfloor p|P| + 0,5 \rfloor$;
 - 11: izberemo $\mathbf{x}_{p_{\text{best}},g}$ pri p_{best} iz \mathbf{P} , rangirani po ustreznosti;
 - 12: $\mathbf{v}_{i,g+1} = \mathbf{x}_{i,g} + F_i(\mathbf{x}_{p_{\text{best}},g} - \mathbf{x}_{i,g}) + F_i(\mathbf{x}_{r_1,g} - \mathbf{x}_{r_2,g})$;
 - 13: $\forall j \in \{1, \dots, D\}$: $u_{i,j,g+1} = \begin{cases} v_{i,j,g+1} & \text{če } \text{rand}(0,1) \leq CR_{i,g+1} \\ & \text{ali } j = j_{\text{rand}} \\ x_{i,j,g} & \text{sicer} \end{cases}$;
- 14: Ovrednotenje ustreznosti v **DE** (kličemo kriterijsko funkcijo):
- 15: $f(\mathbf{u}_{i,g}) = \text{ovrednoti_kriterijsko_funkcijo}(u_{i,1,g+1})$;
- 16: $\mathbf{x}_{i,g+1} = \begin{cases} \mathbf{u}_{i,g+1} & \text{če } f(\mathbf{u}_{i,g+1}) \leq f(\mathbf{x}_{i,g}) \\ \mathbf{x}_{i,g} & \text{sicer} \end{cases}$;
- 17: **if** $f(\mathbf{u}_{i,g+1}) < f(\mathbf{x}_{i,g})$:
 - 18: zamenjaj najslabšega posameznika v \mathbf{A} z $\mathbf{u}_{i,g+1}$ (ali ga dodaj, ko $|\mathbf{A}| < NP_g$);
 - 19: shrani propagirana F_i in CR_i za posodobitev krmilnih pomnilnikov \mathbf{M}_{F} , \mathbf{M}_{CR} ;
- 20: **end if**
- 21: **end for**
- 22: posodobi pomnilnika \mathbf{M}_{F} in \mathbf{M}_{CR} s propagiranimi parametri;
- 23: $NP_{g+1} = NP_{\text{min}} + \left\lfloor \frac{NFE}{MAX_FES} (NP_{\text{init}} - NP_{\text{min}}) + 0,5 \right\rfloor$; // preverimo LPSR
- 24: **if** $NP_{g+1} < NP_g$ **then**
- 25: za populacijo in arhiv: zmanjšaj velikosti obeh na NP_{g+1} z odstranitvijo najslabših posameznikov;
- 26: **end if**
- 27: **end for**
- 28: **vrni** najboljši posameznik, dobljen izmed $\mathbf{x}_{i,G}$;

Primer algoritma za operacijske raziskave: DISH

Ta sekcija podaja primer za načrt algoritma, ki je lahko temelj pri logističnih, transportnih in ekonomskih sistemih. Obravnava algoritem za [na razdalji temelječe prilagajanje krmilnih parametrov za diferencialno evolucijo](#), temelječo na zgodovinskem uspehu (angl. *DIstance Based Parameter Adaptation for Succes-History based Differential Evolution*) (DISH). Sledi zbrano gradivo v tem poglavju, zajeto iz revijalnega prispevka — postopek 4.

```
if  $S_{i,j} \neq 0$  and  $S_{i,j} \neq 0$  then
  Update  $M_{i,j}(0)$  and  $M_{i,k}$ 
  (10) with Lehmer mean computed
  by (11) with distance based weights
  from (11)  $k++$ ;
  if  $k > H$  then
     $k = 1$ 
  end if
   $\sigma_i = \frac{\sqrt{\sum_{j=1}^D (m_{i,j} - n_{i,j})^2}}{\sqrt{\sum_{j=1}^D (m_{i,j} - n_{i,j})}}$ 
end if
```

DISH A. Viktorin, R. Senkerik, M. Pluhacek, T. Kadavy, A. Zamuda. **Distance Based Parameter Adaptation for Success-History based Differential Evolution**. Swarm and Evolutionary Computation, Available online 12 November 2018. <http://dx.doi.org/10.1016/j.swevo.2018.10.013>

Prevod psevdokoda algoritma v slovenščino iz zgornjega članka prikazuje postopek 4. Izvoro kodo algoritma je objavil Ponnuthurai Nagarathnam Suganthan v GitHubu na predlog in potrditev Kennetha Priceja v oktobru 2019 po zaključku tekmovanja CEC 2019 in se nahaja na naslovu <https://github.com/P-N-Suganthan/CEC2019/blob/master/PaperCodes-DISHchain.zip>.

Koda vsebuje programski paket, uporabljen tako za izračune algoritma DISH v SWEVO, kot tudi postopek 5, predstavljen kasneje v tem zbranem gradivu. Z modro barvo je v tem postopku v vrstici 54 obarvan tisti del, ki v ta postopek dodaja drugačen način računanja krmilnih parametrov (na podlagi razdalje).

Postopek 4 Optimizacijski algoritem DISH.

Vhod: Spremenljivi parametri — MAX_FES (največje število dodeljenih ovrednotenj kriterijske funkcije), konstante — p in $|A|$ (konstante L-SHADE za JADE [232] parametre: pohlepnost mutacije p , velikost arhiva $|A|$), NP_{\min} in NP_{init} (konstante L-SHADE za največjo in najmanjšo velikost populacije DE pri mehanizmu spreminjanja velikosti populacije, LPSR), H (konstante L-SHADE za nastavljanje parametra za velikost pomnilnika hrambe zgodovinsko uspešnih krmilnih parametrov F_i in CR_i).

Izhod: \mathbf{x}_{best} — iskani parametri (velikosti D).


```

1: Nastavi  $NP_{\text{init}}$ ,  $NP_f$ ,  $H$  in zaustavitveni pogoj;
2:  $NP = NP_{\text{init}}$ ,  $G = 0$ ,  $\mathbf{x}_{\text{best}} = \{\}$ ,  $k = 1$ ,  $p_{\text{min}} = 2/NP$ ,  $\mathbf{A} = \emptyset$ ;
3: Naključno inicializiraj populacijo  $\mathbf{P} = (\mathbf{x}_{1,G}, \dots, \mathbf{x}_{NP,G})$ ;
4: Nastavi vse vrednosti v  $\mathbf{M}_F$  na 0,5 in  $\mathbf{M}_{CR}$  na 0,8;
5:  $\mathbf{P}_{\text{new}} = \{\}$ ,  $\mathbf{x}_{\text{best}} =$  najboljši iz populacije  $\mathbf{P}$ ;
6: while zaključitveni pogoj ni zadovoljen (upoštevaj  $MAX\_FES$ ) do
7:    $\mathbf{S}_F = \emptyset$ ,  $\mathbf{S}_{CR} = \emptyset$ ;
8:   for  $i = 1$  to  $NP$  do
9:      $r = \mathcal{U}[1, H]$ ;
10:    if  $r = H$  then
11:       $M_{F,r} = 0,9$ ;
12:       $M_{CR,r} = 0,9$ ;
13:    end if
14:    if  $M_{CR,r} < 0$  then
15:       $CR_{i,G} = 0$ ;
16:    else
17:       $CR_{i,G} = \mathcal{N}(M_{CR,r}, 0,1)$ ;
18:    end if
19:    Nastavi  $F_i$  z  $F_i = \mathcal{C}[M_{F,r}, 0,1]$ ;
20:    if  $G < 0,6G_{\text{MAX}}$  in  $F_{i,G} > 0,7$  then
21:       $F_{i,G} = 0,7$ ;
22:    end if
23:    if  $G < 0,25G_{\text{MAX}}$  then
24:       $CR_{i,G} = \max(CR_{i,G}, 0,7)$ ;
25:    else if  $G < 0,5G_{\text{MAX}}$  then
26:       $CR_{i,G} = \max(CR_{i,G}, 0,6)$ ;
27:    end if
28:     $\mathbf{x}_{i,G} = \mathbf{P}[i]$ ,  $p_i = U[p_{\text{min}}, 0,2]$ ;
29:     $\mathbf{v}_{i,G}$  z mutacijo:  $\mathbf{v}_i = \mathbf{x}_i + F_w(x_{p_{\text{Best}}} - \mathbf{x}_i) + F(\mathbf{x}_{r1} - \mathbf{x}_{r2})$ 
    kjer je  $F_w$  izračunan kot :  $F_w = \begin{cases} 0,7F, FES < 0,2MAX\_FES, \\ 0,8F, FES < 0,4MAX\_FES, \\ 1,2F, \text{sicer.} \end{cases}$  ;
30:     $\mathbf{u}_{i,G}$  s križanjem:  $u_{j,i} = \begin{cases} v_{j,i} & \text{če } \mathcal{U}[0, 1] \leq CR_i \text{ ali } j = j_{\text{rand}} \\ x_{j,i} & \text{sicer} \end{cases}$  ;
31:    if  $f(\mathbf{u}_{i,G}) \leq f(\mathbf{x}_{i,G})$  then
32:       $\mathbf{x}_{i,G+1} = \mathbf{u}_{i,G}$ ;
33:    else
34:       $\mathbf{x}_{i,G+1} = \mathbf{x}_{i,G}$ ;
35:    end if
36:    if  $f(\mathbf{u}_{i,G}) < f(\mathbf{x}_{i,G})$  then

```

```

37:      $\mathbf{x}_{i,G} \rightarrow \mathbf{A}$ ; ( $\rightarrow$  označuje, da potisnemo  $\mathbf{x}_{i,G}$  v arhiv  $\mathbf{A}$ , na konec)
38:      $F_i \rightarrow \mathbf{S}_F, CR_i \rightarrow \mathbf{S}_{CR}$ ;
39: end if
40: if  $|\mathbf{A}| > NP$  then
41:     Naključno odstrani  $|\mathbf{A}| - NP$  posameznikov iz  $\mathbf{A}$ ;
42: end if
43:      $\mathbf{x}_{i,G+1} \rightarrow \mathbf{P}_{new}$ ;
44: end for
45: Izračunaj  $NP_{new}$  glede na:
46:  $NP_{new} = \text{round} \left( NP_{init} - \frac{FES}{MAXFES} * (NP_{init} - NP_f) \right)$ ;
47: if  $NP_{new} < NP$  then
48:     Uredi posameznike v  $\mathbf{P}$  glede na njihove vrednosti kriterijske funkcije
49:     in odstrani  $NP - NP_{new}$  najslabših;
50:      $NP = NP_{new}$ ;
51: end if
52: if  $|\mathbf{A}| > NP$  then
53:     Naključno odstrani  $|\mathbf{A}| - NP$  posameznikov iz  $\mathbf{A}$ ;
54: end if
55: if  $\mathbf{S}_F \neq \emptyset$  in  $\mathbf{S}_{CR} \neq \emptyset$  then
56:     Posodobi  $M_{F,k}$ :  $M_{F,k} = \begin{cases} \text{mean}_{WL}(\mathbf{S}_F) & \text{if } \mathbf{S}_F \neq \emptyset \\ M_{F,k} & \text{sicer} \end{cases}$  in
57:      $M_{CR,k}$ :  $M_{CR,k} = \begin{cases} \text{mean}_{WL}(\mathbf{S}_{CR}) & \text{if } \mathbf{S}_{CR} \neq \emptyset \\ M_{CR,k} & \text{sicer} \end{cases}$ ,
58:     s pomočjo Lehmerjevega povprečja:  $\text{mean}_{WL}(\mathbf{S}) = \frac{\sum_{k=1}^{|\mathbf{S}|} w_k \bullet S_k^2}{\sum_{k=1}^{|\mathbf{S}|} w_k \bullet S_k}$ 
59:     z utežmi, temelječimi na razdalji, iz  $w_k = \frac{\sqrt{\sum_{j=1}^D (u_{k,j,G} - x_{k,j,G})^2}}{\sum_{m=1}^{|\mathbf{S}_{CR}|} \sqrt{\sum_{j=1}^D (u_{m,j,G} - x_{m,j,G})^2}}$ ,
60:      $k++$ ;
61: if  $k > H$  then
62:      $k = 1$ ;
63: end if
64: end if
65:  $\mathbf{P} = \mathbf{P}_{new}$ ,  $\mathbf{P}_{new} = \{\}$ ,  $\mathbf{x}_{best}$  = najboljši iz populacije  $\mathbf{P}$ ,  $G++$ ;
66: end while
67: vrni  $\mathbf{x}_{best}$  kot najboljšo najdeno rešitev;

```

Primer razširitve algoritma za operacijske raziskave: DISHv

Ta sekcija podaja primer za načrt algoritma, ki je lahko temelj pri logističnih, transportnih in ekonomskih sistemih. Obravnava algoritem DISHv [227] (DISH [185] s prenosno funkcijo V4 [134]).

Članek DISHv [227] vsebuje tudi kodo za modifikacijo izvora kode algoritma DISH, za katerega je izvor kodo algoritma objavil Ponnuthurai Nagarathnam Suganthan v GitHubu na predlog in potrditev Kennetha Priceja v oktobru 2019 po zaključku tekmovanj CEC 2019; koda za DISH se nahaja na naslovu <https://github.com/P-N-Suganthan/CEC2019/blob/master/PaperCodes-DISHchain.zip>. Koda vsebuje programski paket, uporabljen tako za izračune algoritma DISH v SWEVO, kot tudi postopek 5. Sledi zbrano gradivo v tem poglavju, ki je zajeto iz konferenčnega prispevka:



A. Zamuda. **Binary 100-digit challenge using IEEE-754 coded numerical optimization scenarios (100b-digit) and V-shape binary distance-based success history differential evolution (DISHv)**. Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO 2019, Prague, Czech Republic, July 13-17, 2019: 1821–1828. <http://dx.doi.org/10.1145/3319619.3326898>

Prevod psevdokoda algoritma v slovenščino iz zgornjega članka prikazuje postopek 5, podčrtan in z modro barvo je v tem postopku v vrsticah 4, 5, 33, 34, 35 in 40 označen tisti del, ki je spremenjen glede na osnovni algoritem DISH.

Postopek 5 Optimizacijski algoritem DISHv.

Vhod: Spremenljivi parametri — MAX_FES (največje število dodeljenih ovrednotenj kriterijske funkcije), konstante — p in $|A|$ (konstante L-SHADE za JADE [232] parametre: pohlepnost mutacije p , velikost arhiva $|A|$), NP_{min} in NP_{init} (konstante L-SHADE za največjo in najmanjšo velikost populacije DE pri mehanizmu spreminjanja velikosti populacije, LPSR), H (konstante L-SHADE za nastavljanje parametra za velikost pomnilnika hrambe zgodovinsko uspešnih krmilnih parametrov F_i in CR_i).

Izhod: \mathbf{x}_{best} — iskani binarizirani parametri (velikosti D).

- 1: Nastavi $NP_{\text{init}}, NP_f, H$ in zaustavitveni pogoj;
- 2: $NP = NP_{\text{init}}, G = 0, \mathbf{x}_{\text{best}} = \{\}, k = 1, p_{\text{min}} = 2/NP, \mathbf{A} = \emptyset$;
- 3: Naključno inicializiraj populacijo $\mathbf{P} = (\mathbf{x}_{1,G}, \dots, \mathbf{x}_{NP,G})$,
kjer $\forall i : \mathbf{x}_{i,G} = \{x_{1,i}, x_{2,i}, \dots, x_{D,i}\}$ v generaciji G ;
- 4: Binariziraj: $\forall_{i,j} : x_{i,j}^B = \mathcal{U}[0, 1] < |(\frac{2}{\pi} \arctan(\frac{\pi}{2} x_{i,j})|$
- 5: Ovrednoti $\forall_i f(\mathbf{x}_{i,G}) := f^B(\mathbf{x}_{i,G}^B)$
- 6: Nastavi vse vrednosti v \mathbf{M}_F na 0,5 in \mathbf{M}_{CR} na 0,8;
- 7: $\mathbf{P}_{\text{new}} = \{\}, \mathbf{x}_{\text{best}} = \text{najboljši iz populacije } \mathbf{P}$;
- 8: **while** zaključni pogoj ni zadovoljen (upoštevaj MAX_FES) **do**
- 9: $\mathbf{S}_F = \emptyset, \mathbf{S}_{CR} = \emptyset$;
- 10: **for** $i = 1$ to NP **do**
- 11: $r = \mathcal{U}[1, H]$;
- 12: **if** $r = H$ **then**
- 13: $M_{F,r} = M_{CR,r} = 0,9$;
- 14: **end if**
- 15: **if** $M_{CR,r} < 0$ **then**
- 16: $CR_{i,G} = 0$;
- 17: **else**
- 18: $CR_{i,G} = \mathcal{N}(M_{CR,r}, 0, 1)$;
- 19: **end if**
- 20: $F_{i,G} = \mathcal{C}[M_{F,r}, 01]$;
- 21: **if** $G < 0,6G_{\text{MAX}}$ **and** $F_{i,G} > 0,7$ **then**
- 22: $F_{i,G} = 0,7$;
- 23: **end if**
- 24: **if** $G < 0,25G_{\text{MAX}}$ **then**
- 25: $CR_{i,G} = \max(CR_{i,G}, 0,7)$;
- 26: **else if** $G < 0,5G_{\text{MAX}}$ **then**
- 27: $CR_{i,G} = \max(CR_{i,G}, 0,6)$;
- 28: **end if**
- 29: $\mathbf{x}_{i,G} = \mathbf{P}[i], p_i = \mathcal{U}[p_{\text{min}}, 0,2]$;
- 30:
$$F_w = \begin{cases} 0,7 \times F, & FES < 0,2MAXFES, \\ 0,8 \times F, & FES < 0,4MAXFES, \\ 1,2 \times F, & \text{sicer.} \end{cases}$$
- 31: $\mathbf{v}_{i,G} = \mathbf{x}_i + F_w(x_{p_{\text{Best}}} - \mathbf{x}_i) + F(\mathbf{x}_{r1} - \mathbf{x}_{r2})$;
- 32:
$$u_{j,i,G} = \begin{cases} v_{j,i} & \text{if } \mathcal{U}[0, 1] \leq CR_i \text{ or } j = j_{\text{rand}} \\ x_{j,i} & \text{sicer} \end{cases} ;$$
- 33: Binariziraj: $\forall_{i,j} : u_{i,j}^B = \mathcal{U}[0, 1] < |(\frac{2}{\pi} \arctan(\frac{\pi}{2} u_{i,j})|$
- 34: Ovrednoti $\forall_i f(\mathbf{u}_{i,G}) := f^B(\mathbf{u}_{i,G}^B)$
- 35: **if** $f^B(\mathbf{u}_{i,G}) \leq f^B(\mathbf{x}_{i,G})$ **then**

```

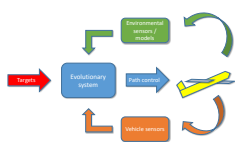
36:      $\mathbf{x}_{i,G+1} = \mathbf{u}_{i,G}$ ;
37: else
38:      $\mathbf{x}_{i,G+1} = \mathbf{x}_{i,G}$ ;
39: end if
40: if  $f^B(\mathbf{u}_{i,G}) < f^B(\mathbf{x}_{i,G})$  then
41:      $\mathbf{x}_{i,G} \rightarrow \mathbf{A}$ ;
42:      $F_i \rightarrow \mathbf{S}_F, CR_i \rightarrow \mathbf{S}_{CR}$ ;
43: end if
44: if  $|\mathbf{A}| > NP$  then
45:     Naključno odstrani  $|\mathbf{A}| - NP$  posameznike iz  $\mathbf{A}$ ;
46: end if
47:      $\mathbf{x}_{i,G+1} \rightarrow \mathbf{P}_{\text{new}}$ ;
48: end for
49: Izračunaj  $NP_{\text{new}}$  glede na:
50:  $NP_{\text{new}} = \text{round} \left( NP_0 - \frac{FES}{MAXFES} (NP_0 - NP_f) \right)$ ;
51: if  $NP_{\text{new}} < NP$  then
52:     Uredi posameznike v  $\mathbf{P}$  glede na njihove vrednosti kriterijske funkcije
53:     in odstrani  $NP - NP_{\text{new}}$  najslabših;
54:      $NP = NP_{\text{new}}$ ;
55: end if
56: if  $|\mathbf{A}| > NP$  then
57:     Naključno odstrani  $|\mathbf{A}| - NP$  posameznikov iz  $\mathbf{A}$ ;
58: end if
59: Posodobi  $M_{F,k}, M_{CR,k}$  s pomočjo Lehmerjevega povprečja:
60: if  $\mathbf{S}_F \neq \emptyset$  in  $\mathbf{S}_{CR} \neq \emptyset$  then
61:      $w_k = \frac{\sqrt{\sum_{j=1}^D (u_{k,j,G} - x_{k,j,G})^2}}{\sum_{m=1}^{|\mathbf{S}_{CR}|} \sqrt{\sum_{j=1}^D (u_{m,j,G} - x_{m,j,G})^2}}$ 
62:      $\text{mean}_{WL}(\mathbf{S}) = \frac{\sum_{k=1}^{|\mathbf{S}|} w_k \bullet S_k^2}{\sum_{k=1}^{|\mathbf{S}|} w_k \bullet S_k}$ ;
63:      $M_{F,k} = \begin{cases} \text{mean}_{WL}(\mathbf{S}_F) & \text{if } \mathbf{S}_F \neq \emptyset \\ M_{F,k} & \text{sicer} \end{cases}$ 
64:      $M_{CR,k} = \begin{cases} \text{mean}_{WL}(\mathbf{S}_{CR}) & \text{if } \mathbf{S}_{CR} \neq \emptyset \\ M_{CR,k} & \text{sicer} \end{cases}$ 
65:     Povečaj  $k := k + 1$  in če je novi  $k > H$ , ponastavi  $k$  na 1;
66: end if
67:  $\mathbf{P} = \mathbf{P}_{\text{new}}, \mathbf{P}_{\text{new}} = \{\}$ ,  $\mathbf{x}_{\text{best}} =$  najboljši iz populacije  $\mathbf{P}$ ,  $G++$ ;
68: end while
69: vrni  $\mathbf{x}_{\text{best}}$  kot najboljšo najdena rešitev;

```

Združen pristop z optimizacijo za ORLTES

V začetku tega poglavja so bili predstavljeni osnovni primeri, ki predstavljajo optimizacijske algoritme tipa DE (postopki 3, 4 in 5). Ti osnovni (temeljni) primeri bodo sedaj vstavljeni kot gradniki v primeru združenega pristopa za ORLTES.

Temeljni algoritem za optimizacijo iz začetka poglavja je v tem podpoglavju tako vstavljen kot aplikacija pri operacijskih raziskavah logističnih, transportnih in ekonomskih sistemov. S tem dobimo aplikacijo takega temeljnega algoritma. Primer take aplikacije je razložen v nadaljevanju skozi logistični sistem za planiranje poti podmorskih sond (angl. *Underwater Glider Path Planning*) (UGPP) — slika 3.1.



Aleš Zamuda and José Daniel Hernández Sosa. **Success history applied to expert system for underwater glider path planning using differential evolution.** *Expert Systems with Applications*, 2019, vol. 119, pp. 155–170. <http://dx.doi.org/10.1016/j.eswa.2018.10.048>

Arhitektura načrtovanega sistema algoritmov za operacijske raziskave logističnih, transportnih in ekonomskih sistemov je bila na primeru načrtovanja navigacije podvodnih sond prikazana že na sliki 3.1. Zbrano gradivo v tem podpoglavju, ki je zajeto in prevedeno iz zgornjega revijalnega prispevka (v reviji *ESWA*), prikazuje postopek 6, več o sistemu UGPP pa v poglavju 5. V naslednjem poglavju je predstavljen primer kompleksne prostorske geometrije, na primeru naravnih tvorb, kot so drevesa.

Postopek 6 Algoritem za ORLTES: L-SHADE@UGPP.

Vhod: Spremenljivi parametri — \mathbf{p}_0 (trenutna lokacija podmorske sonde), zemljevid (MyOcean IBI), $\mathbf{p}_{\text{target}}$ (namišljena ciljna točka misije), MAX_FES (največje število dodeljenih ovrednotenj kriterijske funkcije), konstante — p in $|A|$ (konstante L-SHADE za JADE [232] parametre: pohlepnost mutacije $p = 0,11$, velikost arhiva $|A|$ enaka NP), $NP_{\text{min}} = 4$ in $NP_{\text{init}} = 5D = 60$ (konstante L-SHADE za največjo in najmanjšo velikost populacije DE pri mehanizmu spreminjanja velikosti populacije, LPSR), H (konstante L-SHADE za nastavljanje parametra za velikost pomnilnika hrambe zgodovinsko uspešnih krmilnih parametrov F_i in CR_i).

Izhod: \mathbf{x} – seznam navodil (D postopnih sprememb smeri) za navigacijo podmorske sonde.

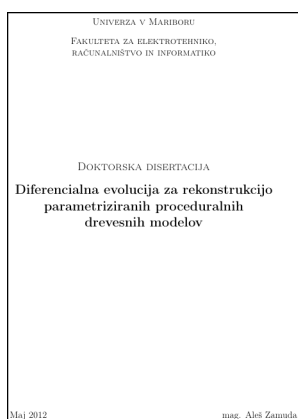
- 1: uniformno naključno generiramo začetno populacijo **DE** $\mathbf{x}_{i,0}, \forall i \in \{1, 2, \dots, NP_{\text{init}}\}; NP_0 = NP_{\text{init}};$
- 2: inicializiramo shrambe **L-SHADE**: $A = \emptyset, \forall k \in \{1, 2, \dots, H\}: M_{\text{CR},k} = M_{\text{F},k} = 0, 5;$
- 3: **for** Generacijska zanka **DE** $g = 1$ do $G, NFE < MAX_FES$ **do**
- 4: **for** Iteracijska zanka **DE**, za i (vsi posamezniki $\mathbf{x}_{i,g}$ v populaciji) **do**
- 5: Izračun novega poskusnega vektorja **DE**, $\mathbf{x}_{i,g}$:
 - 6: uniformno naključno generiramo $r_i \in [1, H];$
 - 7: $F_i = \text{randc}_i(M_{\text{F},r_i}, 0, 1);$
 - 8: $CR_i = \text{randn}_i(M_{\text{CR},r_i}, 0, 1);$
 - 9: $\mathbf{P} = \{\forall k \in [1, NP_g] : \mathbf{x}_{k,g}\} \cup \mathbf{A};$
 - 10: $p_{\text{best}} = \lfloor p|P| + 0, 5 \rfloor;$
 - 11: izberemo $\mathbf{x}_{p_{\text{best}},g}$ pri p_{best} iz \mathbf{P} , rangirani po ustreznosti;
 - 12: $\mathbf{v}_{i,g+1} = \mathbf{x}_{i,g} + F_i(\mathbf{x}_{p_{\text{best}},g} - \mathbf{x}_{i,g}) + F_i(\mathbf{x}_{r_1,g} - \mathbf{x}_{r_2,g});$
 - 13: $\forall j \in \{1, \dots, D\}: u_{i,j,g+1} = \begin{cases} v_{i,j,g+1} & \text{if } \text{rand}(0, 1) \leq CR_{i,g+1} \\ & \text{ali } j = j_{\text{rand}} \\ x_{i,j,g} & \text{sicer} \end{cases};$
- 14: Ovrednotenje ustreznosti v **DE** (simulacija **UGPP**):
 - 15: $b_0 = \text{bearing}(\mathbf{p}_0, \mathbf{p}_{\text{target}});$
 - 16: $b_1 = b_0 + u_{i,1,g+1};$
 - 17: $\mathbf{p}_1 = \text{simulate_stint}(\mathbf{p}_0, b_1, \text{map});$
 - 18: **for** $\forall j \in \mathbb{N} \mid 2 \leq j \leq D:$
 - 19: $b_j = b_{j-1} + u_{i,j,g};$
 - 20: $\mathbf{p}'_{j-1} = \text{simulate_drift}(\mathbf{p}_{j-1}, \text{map});$
 - 21: $\{\mathbf{p}_j\} = \{\mathbf{p}_{j-1}\} \cup \text{simulate_stint}(\mathbf{p}'_{j-1}, b_j, \text{map});$
 - 22: **end for**
 - 23: $f(\mathbf{u}_{i,g}) = \|\mathbf{p}_D - \mathbf{p}_{\text{target}}\|;$
 - 24: $\mathbf{x}_{i,g+1} = \begin{cases} \mathbf{u}_{i,g+1} & \text{če } f(\mathbf{u}_{i,g+1}) \leq f(\mathbf{x}_{i,g}) \\ \mathbf{x}_{i,g} & \text{sicer} \end{cases};$
 - 25: **if** $f(\mathbf{u}_{i,g+1}) < f(\mathbf{x}_{i,g}):$
 - 26: zamenjaj najslabšega posameznika v \mathbf{A} z $\mathbf{u}_{i,g+1}$ (ali ga dodaj, ko $|\mathbf{A}| < NP_g$);
 - 27: shrani propagirana F_i in CR_i za posodobitev krmilnih pomnilnikov $\mathbf{M}_{\text{F}}, \mathbf{M}_{\text{CR}};$
 - 28: **end if**
 - 29: **end for**
 - 30: posodobni pomnilnika \mathbf{M}_{F} in \mathbf{M}_{CR} s propagiranimi parametri;

```
31:   $NP_{g+1} = NP_{\min} + \left\lfloor \frac{NFE}{MAX\_FES} (NP_{\text{init}} - NP_{\min}) + 0,5 \right\rfloor$ ; // preve-  
    rimo LPSR  
32:  if  $NP_{g+1} < NP_g$  then  
33:    za populacijo in arhiv: zmanjšaj velikosti obeh na  $NP_{g+1}$  z odstra-  
    nitvijo najslabših posameznikov;  
34:  end if  
35: end for  
36: vrni najboljši posameznik, dobljen izmed  $\mathbf{x}_{i,G}$ ;
```

Poglavje 4

Kompleksnost v operacijskih raziskavah logističnih, transportnih in ekonomskih sistemov

Zbrano gradivo v tem poglavju je zajeto iz prispevkov:



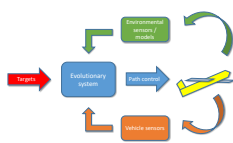
A. Zamuda. Diferencialna evolucija za rekonstrukcijo parametriziranih proceduralnih drevesnih modelov: doktorska disertacija. Fakulteta za elektrotehniko, računalništvo in informatiko, 2012. <https://dk.um.si/Dokument.php?id=30084>



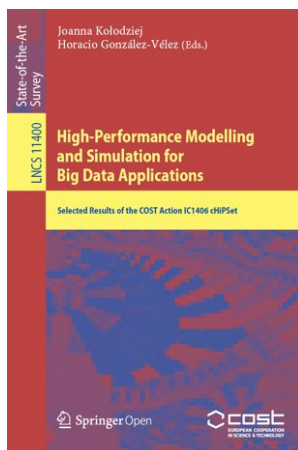
A. Zamuda, J. Brest. **Vectorized Procedural Models for Animated Trees Reconstruction using Differential Evolution**. Information Sciences, vol. 278, pp. 1-21, 2014. <http://dx.doi.org/10.1016/j.ins.2014.04.037>



H. Hamann, Y. Khaluf, J. Botev, M. Divband Sorati, E. Ferrante, O. Kosak, J.-M. Montanier, S. Mostaghim, R. Redpath, J. Timmis, F. Veenstra, M. Wahby and A. Zamuda. **Hybrid Societies: Challenges and Perspectives in the Design of Collective Behavior in Self-organizing Systems**. *Frontiers in Robotics and AI*, 2016, vol. 3, no. 14. <http://dx.doi.org/10.3389/frobt.2016.00014>



Aleš Zamuda and José Daniel Hernández Sosa. **Success history applied to expert system for underwater glider path planning using differential evolution**. *Expert Systems with Applications*, 2019, vol. 119, pp. 155–170. <http://dx.doi.org/10.1016/j.eswa.2018.10.048>



A. Zamuda, V. Crescimanna, J. C. Burguillo, J. Matos Dias, K. Wegrzyn-Wolska, I. Rached, H. González-Vélez, R. Senkerik, C. Pop, T. Cioara, I. Salomie, A. Bracciali. **Forecasting Cryptocurrency Value by Sentiment Analysis: An HPC-Oriented Survey of the State-of-the-Art in the Cloud Era**. Kołodziej J., González-Vélez H. (eds) *High-Performance Modelling and Simulation for Big Data Applications*. *Lecture Notes in Computer Science*, vol 11400, 2019, pp. 325-349. http://dx.doi.org/10.1007/978-3-030-16272-6_12

Zaradi nelinearnosti, soodvisnosti in nekonveksnosti kriterijev je ocenitvena funkcija pri zahtevnih modelih kompleksna. Primeri takšnih funkcij so rekurzivne funkcije, ki nastopajo pri rekonstrukciji morfologije naravnih dreves [208, 216].

Primer psevdokoda takšnega postopka je podrobno opisan v [216, 206], pri katerem je glavni del algoritma za rekonstrukcijo parametriziranega proceduralnega modela drevesa iz slike prikazan kot postopek 8.

Diferencialna evolucija za rekonstrukcijo parametriziranih proceduralnih drevesnih modelov

Pokažimo, kako je možno združiti optimizacijski algoritem diferencialne evolucije (opisan v podpoglavju 1) in numerično kodiran proceduralni model dreves (opisan v [206, 216]). Predlagan rekonstrukcijski postopek kodira parametre proceduralnega modela drevesa iz ogrodja EcoMod [217] v genotipu posameznika iz populacije algoritma jDE [21]. S predlaganim postopkom rekonstruiramo trodimenzionalni proceduralni model drevesa, tako da z algoritmom jDE skušamo ustrezno parametrizirati parametre proceduralnega modela [219, 207]. Rekonstrukcijski postopek temelji na primerjavi slik dreves, ki predstavljajo dvodimenzionalne projekcije referenčnih trodimenzionalnih dreves. Rekonstrukcija je toliko bolj uspešna, kolikor so si slike referenčnih dreves in slike upodobljenih parametriziranih proceduralnih modelov podobne. Parametrizacija proceduralnega modela poteka postopno in, bolj kot so si slike podobne, boljša (t.j. manjša) je optimirana vrednost ovrednotenja primerjave razlik med drevesi [215, 216].

V našem pristopu bomo za predstavitev trodimenzionalnega drevesa uporabili svoj zgrajen proceduralni model [224, 217], ki ga skušamo z rekonstrukcijo ustrezno parametrizirati. Rekonstrukcijski postopek temelji na razpoznavi dvodimenzionalnih fotografij naravnih dreves. Rekonstrukcija je toliko bolj uspešna, kolikor so si slike fotografij dreves in slike upodobljenih parametriziranih proceduralnih modelov podobne [11, 155]. Parametrizacija proceduralnega modela poteka postopno [186, 187], z evolucijo po vzoru iz narave, na primer algoritem jDE [21, 27, 215, 22, 209].

V nadaljevanju opisujemo nekatere pomembnejše dele rekonstrukcijskega postopka. Najprej je razloženo kodiranje genotipa, nato sledi preslikava genotipa v fenotip, nato še ovrednotenje fenotipa preko primerjave z referenčno sliko.

Kodiranje genotipa

Za predstavitev drevesa smo uporabili svoj numerično kodiran proceduralni drevesni model iz programskega sistema EcoMod [217], predstavljenega v [206, 216]. Ta model uporabniku pomaga zgraditi drevesno geometrijo, tako da zmanjša nabor parametrov, ki jih uporabnik mora nastaviti, da bi modeliral geometrijski model drevesa. Tak nabor parametrov sestavljajo npr. debeline vej, relativne dolžine vej in proporci vejitvene strukture. Nabor poimenujemo tudi gojitveno seme drevesa [186, 187], saj so v njem zajeti vsi parametri za izračun drevesne geometrije. Ko namreč uporabnik poda to seme, postopek

v proceduralnem modelu samodejno rekurzivno določi drevesno strukturo (t.j. morfologijo) in s tem lokacije, rotacije, velikosti ter teksture sestavnih delov drevesa, kot so več tisoč vejnih odsekov in več tisoč listov. Vsak korak rekurzije izračuna dodaten drevesni gradnik v geometrijskem modelu. Na tak način nad danim naborom numerično kodiranih vhodnih parametrov, kot so npr. širina debla, relativna dolžina vej in vejitvena struktura, procedura zgradi trodimenzionalno drevesno strukturo [200]. Končni geometrijski model posameznega drevesa je generiran s parametrizacijo proceduralnega modela. Parametriziran proceduralni model lahko kasneje uporabimo za računalniško animacijo, saj s spremembo glavnih ali pomožnih parametrov modela dobimo nekoliko spremenjene podobne modele, ki so primerni za izračunavanje ključnih animacijskih okvirjev [218].

Kot je model definiran v [206, 216], parametre izbranega proceduralnega modela ločimo na skalarne in matrične (vektorske). Ta nabor skalarnih in matričnih parametrov tvori v prejšnjem odstavku omenjeno gojitveno seme. V [217] smo predstavili tudi vizualno interaktivno orodje s posebno parametrizacijo matričnih parametrov s konvolucijo parametrov po indeksih, ki smo jih definirali z lomljenko na grafu. Izbran model se od nekaterih drugih modelov [4, 14, 85, 158, 140] razlikuje v tem, da je popolnoma numerično kodiran in je nespremenljive dimenzije. Zaradi tega je ta model še posebej primeren za iskanje parametrov z uporabo diferencialne evolucije [215].

Posamezen genotipni vektor \mathbf{x} , $i \in \{0, \dots, NP - 1\}$ iz populacije v algoritmu jDE kodira nabor parametrov proceduralnega modela v gojitvenem semenu. Dimenzionalnost razvijanega realno-kodiranega genotipa \mathbf{x} je $D = 4509$. Genotip sestoji iz komponent $\mathbf{x} = x_{i,j} \in [0, 1]$, $j \in \{0, \dots, D - 1\}$, ki kodirajo naslednje parametre gojitvenega semena:

- število žil v drevesu $S = 400x_{i,0} + 10$, $S \in [10, 410]$ (določa kompleksnost drevesa: enako številu vej z listi, vejnih odsekov manj ena ali polovici vej manj dve),
- višino prvega odseka debla $l_0^{0,0} = 10x_{i,1}$, $l_0^{0,0} \in [0 \text{ m}, 10 \text{ m}]$,
- koeficient debeline veje $k_d = 0,05x_{i,2}$, $k_d \in [0, 0,05]$,
- kot filotakse $\alpha_p = 360x_{i,3}$, $\alpha_p \in [0^\circ, 360^\circ]$,
- razmerje porazdelitve žil na podveji pri vejitvah $k_s^{g,w} = 0,5x_{i,j} + 0,5$, $\forall j \in \{4, \dots, 753\}$, $k_s^{g,w} \in [0,5, 1]$,
- kot med izhajajočima podvejama pri delitvi $\alpha^{g,w} = 180x_{i,j}$, $\forall j \in \{754, \dots, 1503\}$, $\alpha^{g,w} \in [0^\circ, 180^\circ]$,

- zgornja meja relativne dolžine podvej glede na dolžino osnovne veje $M^{g,w} = 20x_{i,j}$, $\forall j \in \{1504, \dots, 2253\}$, $M^{g,w} \in [0, 20]$,
- spodnja meja relativne dolžine podvej glede na dolžino osnovne veje $m^{g,w} = 20x_{i,j}$, $\forall j \in \{2254, \dots, 3003\}$, $m^{g,w} \in [0, 20]$,
- skalirni faktor dolžine veje $k_1^{g,w} = 20x_{i,j}$, $\forall j \in \{3004, \dots, 3753\}$, $k_1^{g,w} \in [0, 20]$,
- gravicentralizem $k_c = x_{i,3754}$, $k_c \in [0, 1]$,
- gravimorfizem (t.j. upogibanje vej zaradi gravitacije) $\alpha_m^{g,w} = 360x_{i,j} - 180$, $\forall j \in \{3755, \dots, 4504\}$, $\alpha_m^{g,w} \in [-180^\circ, 180^\circ]$,
- vključitev listov na drevesu $B_1 = x_{i,4505} < 0,5 ? 0 : 1$, $B_1 \in \{0, 1\}$,
- velikost listov $l_1 = 0,3x_{i,4506}$, $l_1 \in [0, 0,3]$,
- gostoto listov $\rho_1 = 30x_{i,4507}$, $\rho_1 \in [0, 30]$,
- tip porazdelitve listov $l_{\text{type}} = 5 \lfloor x_{i,4508} \rfloor$ z indeksiranimi vrednostmi spiralno (nastavitev v EcoMod, **Spiral**), naloženo (nastavitev v EcoMod, **Stacked**), zloženo (nastavitev v EcoMod, **Staggered**), speto (nastavitev v EcoMod, **Bunched**) in igličasto (nastavitev v EcoMod, **Coniferous**),

kjer je $g \in [0, 15]$, $w \in [0, 50]$ in po 750 realnih parametrov kodira po eno matriko matričnega parametra. Pri tem j označuje indeks komponente v genotipu in za matrične parametre v gojitvenem semenu teče kot $j = o + (50g + w)$, kjer z o označimo indeksni odmik začetka kodiranja vektorskega parametra iz gojitvenega semena v genotipu.

Z opisanim kodiranjem je možno predstaviti številne različne proceduralne modele dreves [217], podajmo nekaj primerov. Drevo, podobno bukvi, ki je prikazano kot četrto drevo iz leve na sliki 4.1, nastane iz gojitvenega semena s $S = 400$ žilami in drugimi skalarnimi parametri $S = 400$, $L_0 = 5$, $k_d = 0,01$, $\alpha_p = 85$, $k_c = 0,3$, $l_1 = 0,1$, $\rho_1 = 5$, $B_1 = 1$, $l_{\text{type}} = \text{Spiral}$, $w_g = 1,5$, $\mathbf{w} = [0 \ 1 \ 0]^T$, $k_f = 1$ ter $l_{\text{LOD}} = 0$. Da lažje opišemo še matrične parametre tega semena, jih bomo podali s pomočjo pomožnih enodimenzionalnih vektorskih parametrov, ki preko konvolucije svojih grafov posredno izražajo pripadajoče matrične parametre. Grafe, ki jih nad temi vektorji definiramo, podamo z lomljenkami, kot smo omenili zgoraj. Definicija teh grafov lahko v našem orodju EcoMod poteka tudi interaktivno, z določitvijo krmilnih točk njihovih lomljenk [217]. Vrednosti na ordinatni osi so realna števila,

ki jih na položajih indeksov g in w vzorčimo iz interpolirane lomljenke. Matrični parameter porazdelitve žil tako izračunamo iz pomožnih enodimenzionalnih vektorskih parametrov iz enačbe za porazdelitev žil [206, 216]. Za omenjeno drevo iz slike 4.1 pomožna parametra $k_s^g \in [\frac{1}{2}, 1]$ in $k_s^w \in [0, 2]$ zavzameta vrednosti $\mathbf{k}_s^g = \{(0, 0, 75), (15, 0, 9)\}$ in $\mathbf{k}_s^w = \{(0, 1), (50, 1)\}$. Matrični parameter kotov vejitev med izhajajočima podvejama izračunamo iz enačbe za kot vejitve [206, 216] in za omenjeno drevo iz slike 4.1 pomožna parametra $\alpha^g \in [0^\circ, 180^\circ]$ in $\alpha^w \in [0, 2]$ zavzameta vrednosti $\boldsymbol{\alpha}^g = \{(0, 45^\circ), (15, 45^\circ)\}$ in $\boldsymbol{\alpha}^w = \{(0, 1), (50, 1)\}$. Matrični parameter največje relativne dolžine vej izračunamo iz enačbe za dolžine vej [206, 216] in za omenjeno drevo iz slike 4.1 pomožna parametra $M^g \in [0, 10]$ in $M^w \in [0, 2]$ zavzameta vrednosti $\mathbf{M}^g = \{(0, 1), (15, 1)\}$ in $\mathbf{M}^w = \{(0, 1), (50, 1)\}$. Matrični parameter najmanjše relativne dolžine vej izračunamo iz enačbe za gravimorfizem [206, 216] in za omenjeno drevo iz slike 4.1 pomožna parametra $m^g \in [0, 10]$ in $m^w \in [0, 2]$ zavzameta vrednosti $\mathbf{m}^g = \{(0, 1), (7, 5, 0, 45), (15, 1)\}$ in $\mathbf{m}^w = \{(0, 1), (50, 1)\}$. Matrični parameter skaliranja dolžine vej izračunamo iz enačbe za skaliranje vej [206, 216] in za omenjeno drevo iz slike 4.1 pomožna parametra $k_l^g \in [0, 10]$ in $k_l^w \in [0, 2]$ zavzameta vrednosti $\mathbf{k}_l^g = \{(0, 1), (15, 1)\}$ in $\mathbf{k}_l^w = \{(0, 1), (50, 1)\}$. Matrični parameter vpliva gravitacije, gravimorfizem, izračunamo iz enačbe za značilne dolžine vej [206, 216] in za omenjeno drevo iz slike 4.1 pomožna parametra $\alpha_m^g \in [-180^\circ, 180^\circ]$ in $\alpha_m^w \in [0, 2]$ zavzameta vrednosti $\boldsymbol{\alpha}_m^g = \{(0, 0^\circ), (15, 0^\circ)\}$ in $\boldsymbol{\alpha}_m^w = \{(0, 1), (50, 1)\}$. Gojitvena semena prvih štirih dreves iz leve na sliki 4.1 se razlikujejo zgolj v enem parametru, ki zaporedoma zavzame vrednosti $S = \{10, 50, 100, 400\}$. Gojitveno seme petega drevesa iz leve na tej sliki je enako kot pri četrtem, vendar je upodobljeno brez tekstur in listov ($B_1 = 0$), z žičnim modelom. Skrajno desno drevo se razlikuje glede na tretje v parametrih $k_s^g = \{(0, 0, 5), (15, 0, 9)\}$.

Preslikava genotipa v fenotip

Naš rekonstrukcijski postopek temelji na rekonstrukciji iz dvodimenzionalnih slik naravnih dreves $\mathbf{z}^* = \{z_{x,y}^*\}$, $\forall x = \{0, \dots, X-1\}$, $\forall y = \{0, \dots, Y-1\}$. Da bi z drevesom iz referenčnih slik \mathbf{z}_i^* primerjali razvito tridimenzionalno drevo, kodirano v genotipu \mathbf{x} dimenzije D , genotip najprej preslikamo v fenotip. Fenotip je nabor upodobljenih dvodimenzionalnih slik razvitega trodimenzionalnega drevesa, $\mathbf{z}_i = \{z_{x,y}^i\}$, $\forall x = \{0, \dots, X-1\}$, $\forall y = \{0, \dots, Y-1\}$. Fenotip izračunamo z grafičnim upodabljanjem genotipa \mathbf{x} , ki predstavlja seme za vhodne parametre parametriziranega proceduralnega modela za drevo v sistemu EcoMod. Geometrijski model in fenotipne slike izračunamo z algoritmom 7. Slednji algoritem pri gradnji sestavnih delov drevesa uporablja



Slika 4.1: Upodobitev proceduralnih modelov drevesa, podobnega bukvi, z različnim številom žil, žični model skeleta drevesne strukture brez listov in skrajno desno, enako drevo z drugačno vejitveno strukturo.

geometrijske interpolacije pri postavitvi krmilnih točk za vozlišča poligonov, ki jih upodobi in s projekcijo pretvori iz treh dimenzij v dve dimenziji za teksturirane dvodimenzionalne barvne slike.

Slike \mathbf{z}_i^* in \mathbf{z}_i so dimenzij $X \times Y$. Slike pred medsebojno primerjavo še pretvorimo v črno bele, tako da vsak piksel materiala, ki je različen od ozadja, postavimo na 1 (črno), ostale piksele na 0 (belo). Slike referenčnega drevesa pred pričetkom optimizacije še skaliramo na dimenzijo slik projekcij razvitih modelov. Upodobitev istega razvitega parametriziranega proceduralnega modela vzorčimo večkrat v trodimenzionalni geometrijski model in nato dvodimenzionalno projekcijo, da dobimo različne projekcije. Vzorčimo vsaj dvakrat, iz različnih kotov pogleda kamere, npr. zamaknjenih za $\beta = 90^\circ$ vzdolž debla. Vzrok za to je, da z izogibanjem ploskim drevesom zagotovimo trodimenzionalno razsežnost rekonstruiranih modelov. Če označimo algoritem 7 s funkcijskim predpisom \mathbf{g} , potem je $\mathbf{z}_i = \mathbf{g}(\mathbf{x}, \beta_c)$, $c = \{1, 2\}$ kjer $\beta_1 = 0^\circ$ in $\beta_2 = 90^\circ$.

Primerjava fenotipa z referenčno sliko

Fenotip ovrednotimo z oceno rekonstrukcijske napake. To oceno merimo z vsoto razlikovanj med slikami razvitih modelov in referenčnimi slikami. Rekonstrukcija je toliko bolj uspešna, kolikor bolj so istoležni piksli v referenčnih slikah dreves in v slikah upodobljenih parametriziranih proceduralnih modelov enaki. V ta namen smo izbrali naslednji kriterij. Metrika za kriterij je vsota manhattanskih razdalj od vsakega črnega (t.j. materialnega) piksla slike do najbližjega črnega piksla referenčne slike in obratno [11]. Posamezni fenotip tako ovrednotimo s kriterijem f_1 :

$$f_1(\mathbf{x}) = f(\mathbf{g}(\mathbf{x}, \beta_1), \mathbf{g}(\mathbf{x}, \beta_2)) = h_1(\mathbf{z}_1) + h_1(\mathbf{z}_2) \quad (4.1)$$

$$h_1(\mathbf{z}_i) = \sum_{x,y} m_1(z_{x,y}^i, z_{i;x,y}^*) + \sum_{x,y} m_1(z_{i;x,y}^*, z_{x,y}^i) \quad (4.2)$$

kjer z m_1 označimo funkcijo za izračun manhattanske razdalje do najbližjega piksla v sliki \mathbf{z}^* z vrednostjo 1 (t.j. črno, material), $i \in \{1, 2\}$. Kot vidimo, razlikovanje slik ovrednotenje ocene s povečanjem poslabša, zato je naloga optimizacije v našem primeru, najti čim manjše vrednosti obeh funkcij.

Definicijo celotnega predlaganega postopka *diferencialne evolucije za rekonstrukcijo parametriziranih proceduralnih modelov* podaja algoritem 8. Kot je vidno iz uporabe funkcije `g`, algoritem z dvojno zanko `for` ovija algoritem 7, ki ocenjuje razvite modele. Po zaključku omenjenih dvojnih zank algoritem vrne nabor parametriziranih proceduralnih modelov, t.j. njihova gojitvena semena.

Kompleksnost v kibernetско-tehnični sistemih postopkov

V kibernetско-tehničnih sistemih najdemo številne izzive zaradi kompleksnosti, na konkretnem primeru podmorskih sond obravnava nekatere izmed teh že omenjena objava [231] (algoritem je podan v prejšnjem poglavju). Splošneje pa to tematiko obravnava objava [77], ki osvetli nekaj določenih izzivov iz tematike modeliranja kibernetско-tehničnih sistemov, posebej v kontekstu sodelovanja med družbami (hibridne družbe).

Iz vidika analize ekonomskih sistemov je na primer obravnavana sestava portfelja z optimizacijskimi algoritmi, kar je opisano v [230].

Postopek 7 Izračun geometrijske strukture s proceduralnim drevesnim modelom. Rekurzivni algoritem pričnemo s klicem $\text{odsekveje}(0, 0, S, 1, l_0^{0,0}, \mathbf{I}, \mathbf{I})$, kjer \mathbf{I} označuje matriko identitete.

Vhod: g, w - Graveliusov in Weibullov indeks osnovne veje; S_0 - skupno število žil v osnovni veji; L_0, l_0 - relativna in dejanska dolžina osnovne veje; \mathbf{M}_0 - koordinatni sistem osnovne veje; $\mathbf{M}_{m;0}^{-1}$ - v koordinatnem sistemu osnovne veje izražena inverzna rotacijska matrika za gravimorfizem; $k_d, k_c, l_{\text{type}}, k_s^{g,w}, M^{g,w}, m^{g,w}, k_1^{g,w}, \alpha_m^{g,w}, \alpha^{g,w}, t, k_f, w_s, w_g$ - podani parametri *semena*.

Izhod: slika upodobljenega drevesa,

- 1: **postopek** $\text{odsekveje}(g, w, S_0, L_0, l_0, \mathbf{M}_0, \mathbf{M}_{m;0}^{-1})$:
 - 2: $d := k_d \sqrt{S_0}$; {izračun debeline osnovne veje}
 - 3: upodobi osnovno vejo(\mathbf{M}_0, l_0, d);
 - 4: **if** $S_0 = 1$ **then**
 - 5: upodobi liste(l_{type}); **return**; {zadnji nivo, upodobimo liste}
 - 6: **end if**
 - 7: $S_1 := \lceil 1 + k_s^{g,w} (S_0 - 2) \rceil, S_2 = S_0 - S_1$; {število žil v podvejah}
 - 8: $r_1 := \max \left\{ \min \left\{ \sqrt{\frac{S_1}{S_0}}, M^{g,w} \right\}, m^{g,w} \right\}$; {razmerja dolžin podvej glede na žile}
 - 9: $r_2 := \max \left\{ \min \left\{ \sqrt{\frac{S_2}{S_0}}, M^{g,w} \right\}, m^{g,w} \right\}$;
 - 10: $L_1 := r_1 L_0, L_2 := r_2 L_0$; {relativna dolžina podvej}
 - 11: $l_1 := k_1^{g,w} L_1, l_2 := k_1^{g,w} L_2$; {aktivna dolžina podvej}
 - 12: $\alpha_1 := k_c \sqrt{\frac{S_2}{S_0}} \alpha^{g,w}, \alpha_2 := \alpha^{g,w} - \alpha_1$; {koti vejitve}
 - 13: $\mathbf{M}_1 := \mathbf{R}_z(\alpha_1) \mathbf{R}_y(\alpha_p) \mathbf{R}_{\mathbf{y} \times \mathbf{y}_m}(\alpha_m^{g,w}) \mathbf{T}_y(l_0) \mathbf{M}_0$; {matrike premika in rotacije}
 - 14: $\mathbf{M}_2 := \mathbf{R}_z(\alpha_2) \mathbf{R}_y(\alpha_p) \mathbf{R}_{\mathbf{y} \times \mathbf{y}_m}(\alpha_m^{g,w}) \mathbf{T}_y(l_0) \mathbf{M}_0$;
 - 15: $\mathbf{M}_{m;1}^{-1} := \mathbf{R}_{\mathbf{y} \times \mathbf{y}_m}(-\alpha_m^{g,w}) \mathbf{R}_y(-\alpha_p) \mathbf{R}_x(-\alpha_x(t)) \mathbf{R}_z(-\alpha_1 - \alpha_z(t)) \mathbf{M}_{m;0}^{-1}$; {osvežitev inverzne matrike za postavitev vektorja gravimorfizma}
 - 16: $\mathbf{M}_{m;2}^{-1} := \mathbf{R}_{\mathbf{y} \times \mathbf{y}_m}(-\alpha_m^{g,w}) \mathbf{R}_y(-\alpha_p) \mathbf{R}_x(-\alpha_x(t)) \mathbf{R}_z(-\alpha_2 - \alpha_z(t)) \mathbf{M}_{m;0}^{-1}$;
 - 17: $\text{odsekveje}(g + 1, w + 1, S_2, L_2, l_2, \mathbf{M}_2, \mathbf{M}_{m;2}^{-1})$; {razvoj stranske veje}
 - 18: $\text{odsekveje}(g, w + 1, S_1, L_1, l_1, \mathbf{M}_1, \mathbf{M}_{m;1}^{-1})$; {razvoj glavne veje}
 - 19: **return**; {sestop iz rekurzivnega izračuna za posamezen odsek veje v drevesu}
-

Postopek 8 Morfološka rekonstrukcija kompleksnega drevesa iz narave(\mathbf{z}^*).

Vhod: S_0 - največje število žil v deblu; drugi parametri jDE [21] in Eco-Mod [217].

Izhod: rekonstruiran parametriziran proceduralni trodimenzionalni model drevesa.

1: Uniformno naključno generiraj začetno populacijo **DE** $\mathbf{x}_{i,0} \in [0, 1]$ za $i = 1..NP$;

2: **for** **DE** generacijska zanka g (dokler FEs < 10000) **do**

3: **for** **DE** iteracijska zanka i (za vse vektorje $\mathbf{x}_{i,g}$ v trenutni populaciji) **do**

4: **DE** izračun poskusnega vektorja $\mathbf{x}_{i,g}$ (samoprilagajanje, mutacija, križanje):

5:
$$F_{i,g+1} = \begin{cases} F_1 + rand_1 \times F_u & \text{če } rand_2 < \tau_1, \\ F_{i,g} & \text{sicer} \end{cases};$$

6:
$$CR_{i,g+1} = \begin{cases} rand_3 & \text{če } rand_4 < \tau_2, \\ CR_{i,g} & \text{sicer} \end{cases};$$

7:
$$\mathbf{v}_{i,g+1} = \mathbf{x}_{r_1,g} + F_{i,g+1}(\mathbf{x}_{r_2,g} - \mathbf{x}_{r_3,g});$$

8:
$$u_{i,j,g+1} = \begin{cases} v_{i,j,g+1} & \text{če } rand(0, 1) \leq CR_{i,g+1} \text{ ali } j = j_{rand}; \\ x_{i,j,g} & \text{sicer} \end{cases};$$

9: **DE** ocena ustreznosti (preslikava genotipa, upodabljanje in primerjava):

10:
$$\mathbf{z}_1 = \mathbf{g}(\mathbf{u}_{i,g}, \beta_1), \mathbf{z}_2 = \mathbf{g}(\mathbf{u}_{i,g}, \beta_2) \text{ \{klic postopka za upodobitev drevesa, dvakrat za en genotip\}}$$

11:
$$h(\mathbf{z}_1) = \sum_{x,y} m_1(z_{x,y}^1, z_{x,y}^*) + \sum_{x,y} m_1(z_{x,y}^*, z_{x,y}^1); \text{ \{Metrika podobnosti\}}$$

12:
$$h(\mathbf{z}_2) = \sum_{x,y} m_1(z_{x,y}^2, z_{x,y}^*) + \sum_{x,y} m_1(z_{x,y}^*, z_{x,y}^2); \text{ \{Za 3D, rotacija } 90^\circ \}}$$

13:
$$f(\mathbf{u}_{i,g}) = f(\mathbf{g}(\mathbf{u}_{i,g}, \beta_1), \mathbf{g}(\mathbf{u}_{i,g}, \beta_2)) = h(\mathbf{z}_1) + h(\mathbf{z}_2); \text{ \{Ocena ustreznosti\}}$$

14: **DE** selekcija:

15:
$$\mathbf{x}_{i,g+1} = \begin{cases} \mathbf{u}_{i,g+1} & \text{če } f(\mathbf{u}_{i,g+1}) < f(\mathbf{x}_{i,g}); \\ \mathbf{x}_{i,g} & \text{sicer} \end{cases};$$

16: **end for**

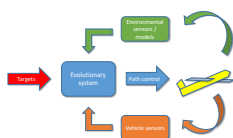
17: **end for**

18: **vrni** najboljšega najdenega posameznika;

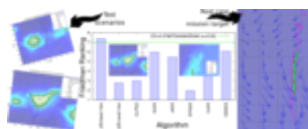
Poglavje 5

Planiranje kompleksnih misij in scenarijev v nepredvidljivih okoljih

Zbrano gradivo, ki je zajeto v tem poglavju, je prevedeno iz prispevkov:



Aleš Zamuda and José Daniel Hernández Sosa. **Success history applied to expert system for underwater glider path planning using differential evolution.** *Expert Systems with Applications*, 2019, vol. 119, pp. 155–170. <http://dx.doi.org/10.1016/j.eswa.2018.10.048>



A. Zamuda, J. D. Hernández Sosa. **Differential Evolution and Underwater Glider Path Planning Applied to the Short-Term Opportunistic Sampling of Dynamic Mesoscale Ocean Structures.** *Applied Soft Computing*, vol. 24, November 2014, pp. 95-108. <http://dx.doi.org/10.1016/j.asoc.2014.06.048>

V tem poglavju je predstavljena tematika implementacija podpore **navigaciji avtonomnih globokomorskih sond**. S **prostorskimi omejitvami** in **realno-časovnimi spremenljivimi** podatki o **oceanskih tokovih** pod vodo, ta podpora zahteva pripravo misij (ki so precej kompleksne). Uvedena je pomoč pri avtomatizaciji optimizacije poti: pilotu sond olajšamo navigacijo znotraj **časovnih omejitev za planiranje**, posebej pri **upravljanju flot**. Takšna avtomatizacija je še toliko bolj dobrodošla pri primerih **vzorčenja morskih vrtincev**, saj mora pilot v teh primerih pogosto pošiljati **nove popravke smeri**; pogosto tudi ob **vnovičnih zagonih različnih scenarijev** dela misije in ker se spreminja tudi vreme in razmere na morju, je potrebno upoštevati tudi te.

Za ta namen je pripravljen algoritem po nedavnih študijah napredka v **modernih optimizacijskih algoritmih po vzoru iz narave**, z **diferencialno evolucijo** in prilagodljivo obravnavo **omejitev** ter primerjan z nekaterimi drugimi pristopi.

Robotska morska sonda Slocum G2

Glede na tehnične specifikacije¹ ima ta podmorska robotska sonda veliko vzdržljivost, lahko je 25 do 365 dni na morju za eno misijo. Njen doseg je od 600 do 1500 km (z alkalnimi baterijami) oziroma od 4000 do 6000 km (z Litij-ionskimi baterijami). Njen vzgonski pogon je horizontalno 0,35m/s (0,68 vozla) in 2 vozla s propelerjem. Potopi se do globine 1000 metrov, ima dolg doseg in je modularna, saj integrira senzorje tako fizikalnih parametrov, kot biokemičnih – senzorje temperature, zasoljenosti, raztopljenega kisika, kalnosti, klorofila in morskih tokov, možna pa je tudi hitra zamenjava senzorjev.

Vzgonski pogon in uporabnost podmorskih sond

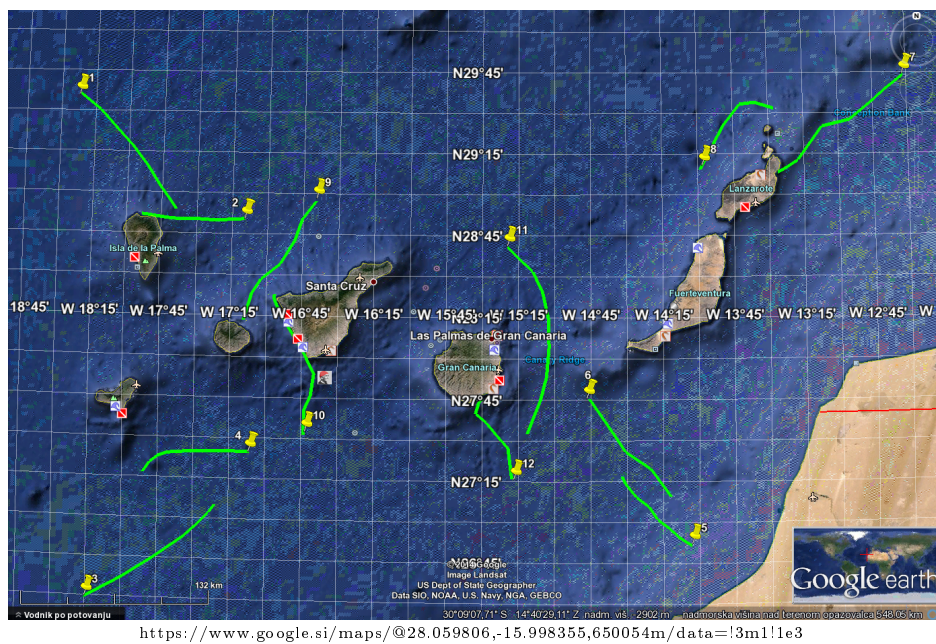
Podmorske sonde (jadrilice) delujejo na pogon "yoyo", ki porabi malo energije, večinoma ob spustu in dvigu (saj se vklopi črpalka); tudi za ohranjanje smeri porabi malo energije [231]. Uporabljajo se za izboljševanje oceanskih modelov z realnimi podatki, saj zajemajo dejanske podatke na mestu zajema, vzorčijo lahko tokove naftnih izpustov, služijo lahko za pregled in detekcijo nepravilnosti (ploščadi), spremljanje (naftnih) vodov in realnočasovno spremljanje senzorskih podatkov.

Satelitska navigacija in avtonomnost

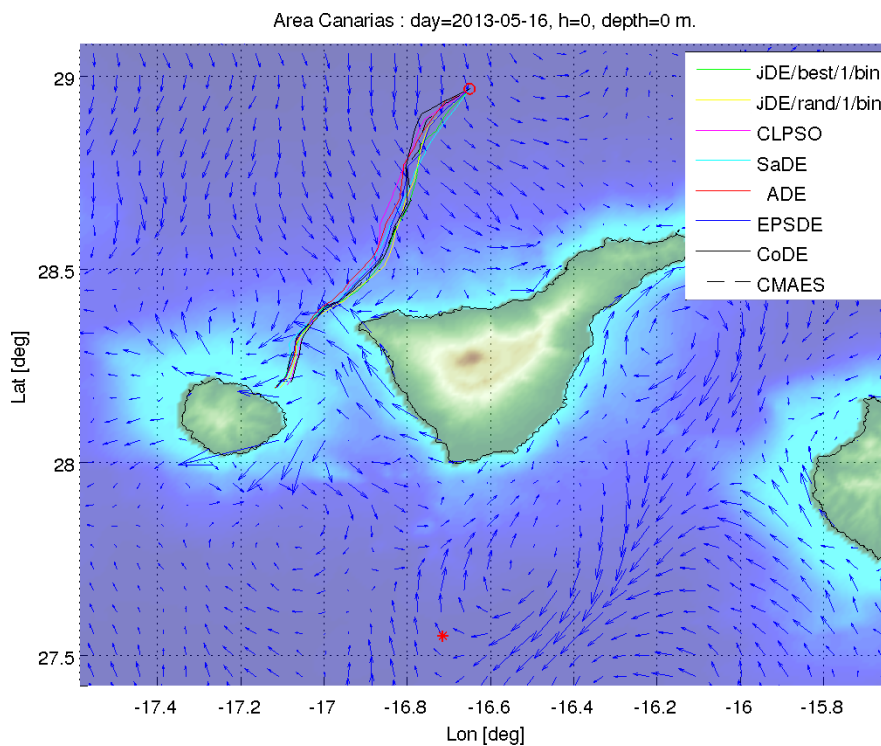
Radijski valovi vode ne prodrejo globoko, avtonomna sonda (AUV) hitro izgubi signal GPS [231].

Med potopom AUV navigira avtonomno, AUV za to uporablja interne senzorje, kot so kompas, globinomer, sonar (dna¹), žiroskop, pospeškometer, magnetometer, termistor (termometer), merilec prevodnosti in akustični modem za bežično komunikacijo s podvodnimi privezanimi senzorji.

¹http://www.webbresearch.com/pdf/Slocum_Glider_Data_Sheet.pdf



Slika 5.1: Scenariji za UGPP: izris površja trajektorij globinskih vzorčenj.



Slika 5.2: Primer različnih trajektorij za UGPP, izrisanih le na površju globinskih vzorčenj.

Postopek 9 Ovrednotenje ustreznosti poti **AUV** (simulacija **UGPP**).

Vhod: \mathbf{p}_0 (trenutna pozicija vozila), zemljevid (MyOcean IBI), $\mathbf{p}_{\text{target}}$ (navidezna točka tarče), $\mathbf{u}_{i,g+1}$ (poskusni vektor v **DE** z D postopnih sprememb smeri);

Izhod: $f(\mathbf{u}_{i,g})$ – ovrednotenje podanega **DE** vektorja;

- 1: $b_0 = \text{bearing}(\mathbf{p}_0, \mathbf{p}_{\text{target}})$;
 - 2: $b_1 = b_0 + u_{i,1,g+1}$;
 - 3: $\mathbf{p}_1 = \text{simulate_stint}(\mathbf{p}_0, b_1, \text{zemljevid})$;
 - 4: **for** $\forall j \in \mathbb{N} \mid 2 \leq j \leq D$:
 - 5: $b_j = b_{j-1} + u_{i,j,g}$;
 - 6: $\mathbf{p}'_{j-1} = \text{simulate_drift}(\mathbf{p}_{j-1}, \text{zemljevid})$;
 - 7: $\{\mathbf{p}_j\} = \{\mathbf{p}_{j-1}\} \cup \text{simulate_stint}(\mathbf{p}'_{j-1}, b_j, \text{zemljevid})$;
 - 8: **end for**
 - 9: $f(\mathbf{u}_{i,g}) = \|\mathbf{p}_D - \mathbf{p}_{\text{target}}\|$;
-

Realno-časovni podatkovni tokovi o okolju in naloga optimalne morske trajektorije

Za dostop do realno-časovnih podatkovnih tokov o okolju za modeliranje misije lahko uporabimo MyOcean IBI (<http://myodata.puertos.es/>), ki hrani raznovrstne satelitske podatke o morju (npr. morski tokovi). Podrobneje, uporabimo lahko *Regional Ocean Modelling System*, ki vsebuje osvežitev modela na 4 ure, pokriva 19°Z 5°V 26°S 56°S , resolucija $1/36^\circ$, poleg tega pa dodamo še nadomestni 3D model tokov, ki je ekstrapolacija iz urnih 2D površinskih podatkov in 3D interpolacija iz sosednjih točk.

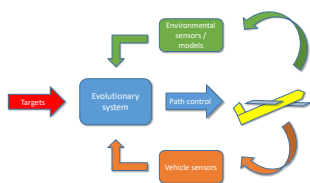
Izziv srednjeročnih vrtnicev nato definiramo nad vulkanski otoki in oceanskem morskem okolju, v katerem so umeščeni. Naloga optimalne trajektorije nastopi na podlagi tega izziva. Ocenitvena funkcija določi ustreznostni kriterij, ki na podlagi simulacije modela realne misije (postopek 9 [231]) lahko služi za izračunavanje optimalnih trajektorij, ki so nato uporabljene v ekspertnem sistemu za avtomatsko vodenje avtonomnih podmorskih sond [231]. Primer Slika 5.1 prikazuje scenarije za **UGPP**, kjer so približki trajektorij izrisani na površju. Slika 5.2 prikazuje primer enega izračuna scenarija.

V tem poglavju so bile predstavljene nekatere podrobnosti uporabe postopka za ovrednotenje ustreznosti poti **AUV** v sklopu **ORLTES** za **UGPP** kot naloge. To nalogo prejme **DE** v izračun (optimalne morske trajektorije), kot je že prikazal postopek 6, v vrstici 14 v tem postopku je ovrednotenje vektorja **DE** preko simulacije. V naslednjem poglavju sledi obravnava omejitev, ki jo lahko vključimo v **ORLTES** tako znotraj **DE** ali simulatorja.

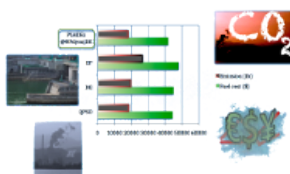
Poglavje 6

Obravnavanje omejitev pri operacijskih raziskavah logističnih, transportnih in ekonomskih sistemov: primeri iz globokomorske navigacije in ekonomskega razporejanja s scenariji za ovrednotenje kakovosti

Zbrano gradivo, ki je zajeto v tem poglavju, je prevedeno iz prispevkov:



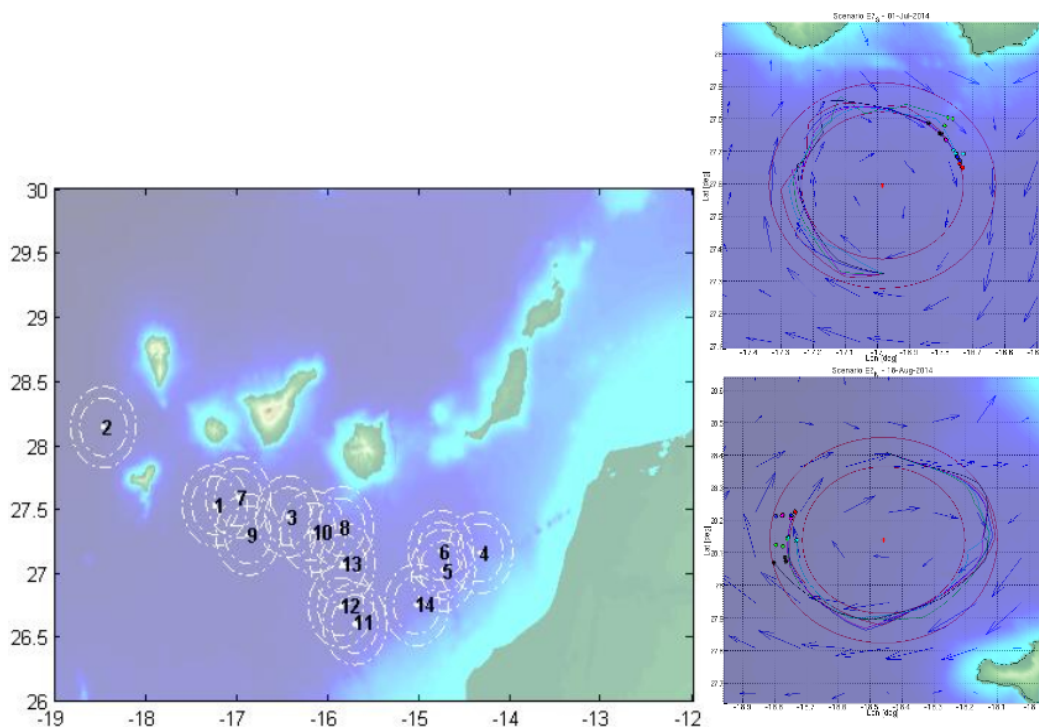
Aleš Zamuda and José Daniel Hernández Sosa. **Success history applied to expert system for underwater glider path planning using differential evolution.** *Expert Systems with Applications*, 2019, vol. 119, pp. 155–170. <http://dx.doi.org/10.1016/j.eswa.2018.10.048>



A. Glotić, A. Zamuda. **Short-term combined economic and emission hydrothermal optimization by surrogate differential evolution.** *Applied Energy*, 1 March 2015, vol. 141, pp. 42-56. <http://dx.doi.org/10.1016/j.apenergy.2014.12.020>

Morske trajektorije za globinsko pokrivanje vrtnicev na srednjeročnem prostorskem območju z omejitvami oblike koridorja

V delu [222] je predstavljen pristop **UGPP**, kjer imajo morske trajektorije za globinsko pokrivanje vrtnicev na srednjeročnem prostorskem območju omejitve oblike koridorja.

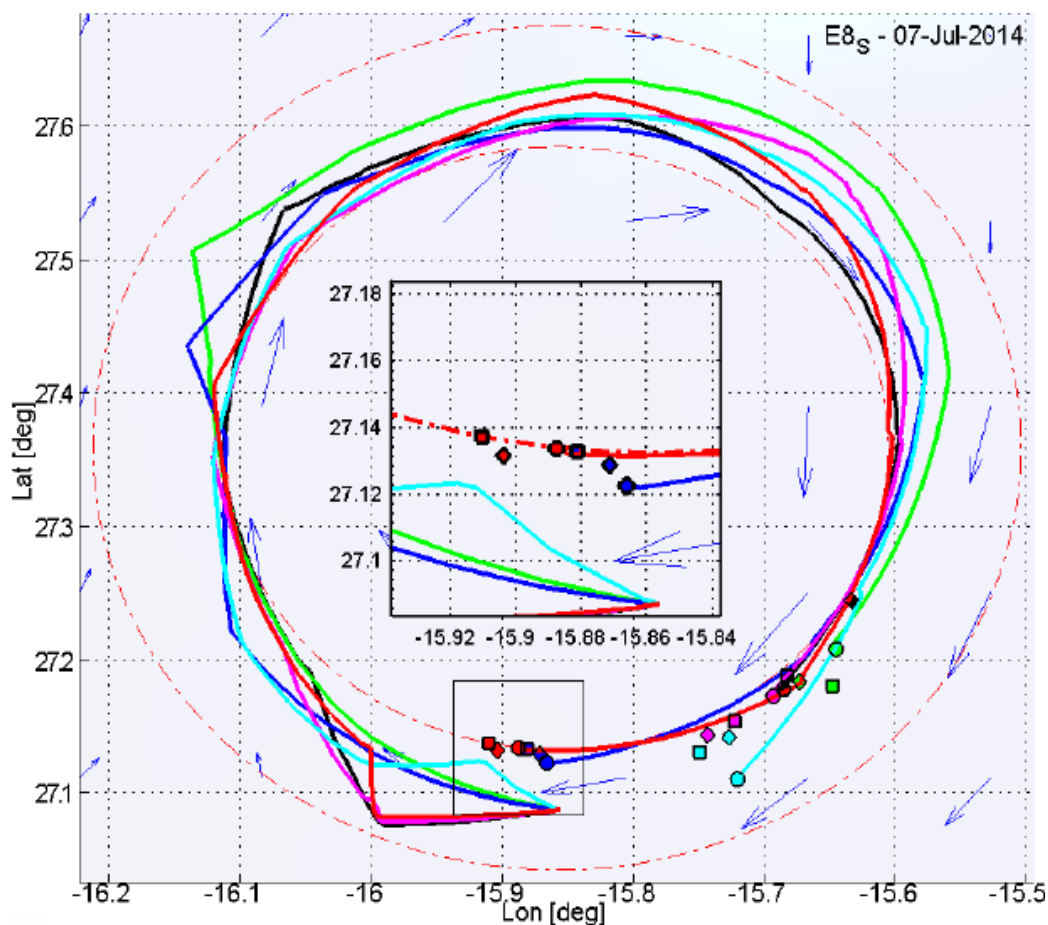


Slika 6.1: Nabor scenarijev UGPP z omejenim koridorjem.

Slika 6.1 v pogledu od zgoraj prikazuje scenarije, definirane v [222]: na levi strani je viden geografski razpon celotne mape, kjer so vstavljeni scenariji; na desni strani sta prikazana primera dveh scenarijev.

Natančneje, za določen koridor je definirana naloga pokrivanja čim večjega kotnega razpona krožnega koridorja (slika 6.2: tukaj so prikazani zemljevidi z morskimi tokovi; slika 6.3: koridor z omejitvami za plovilo; slika 6.4: shematsko prikazano plovilo UGPP, ki mora upoštevati dane omejitve koridorjev). Taki koridorji se pogosto pojavljajo kot naloge pilotov, ki v primeru krožnih (ali drugače nelinearno kompleksnih oblik) zahtevajo pogoste popravke linearnih smeri vozila. Kot že omenjeno, so ti scenariji prostorski (globine do 1000 metrov), a so zaradi kompleksnosti izrisa prikazani le povezani odseki trajektorij, ko se vozilo pojavi na površini. Ker se tokovi časovno spreminjajo, se ob vsaki novi osvežitvi napovedi tokov načeloma lahko spremeni tudi predvidena optimalna trajektorija, kar naloži še dodatno zahtevo pilotom. V takih scenarijih je sistem UGPP toliko bolj obvladljiv, če je izveden s pomočjo avtomatizacije operacijske raziskave logističnih, transportnih in ekonomskih sistemov (angl. *Operational research of logistic, transport, and economic systems*) (ORLTES).

Slika 6.3 prikazuje še dva scenarija, kjer je posebej obarvan del izven



Slika 6.2: Naloga **UGPP** s scenarijem, kjer je koridor omejen. Prikazane so še trajektorije, izračunane z različnimi **EA**.

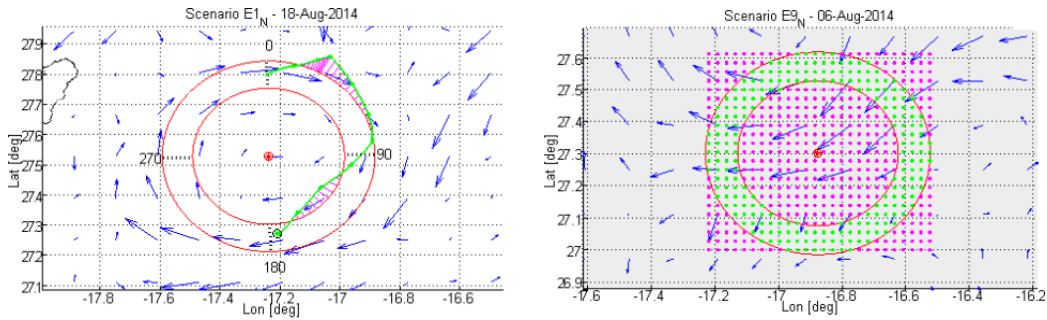
koridorja (violična barva) oz. ponazorjeno vozilo **AUV** v prostorskem koordinatnem sistemu.

Postopek **10** prikazuje združen sistem **ORLTES**, ki ga dobimo z združitvijo **DE** in obravnavo omejitev na scenarijih **UGPP**: vrstice od 11 do 25 izračunajo ustreznost in dopustnost poskusnega vektorja, kar je ovito z zanko za algoritem **DE** z obravnavo omejitev.

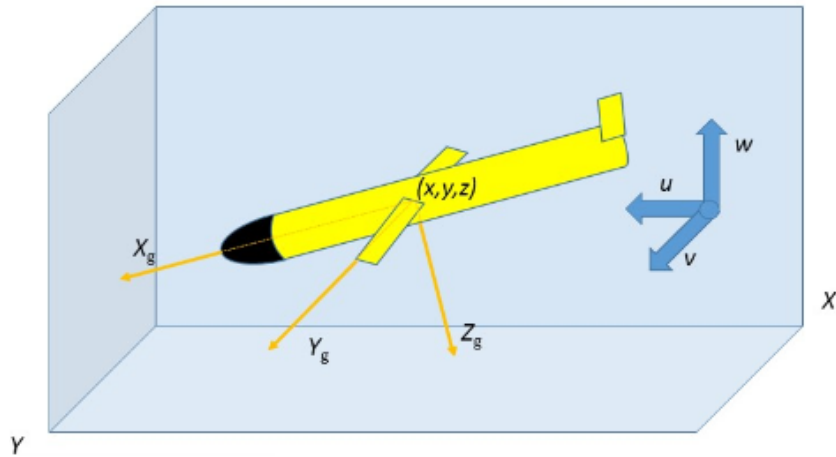
Postopek 10 Postopek za avtonomno navigacijo pri **UGPP** z omejitvami koridorja: CUGPP@DE

1: **postopek** $NP_{dyn} \epsilon_{sum} jDE/best/1/bin@cUGPP(x)$

Vhod: p_0 (trenutna pozicija vozila), zemljevid (MyOcean IBI), $p_{eddy\ center}$ (središče vrtnice kot navidezna točka tarče), L (število točk trajektorije na potop), L' (število



Slika 6.3: Prikaz morskih tokov (modre puščice) na scenariju **UGPP**, kjer je koridor omejen (rdeči krogi), skupaj s prikazom nedopustnih poti (vijolične tangente od koridorja do trajektorije, ki je zelene barve).



Slika 6.4: Plovilo v scenarijih **UGPP**, ki se premika v prostoru po oceanu, predstavljen v različnih koordinatnih sistemih.

točk trajektorije ob lebdenju), MAX_FES (dodeljeno največje število ovrednotenih kriterijske funkcije), NP_{init} (začetna velikost populacije **DE**), NP_{min} (najmanjša velikost populacije **DE**), p_{max} (število zmanjšanj velikosti populacije), F_l , F_u , τ_1 , in τ_2 (konstante za jDE, $\tau_1 = \tau_2 = \tau$ ad hoc), $\eta\%$, g_c (parametri za obravnavo omejitev), r_{min} , r_{max} (radij mej koridorja).

Izhod: \mathbf{x} – seznam instrukcij (D postopnih sprememb smeri) za navigacijo podmorske sonde; prav tako, za možno nadaljnje obravnavanje plana misije, so generirani nekateri pomožni rezultati v evolucijskem zagonu, kot so: 1) čas in datum računanja, 2) vrednost ustreznosti in kršitev omejitev dobljene najboljše rešitve \mathbf{x} , 3) konvergenca po generacijah za: a) najboljšo vrednost ali vrednost z najmanj kršitev omejitev, b) najmanjša vrednost omejitev, c) povprečna vrednost ustreznosti v populaciji in d) standardna deviacija ocene ustreznosti za populacijo.

- 2: uniformno naključno generiral začetno populacijo **DE**, $\mathbf{x}_{i,0}$, $\forall i \in \{1, 2, \dots, NP_{init}\}$;

```

3: inicializiraj metodo obravnave  $\epsilon$ -omejitev:  $\eta = \lfloor \nu\% NP + 1 \rfloor$ ;
4: for DE generacijska zanka  $g$  (dokler  $FES < MAX\_FES$ ) do
5:   for DE iteracijska zanka  $i$  (za vse posameznike  $\mathbf{x}_{i,g}$  v populaciji) do
6:     if populacijo zmanjšamo glede na pogoj  $G_p > \frac{MAX\_FES}{p_{max}NP_p}$  then
7:       zmanjšamo populacijo na pol z uporabo enačbe za selekcijo po indeksih (glej [23]);
8:     end if
9:     DE izračun novega posameznika  $\mathbf{x}_{i,g}$  (prilagajanje, mutacija, križanje):
10:      
$$F_{i,g+1} = \begin{cases} F_l + rand_1 \times F_u & \text{če } rand_2 < \tau_1, \\ F_{i,g} & \text{sicer} \end{cases};$$

11:      
$$CR_{i,g+1} = \begin{cases} rand_3 & \text{če } rand_4 < \tau_2, \\ CR_{i,g} & \text{sicer} \end{cases};$$

12:      
$$\mathbf{v}_{i,g+1} = \mathbf{x}_{r_{best,g}} + F_{i,g+1}(\mathbf{x}_{r_1,g} - \mathbf{x}_{r_2,g});$$

13:      
$$u_{i,j,g+1} = \begin{cases} v_{i,j,g+1} & \text{če } rand(0,1) \leq CR_{i,g+1} \text{ ali } j = j_{rand}, \forall j \in \{1, \dots, D\}; \\ x_{i,j,g} & \text{sicer} \end{cases}$$

14:     DE ovrednotenje ustreznosti in določitev kršitve omejitev (simulacija UGPP):
15:      $b_0 = \text{bearing}(\mathbf{p}_0, \mathbf{p}_{eddy \text{ center}});$ 
16:     for  $j = \{1, 2, \dots, D\}$ :
17:        $b_j = b_{j-1} + u_{i,j,g+1};$ 
18:        $\{\mathbf{t}_l^j; \forall l = 1, 2, \dots, L\} = \text{simulate\_stint}(\mathbf{t}_{L+L'}^{j-1}, b_j, \text{map}),$  kjer je  $\mathbf{p}_0$ 
19:       uporabljen kot  $\mathbf{t}_{L+L'}^0$ ;
20:        $\{\mathbf{t}_l^j; \forall l = L+1, L+2, \dots, L+L'\} = \text{simulate\_drift}(\mathbf{t}_L^j, \text{map});$ 
21:     end for
22:     for  $k = \{1, 2, \dots, K\}$  kjer  $K = D(L+L')$  in  $\forall \mathbf{t}_k \in \{\mathbf{t}_l^j\}, \forall j = 1, 2, \dots, D, \forall l =$ 
23:      $1, 2, \dots, L+L'$ :
24:        $\mathbf{R}t_k = \text{deg2utmCan}(\mathbf{t}_k) - \text{deg2utmCan}(\mathbf{p}_{eddy \text{ center}});$ 
25:        $\gamma_k = \text{atan2}(\mathbf{R}t_{k,y}, \mathbf{R}t_{k,x}), r_k = \sqrt{\mathbf{R}t_{k,x}^2 + \mathbf{R}t_{k,y}^2};$ 
26:       
$$\alpha_k^{\text{cum}} = \begin{cases} 0, & k = 1, \\ \alpha_{k-1}^{\text{cum}} + (\gamma_k - \gamma_{k-1}) & \forall k = 2, 3, \dots, K; \end{cases};$$

27:       
$$\nu_k = \begin{cases} 0, & r_k > r_{\min} \wedge r_k < r_{\max}; \\ \max(r_{\min} - r_k, r_k - r_{\max}), & \text{sicer} \end{cases};$$

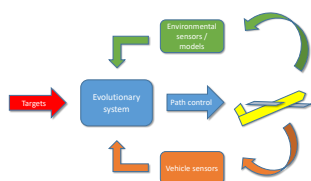
28:     end for
29:      $f(\mathbf{u}_{i,g+1}) = 2\pi - \left( \max_{k=1}^K(\alpha_k^{\text{cum}}) - \min_{k=1}^K(\alpha_k^{\text{cum}}) \right);$  FES++;
30:      $\bar{\nu}_{u,g+1} = \sum_{k=1}^K \nu_k;$ 
31:     DE selekcija: uporabi primerjavo z upoštevanjem omejitev [222]; posodobi še
32:      $\mathbf{x}_{best}$ , če je izboljššan:
33:   end for
34:   prilagodi parametre metode  $\epsilon$ -omejitev s ponovnim izračunom nivoja  $\eta$ ;
35: end for
36: vrni najboljši najden posameznik  $\mathbf{x}_{best}$ ;

```

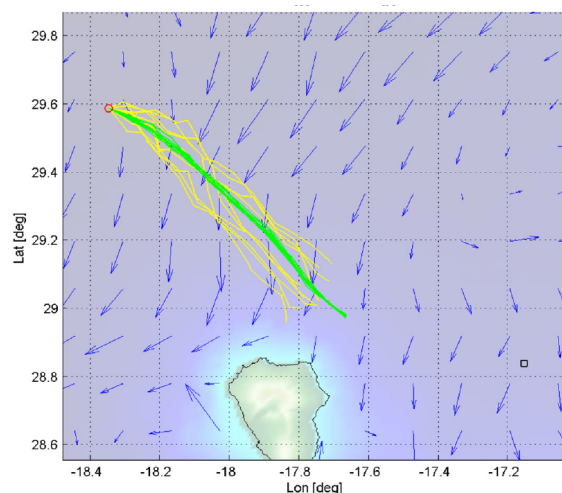
Poglavje 7

Robustnost na napake med misijami in zanesljivost transporta

Zbrano gradivo, ki je zajeto v tem poglavju, je prevedeno iz prispevka:



Aleš Zamuda and José Daniel Hernández Sosa. **Success history applied to expert system for underwater glider path planning using differential evolution.** *Expert Systems with Applications*, 2019, vol. 119, pp. 155–170. <http://dx.doi.org/10.1016/j.eswa.2018.10.048>



Slika 7.1: Izboljšava robustnosti: primerjava trajektorij dveh algoritmov, prvi (rumena barva) vrača precej različne rezultate, drugi (zelena barva) robustno in bolj učinkovito vrača boljšo trajektorijo.

Pri planiranju misij se zaradi prevelike kompleksnosti modeliranih sistemov in s tem neskladnosti med modelom in realnim delovanjem narave pojavljajo razhajanja. V primeru, da je teh manj, jih lahko imenujemo izjeme ali napake. Da bi omejili obravnavo izjem in napak in s tem povečali robustnost planiranja ali avtomatizacije ter s tem povečali zmožnost avtonomnosti robota oz. vozila, lahko poskušamo na primer pripraviti čim boljši optimizacijski pristop, ki omogoča hitri odziv in popravljanje v primeru izjem in napak [231].

V primeru, ko želimo izboljšati kakovost optimizirane trajektorije na podlagi obstoječih vhodnih vremenskih in drugih podatkov, lahko poskusimo uporabiti čim boljši optimizacijski algoritem. Avtorji [115] za inteligentne sisteme UGPP so letu 2017 zatrdili, da so **oceanološki znanstveniki sanjali o revoluciji v oceanskem opazovanju skozi avtonomne platforme in senzorje, ki sedaj postaja realnost.**

To problematiko nadalje obravnavajo publikacije [223, 229]. Glede robustnosti in stabilizacije značilik pri rangiranju kriterijev pa lahko preberemo v [226]. Kot primer je na sliki 7.1 prikazana primerjava dveh algoritmov (zagon enega v rumeni barvi, drugega v zeleni), kjer algoritem v zeleni barvi daje precej bolj stabilne (robustne) rezultate — trajektorije so ne le boljše ocenjene (bližje tarči), ampak tudi manj odstopajo od optimalne poti.

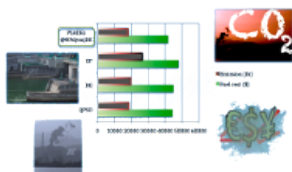
Ponovno pa omenimo, da ker se tokovi časovno spreminjajo, se ob vsaki novi osvežitvi napovedi tokov načeloma lahko spremeni tudi predvidena optimalna trajektorija. To naloži še dodatno zahtevo pilotom, ki lahko morda pilotirajo tudi celotno floto plovil in imajo precej omejen čas planiranja. V takih scenarijih je sistem UGPP toliko bolj obvladljiv, če je izveden s pomočjo robustne avtomatizacije ORLTES, ki v primeru pilotovega neodzivanja sam vozilu pošlje nova navodila. To je še posebej uporabno v primeru krožnih koridorjev, kjer bi sicer vozilo z vožnjo v ravni smeri (smeri b_0 , glej vrstico 12 v algoritmu 10) pogosto bilo izven zadanega dopustnega območja.

V tem poglavju je bila predstavljena robustnost ORLTES s pomočjo primera UGPP, v naslednjem poglavju sledi obravnavo sistemov sistemov in s tem povezanih ORLTES in ekonomsko planiranje.

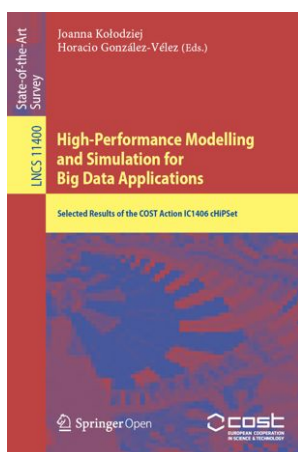
Poglavje 8

Energetska avtonomnost in algoritmi za ekonomsko planiranje v sistemih sistemov

Za planiranje avtonomnosti določenega energetskega sistema potrebujemo dober napovedni model. Primer obravnave modelov za avtonomnost najdemo na primer v naslednjih prispevkih (v prvem, avtonomnost energetskega sistema za delovanje sistema elektrarn, v drugem, primer avtonomnega finančnega planiranja za kapitalske portfelje):



A. Glotić, A. Zamuda. **Short-term combined economic and emission hydrothermal optimization by surrogate differential evolution**. *Applied Energy*, 1 March 2015, vol. 141, pp. 42-56. <http://dx.doi.org/10.1016/j.apenergy.2014.12.020>



A. Zamuda, V. Crescimanna, J. C. Burguillo, J. Matos Dias, K. Wegrzyn-Wolska, I. Rached, H. González-Vélez, R. Senkerik, C. Pop, T. Cioara, I. Salomie, A. Bracciali. **Forecasting Cryptocurrency Value by Sentiment Analysis: An HPC-Oriented Survey of the State-of-the-Art in the Cloud Era**. Kołodziej J., González-Vélez H. (eds) *High-Performance Modelling and Simulation for Big Data Applications*. *Lecture Notes in Computer Science*, vol 11400, 2019, pp. 325-349. http://dx.doi.org/10.1007/978-3-030-16272-6_12

V tem poglavju bo posebej predstavljen eden od pristopov za optimizacijo razporejanja proizvodnje električne energije med hidroelektrarnami in termoelektrarnami [71], ki upošteva okoljske emisije in ekonomsko ceno pri proizvodnji električne energije. Pristop omogoča hitrejši izračun kot je to bilo možno do pred tem. To je mogoče, saj pristop predstavi pred-izračunan nadomestni model, ki se med optimizacijo celotnega optimiziranega modela hidro in termo enot ne spreminja in hrani dobljene vrednosti parametrov nadomestnega modela v praktično uporabni natančnosti.

Pristop [71] sestoji iz dveh stopenj, dveh algoritmov, ki sta predstavljeni v nadaljevanju: prvi algoritem, [samoprilagodljiva diferencialna evolucija z zmanjševanjem velikosti populacije in obravnavo omejitev s prilagajanjem \$\varepsilon\$ nivoja](#) (angl. *self-adaptative differential evolution with population size reduction and constraint handling using ε level adjustment*) (NPdyejDE), naslavlja posebno obravnavo omejitev in optimizira razporejanje za termoelektrarne; drugi algoritem, [paralelna samoprilagodljiva diferencialna evolucija z nadomestnim modelom](#) (angl. *Surrogate Parallel Self-adaptive Differential Evolution*) (PSADEs) [71], uporabi rezultate prvega algoritma, da optimizira skupno proizvodnjo s hidro elektrarnami, pri tem pa oba uporabita praktično natančnost parametrov za urnik obremenitev termoelektrarn. To izboljša učinkovitost produkcije električne energije in zmanjšanja izpustov ter ogljičnega odtisa, s hkratno zadovoljitvijo 24-urnih sistemskih zahtev v razporejanju povpraševane energije ter vseh operativnih zahtev.

Ker je za ta pristop poglavitno razumevanje matrike za praktično natančnost, bo ta razložena v nadaljevanju skozi algoritem NPdyejDE, ki to matriko zgradi.

Pristop razdelitve ocenitvene funkcije HTS v NPdyejDE deluje tako, da razdeli to funkcijo na dva dela (dvonivojsko), po principu proizvajalec-potrošnik, kjer NPdyejDE proizvede matriko, ki jo uporabimo v PSADEs za realno-časovni del pristopa HTS [71]. Za izračunavanje skupne ocenitvene funkcije se namreč potem za termoelektrarne uporabijo vrednosti posameznih TPP razporedov, ki so shranjeni v predizračunani matriki vrednosti za vsako od elektrarn:

$$\mathbf{s}_{pSD}^{\text{case}} = \{\mathbf{s}_x^{\text{case}}, \mathbf{s}_{TC}^{\text{case}}, \mathbf{s}_{TE}^{\text{case}}, \mathbf{s}_{p_{s1}}^{\text{case}}, \mathbf{s}_{p_{s2}}^{\text{case}}, \mathbf{s}_{p_{s3}}^{\text{case}}\}, \quad (8.1)$$

ki deluje kot nadomestni model za termalni del razporejanja proizvodnje električne energije.

Ker je TPP urno neodvisen, je potrebno enkrat izračunati te vrednosti dela TPP in jih shraniti v vektor \mathbf{s} , ki ga združimo v matriko za različne vrednosti moči s korakom natančnosti 0,01 MW. Vpogled v vrednosti razporeda

za določeno obremenitev **TPP** je tako s pomočjo take matrike zelo hitra in odpravi celotni del zagona optimizacije v delu za realno-časovni del.

Za obravnavo omejitev **NPdynεjDE** uporablja metode iz [18, 214, 30]. Parametri, ki so pri tem uporabljeni, so $D = 3$, $NP_{\text{init}} = 1000$, $NP_{\text{min}} = 100$, $G_c = 500$, $C_z = 5$, $\nu = 0, 2NP$ in $pmax = 4$.

Ker za iskalne parametre velja $x_1, x_2, x_3 \in [0, 1]$ in so so meje za pripadajoče spodnje in zgornje in vrednosti referenčnih moči za prvo, drugo in tretjo termoelektrarno v tem modelu nastavljene kot $P_1 \in [20, 175]$, $P_2 \in [40, 300]$ in $P_3 \in [50, 500]$, je kodiranje fenotipa **DE** za te termoelektrarne naslednje:

$$P_1 = 155x_1 + 20, \quad (8.2)$$

$$P_2 = 260x_2 + 40, \quad (8.3)$$

$$P_3 = 450x_3 + 50. \quad (8.4)$$

Združen psevdokod algoritma za predizračun nadomestnega modela za termalno urno planiranje z uporabo **NPdynεjDE** prikazuje postopek 11: v vrstici 11 najprej shranjene vrednosti pri ovrednotenju potem trajno shranimo v izhodno nadomestno matriko modela v vrstici 18. Ostali deli algoritma so podobni obravnavi omejitev za **UGPP**.

V tem poglavju je bil predstavljen primer združevanja dveh sistemov postopkov za **ORLTES** v okolju energetike in s tem povezano **CI** za ekonomsko planiranje. Drug primer ekonomskega planiranja lahko zasledimo npr. tudi v trgovalnih sistemih, več o tem v preglednem prispevku [230]. V naslednjem poglavju je predstavljena primerjava kakovosti algoritmov, kjer je omenjen tudi doprinos z rezultati pristopa, predstavljenega v tem poglavju.

Postopek 11 Postopek HTS NPdyn ϵ jDE za izračun nadomestne matrike v dvonivojskem ekonomskem planiranju sistema sistemov

Vhod: $P_{s1,\min}, P_{s1,\max}, P_{s2,\min}, P_{s2,\max}, P_{s3,\min}, P_{s3,\max}$ — tri meje referenčne moči TPP; $N_{\max\text{Feval}}$ — največje število dodeljenih ovrednotenj kriterijske funkcije; NP_{init} — začetna velikost populacije DE; NP_{\min} — najmanjša velikost populacije DE; z_{\max} — število zmanjšanj velikosti populacije; F_l, F_u, τ_a , and τ_b — jDE konstante; $\alpha_1, \alpha_2, \beta, \eta, C_z, G_c$ — parametri obravnave omejitev; dig — interval natančnosti (resolucija) za predizračunano matriko urnih termalnih modelov razporejanja;

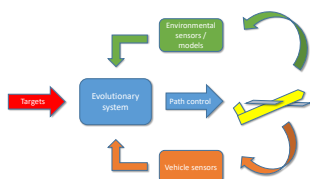
Izhod: $\mathbf{S}_{\text{PSD}}^{\text{case}}$ — matrika urnih modelov razporejanja termoelektrarn; (sestavljena iz po dig korakov natančne referenčne moči, združuje: neskalirane vrednosti DE vektorjev $\{\mathbf{s}_x^{\text{case}},$ skupni stroški $\mathbf{s}_{TC}^{\text{case}},$ skupne emisije $\mathbf{s}_{TE}^{\text{case}},$ moč, ki jo ustvari prva TPP, $\mathbf{s}_{ps1}^{\text{case}},$ moč, ki jo ustvari druga TPP, $\mathbf{s}_{ps2}^{\text{case}}$ in moč, ki jo ustvari tretja TPP, $\mathbf{s}_{ps3}^{\text{case}}\}$)

- 1: **for** izračunaj vse $dig'_l, \forall l \in \{1, \dots, n_m\}$ z natančnostnim korakom dig s pomočjo enačbe (30) iz [71] **do**
 - 2: $NP = NP_{\text{init}};$
 - 3: uniformno naključno generiraj začetno populacijo $\mathbf{x}_{i,0}, \forall i \in \{1, \dots, NP\};$
 - 4: inicializiraj metodo za obravnavo ϵ -omejitev z enačbama (24) in (25) iz [71];
 - 5: **for** DE generacijska zanka $G = 1$ (dokler $\text{FES} < N_{\max\text{Feval}}$) **do**
 - 6: **if** naj bo populacija zmanjšana glede na enačbo (17) iz [71] **then**
 - 7: zmanjšaj populacijo na polovico po enačbi (23) iz [71];
 - 8: **end if**
 - 9: **for** DE iteracijska zanka i (za vse posameznike $\mathbf{x}_{i,G}$ v populaciji) **do**
 - 10: DE izračun novega posameznika $\mathbf{u}_{i,G}$ (prilagajanje, mutacija, križanje): iz [71] uporabimo enačbi (15) in (16), nato še (12) in (13);
 - 11: DE ovrednotenje ustreznosti $\mathbf{u}_{i,G}$: iz [71] uporabimo enačbe (9), (11) in (32)–(34), shranimo njihove vrednosti;
 - 12: DE selekcija: izračunaj in ohrani ϵ -omejitev ν za $\mathbf{u}_{i,G}$ in nato iz [71] uporabi enačbo (23) za propagacijo;
 - 13: **end for**
 - 14: $G = G + 1;$
 - 15: prilagodi parametre metode za obravnavo ϵ -omejitev: ponovno izračunaj β , nato iz [71] uporabi enačbi (26) in (27);
 - 16: **end for**
 - 17: \mathbf{x}^* =najboljši posameznik (in vse ohranjene vrednosti iz koraka 11) dobljene iz $\{\mathbf{x}_{i,G}\}, i = 1, \dots, NP;$
 - 18: shrani \mathbf{x}^* na indeksu (po natančnosti) l v $\mathbf{S}_{\text{PSD}}^{\text{case}}: \mathbf{S}_{\text{PSD}}^{\text{case}}(l) := \mathbf{x}^*;$
 - 19: **end for**
 - 20: **vrni** $\mathbf{S}_{\text{PSD}}^{\text{case}};$
-

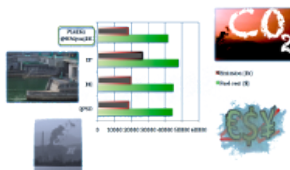
Poglavje 9

Primerjava kakovosti algoritmov za operacijske raziskave logističnih, transportnih in ekonomskih sistemov

Zbrano gradivo, ki je zajeto v tem poglavju, je prevedeno iz prispevkov:

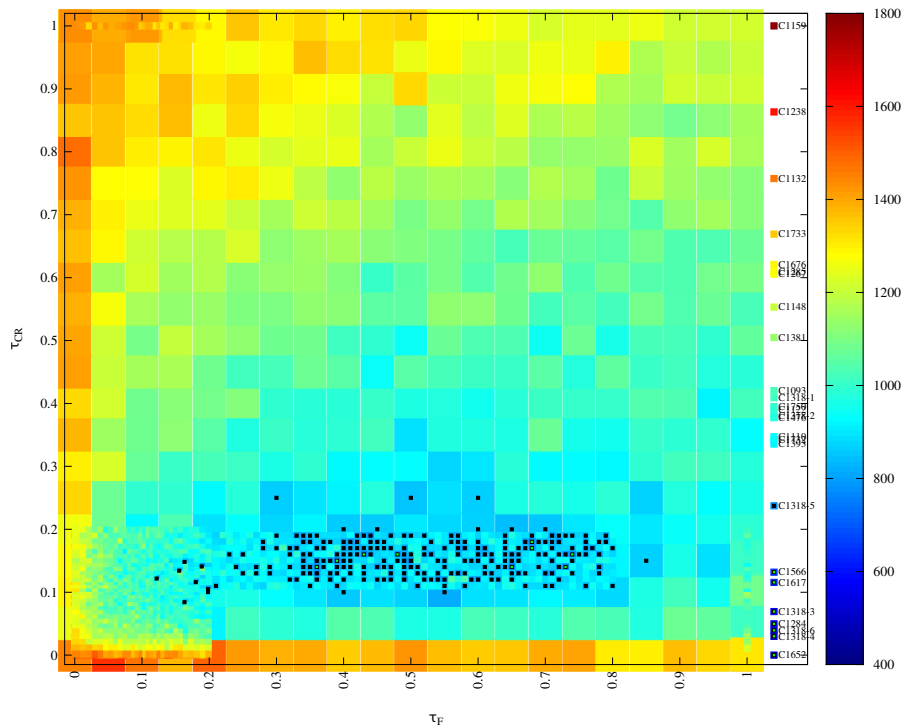


Aleš Zamuda and José Daniel Hernández Sosa. **Success history applied to expert system for underwater glider path planning using differential evolution.** *Expert Systems with Applications*, 2019, vol. 119, pp. 155–170. <http://dx.doi.org/10.1016/j.eswa.2018.10.048>



A. Glotić, A. Zamuda. **Short-term combined economic and emission hydrothermal optimization by surrogate differential evolution.** *Applied Energy*, 1 March 2015, vol. 141, pp. 42-56. <http://dx.doi.org/10.1016/j.apenergy.2014.12.020>

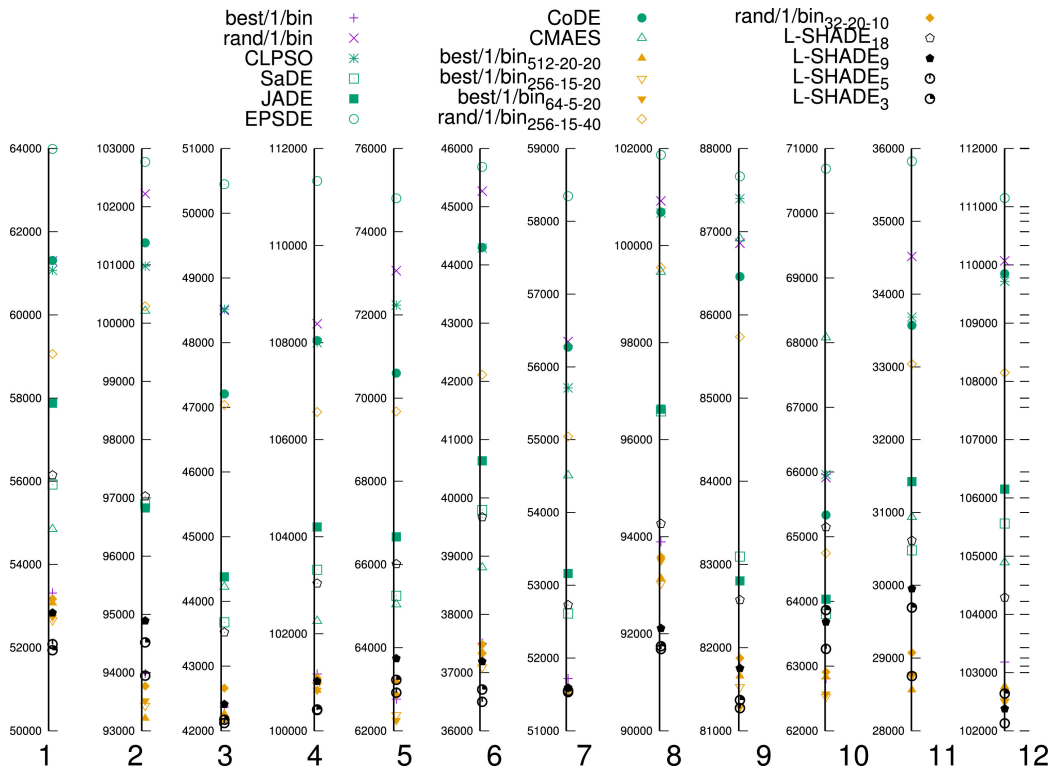
Za oceno kakovosti optimizacijskih algoritmov so bile predlagane že številne metrike [103, 78], zato je večina obstoječih študij in primerjav algoritmov zasnovanih na specifičnih testnih meritvah. Te študije so največkrat statistične, izvedene nad izbranimi testnimi problemi, ki jih algoritem skuša rešiti. Pri primerjavi velja posebno pozornost nameniti definiciji primerjalne metrike oz. testiranju hipoteze med različnimi rezultati posameznih vzorčenj. Primer agregacije s kvadriranjem (angl. *power law*) je Friedmanov test [68], primer ploskega (linearnega) primerjanja razlik pa na primer šahovski rangirni sistem [184].



Slika 9.1: Zgled primera za primerjavo algoritmov [ORLTES](#). Vizualizacija večkratnega rangiranja algoritmov iz primera optimizacijskih funkcij pod različnimi nastavitvami parametrov za določen algoritem (več v [209]).

Poleg izbora načina agregacije moramo v snovanju okolja za primerjavo kakovosti algoritmov [ORLTES](#) nameniti še posebno pozornost izboru dveh zelo pomembnih vhodnih podatkov za optimizacijski algoritem: 1) seme naključnega generatorja in 2) določitev izbrane ocenitvene funkcije. Namreč, ker gre za stohastične algoritme, moramo testirati več neodvisnih zagonov algoritma za isto posamezno testno ocenitveno funkcijo. Poleg tega, pa ker ob stohastičnosti algoritmov za optimizacijo želimo te uporabiti ne le na eni posamezni ocenitveni funkciji v izbrani domeni, moramo zato določiti več reprezentativnih testnih ocenitvenih funkcij za to določeno domeno.

Če dobro zastavimo skupek konkretnih primerov testnih funkcij, bo izdelan pristop [ORLTES](#) toliko bolj uporaben za celotno izbrano domeno v [ORLTES](#). Pri izbiri nabora testnih funkcij gre za izbiro pravih značilik za učenje. Z dobro izbiro želimo doseči stabilnost tega izbora na celotno domeno. Obravnava algoritmov za stabilnost presega to učno gradivo, omenim lahko le, da je možno tudi algoritme za izbiro stabilnosti učiti s pomočjo izkušenj snovanja [ORLTES](#), kot je na primer [226].



Slika 9.2: Vizualizacija primerjave rezultatov optimizacije trajektorij [231].

Kot enostavnejši primer primerjave algoritmov za **ORLTES** je rangiranje algoritmov glede na testne funkcije iz **CEC**. Na sliki 9.1 je zato podan primer rangiranja več kot 2000 algoritmov po metodi na podlagi Friedmanovega testa [68] iz prispevka [209]. Vendar pa za celostni doprinos v **ORLTES** to ni dovolj — tak algoritem ali verzijo algoritma je potrebno vključiti v okolje, ki vsebuje tudi druge dele **ORLTES**, kot je na primer **UGPP** ali **HTS**, kot je zapisano v nadaljevanju tega poglavja.

Primerjava v okolju **UGPP**

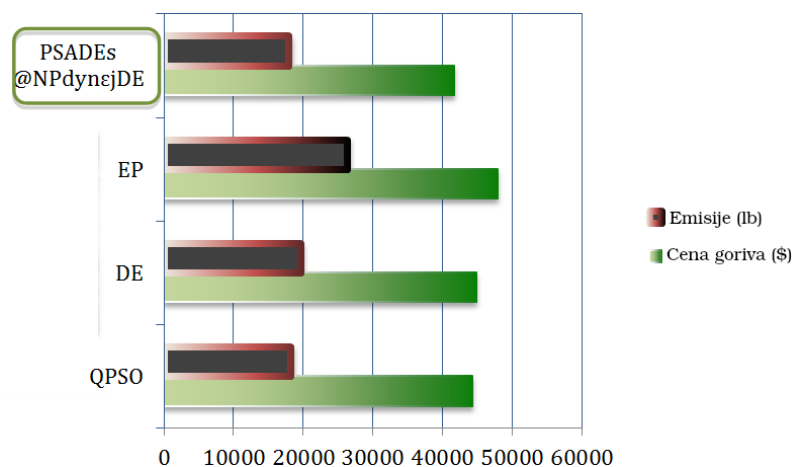
Primer nedavne agregacije (zbor) rezultatov iz različnih scenarijev **UGPP** podaja slika 9.2, o kateri lahko več preberemo v [231], kot zanimivost pa omenimo, da so dobljene razlike pri generiranih trajektorijah v dolžinah tudi po več kilometrov.

Za primerjavo z neparametričnimi testi nad algoritmi so v tabeli 9.1 primerjani različni algoritmi **DE** in **EA** z več različnimi začetnimi velikostmi populacije (NP_{init}). Za izračun testov je v [231] uporabljena koda iz [http:](http://)

[//sci2s.ugr.es/keel/multipleTest.zip](http://sci2s.ugr.es/keel/multipleTest.zip), ki izračunava Friedman rangiranje [53]. V [231] podana uvrstitev prikazuje rangiranje po zbirni statistiki relativnih zmogljivosti algoritmov (nižja vrednost ranga označuje boljše delovanje algoritma), pri meji intervala zaupanja $\alpha = 0,05$ in statistični vrednosti Friedman testa 4776,1. Indeksne oznaki, ki so podpisane pod imeni, označujejo velikost populacije, ki je bila sicer drugače privzeto postavljena na 100. V tabeli 9.1 so podane le najzanimivejše kombinacije, pri drugih je bila uspešnost slabša od objavljenega L-SHADE₅ [231]. Tabela 9.1 nato poroča post-hoc hipoteze o teh Friedmanovih statistikah, kjer post-hoc postopki zavračajo hipoteze iste uspešnosti, t.j. najboljši algoritem L-SHADE₅ prekaša kakšen drug algoritem pri posamezni p_{val} -vrednosti: pri Bonferroni-Dunn, $\leq 0,000909$, Holm, $\leq 0,00714$, Hochberg, $\leq 0,00625$, Hommel, $\leq 0,00714$, Holland, $\leq 0,0073$, Rom, $\leq 0,00657$, Finner, $\leq 0,0447$ in Li, $\leq 0,0288$.

Kot lahko rečemo o splošni uspešnosti na tem testnem naboru v tabeli 9.1, večina post-hoc testov potrjuje, da predlagani L-SHADE₅ presega vse algoritme po uvrstitvi 7 v vseh različicah, kot so JADE, evolucijska strategija s prilagoditvijo kovarijančne matrike (angl. *Covariance Matrix Adaptation Evolution Strategy*) (CMA-ES), diferencialna evolucija s kompozitnimi strategijami generiranja poskusnih vektorjev in krmilnih parametrov (angl. *Differential evolution with composite trial vector generation strategies and control parameters*) (CoDE) in algoritem diferencialne evolucije z ansamblom parametrov in mutacijskih in križnih strategij (angl. *Differential Evolution Algorithm with Ensemble of Parameters and Mutation and Crossover Strategies*) (EPSDE) ter nekaterih različicah, kot je DynNP jDE in diferencialna evolucija, ki se sama prilagaja (angl. *Self-adaptive Differential Evolution*) (SaDE). Ko spremenimo začetno populacijo za algoritme na drugačno vrednost kot 100, opazimo v tabeli 9.1 večje izboljšanje rezultatov za SaDE₃₂ z $NP = 32$ v primerjavi s SaDE z $NP = 100$. Podobno lahko opazimo pri JADE, CoDE in EPSDE, ne pa tudi za CLPSO ali CMA-ES, kar daje še en vpogled v primerjavo zmogljivosti in konfiguracijo teh algoritmov za UGPP.

Za SaDE z $NP = 32$ v tabeli 9.1 post-hoc postopki ne pokažejo pomembnih izboljšav, vendar je na lestvici še vedno slabši od L-SHADE₅. To kaže, da s spremembo velikosti populacije dobimo tudi boljše izvedeno različico SaDE, ki je na tem primeru konkurenčna v [231] predlaganemu algoritmu L-SHADE₅, medtem ko je prvotna različica L-SHADE z $NP = 100$ [220] prav tako izboljšana z L-SHADE₅ za vse obravnavane scenarije UGPP v [231]. Natančneje primerjajmo pare prikazanih primerkov konfiguriranih algoritmov in rezultatov, o katerih so poročali v literaturi, to so tudi nove izboljšave, ki jih vidimo v tabeli 9.1: L-SHADE₅ izboljšuje razvrstitev v primerjavi s prej najboljšimi algoritmi best/1/bin (brez spreminjanja velikosti populacije) in best/1/bin₆₄₋₅₋₂₀ (z zmanjšanjem velikosti populacije), SaDE₃₂ v primer-



Slika 9.3: Primerjava rezultatov ekonomskega in okoljskega planiranja v proizvodnji električne energije s pristopom z DE [71].

javi s SaDE, JADE₃₂ v primerjavi z JADE, CoDE₃₂ v primerjavi s CoDE, EPSDE₃₂ v primerjavi z EPSDE in kot zmanjšanje velikosti populacije v primerjavi s fiksnimi pristopi na osnovi jDE tudi best/1/bin₆₄₋₅₋₂₀ v primerjavi z best/1/bin oz. rand/1/bin₃₂₋₂₀₋₁₀ v primerjavi z rand/1/bin.

Učinkovitost L-SHADE₅ pri uporabi za UGPP se kaže v podpiranju predlogov izboljševanja ORLTES UGPP, kot opisano v [115]: nove primerjave in ovrednotenja izboljšanih EA, ki pomaga pri premagovanju velike računske zahtevnosti. Z izboljšavo trajektorij za misije se namreč pojavijo nove zmožnosti, ki jih planerji misij lahko uporabijo.

Če natančneje pojasnimo prispevek prikazane primerjave kakovosti algoritmov ORLTES na splošno, povejmo, da je izbira verzije algoritma za načrtovanje prave poti za ekspertni sistem v pomoč strokovnjakom ORLTES zelo pomembna pri obravnavi zapletenih in potencialno tveganih situacij, kot so to med prikazanimi realnimi misijami — npr. v misijah [231] so bili prisotni močni tokovi v območju za prevzem vozila v bližini obale. Računsko intenzivni postopki simulacije in optimizacije so potrebni za zagotovitev čim več dokazov za sprejemanje odločitev, zato lahko majhne razlike v predlagani poti povzročijo pomembne spremembe poti UGPP [231].

Pri izračunavanju računsko intenzivnih postopkov si lahko pomagamo tudi s superračunalniki [230], kar daje temu okolju ORLTES še dodatne raziskovalne možnosti. Zaradi stroškovno cenovnih izzivov, s katerimi se je potrebno soočiti pri pripravi projektov za misije, pa ponuja okolje razvoja sistemov ORLTES za planiranje poti podmorskih sond (angl. *Underwater Glider Path Planning*) (UGPP) tudi izzive za pridobivanje podpornikov iz strani financirjev in mednarodnega sodelovanja.

Tabela 9.1: Primerjava s post-hoc postopki nad Friedmanovo statistiko (referenčni najbolje rangiran algoritem z rangom R_0 je algoritem **L-SHADE**₅).

Rang (<i>i</i>)	Algoritem	$z = (R_0 - R_i)/SE$	p_{val}	H/H/H	Holland	Rom	Finner	Li
(0)	(L-SHADE ₅)							
1	SaDE ₃₂	0,752	0,452	0,05	0,05	0,05	0,05	0,05
2	L-SHADE ₃	0,934	0,35	0,025	0,0253	0,025	0,0491	0,0288
3	best/1/bin ₆₄₋₅₋₂₀	0,974	0,33	0,0167	0,017	0,0167	0,0482	0,0288
4	best/1/bin ₂₅₆₋₁₅₋₂₀	1,05	0,292	0,0125	0,0127	0,0131	0,0473	0,0288
5	best/1/bin ₆₄	1,37	0,172	0,01	0,0102	0,0105	0,0464	0,0288
6	best/1/bin ₃₂	1,42	0,157	0,00833	0,00851	0,00876	0,0456	0,0288
7	best/1/bin ₅₁₂₋₂₀₋₂₀	1,63	0,104	0,00714	0,0073	0,00751	0,0447	0,0288
8	L-SHADE ₉	2,79	0,00527	0,00625	0,00639	0,00657	0,0438	0,0288
9	best/1/bin	2,85	0,00438	0,00556	0,00568	0,00584	0,0429	0,0288
10	rand/1/bin ₃₂₋₂₀₋₁₀	2,88	0,00396	0,005	0,00512	0,00526	0,042	0,0288
11	best/1/bin ₁₂₈	3,69	0,000222	0,00455	0,00465	0,00478	0,0411	0,0288
12	JADE ₃₂	3,98	6,96e-05	0,00417	0,00427	0,00438	0,0402	0,0288
13	CMA-ES	4,61	4,09e-06	0,00385	0,00394	0,00405	0,0393	0,0288
14	CMA-ES ₃₂	4,61	4,09e-06	0,00357	0,00366	0,00376	0,0384	0,0288
15	CMA-ES ₆₄	4,61	4,09e-06	0,00333	0,00341	0,00351	0,0375	0,0288
16	CMA-ES ₁₂₈	4,61	4,09e-06	0,00313	0,0032	0,00329	0,0366	0,0288
17	CMA-ES ₂₅₆	4,61	4,09e-06	0,00294	0,00301	0,00309	0,0357	0,0288
18	CMA-ES ₅₁₂	4,61	4,09e-06	0,00278	0,00285	0,00292	0,0348	0,0288
19	SaDE ₆₄	5,94	2,91e-09	0,00263	0,0027	0,00277	0,0339	0,0288
20	best/1/bin ₂₅₆	6,43	1,26e-10	0,0025	0,00256	0,00263	0,033	0,0288
21	JADE ₆₄	7,65	1,94e-14	0,00238	0,00244	0,0025	0,0321	0,0288
22	L-SHADE ₁₈	7,99	1,3e-15	0,00227	0,00233	0,00239	0,0312	0,0288
23	SaDE	8,09	6,18e-16	0,00217	0,00223	0,00229	0,0303	0,0288
24	rand/1/bin ₃₂	8,45	2,82e-17	0,00208	0,00213	0,00219	0,0294	0,0288
25	best/1/bin ₅₁₂	8,75	2,12e-18	0,002	0,00205	0,0021	0,0285	0,0288
26	CoDE ₃₂	8,78	1,6e-18	0,00192	0,00197	0,00202	0,0276	0,0288
27	SaDE ₁₂₈	9,13	7,06e-20	0,00185	0,0019	0,00195	0,0267	0,0288
28	JADE	9,36	7,96e-21	0,00179	0,00183	0,00188	0,0258	0,0288
29	JADE ₁₂₈	10,1	4,88e-24	0,00172	0,00177	0,00181	0,0249	0,0288
30	JADE ₂₅₆	12,1	6,44e-34	0,00167	0,00171	0,00175	0,024	0,0288
31	SaDE ₂₅₆	13,2	1,01e-39	0,00161	0,00165	0,0017	0,023	0,0288
32	rand/1/bin ₂₅₆₋₁₅₋₄₀	13,6	5,62e-42	0,00156	0,0016	0,00164	0,0221	0,0288
33	CoDE ₆₄	13,6	2,63e-42	0,00152	0,00155	0,00159	0,0212	0,0288
34	EPSDE ₃₂	14,1	7,68e-45	0,00147	0,00151	0,00155	0,0203	0,0288
35	SaDE ₅₁₂	14,1	3,92e-45	0,00143	0,00146	0,0015	0,0194	0,0288
36	rand/1/bin ₆₄	14,1	2,8e-45	0,00139	0,00142	0,00146	0,0185	0,0288
37	JADE ₅₁₂	14,1	2,06e-45	0,00135	0,00139	0,00142	0,0176	0,0288
38	CoDE	16	5,82e-58	0,00132	0,00135	0,00138	0,0166	0,0288
39	CLPSO	16,5	3,67e-61	0,00128	0,00131	0,00135	0,0157	0,0288
40	CLPSO ₃₂	16,5	3,67e-61	0,00125	0,00128	0,00132	0,0148	0,0288
41	CLPSO ₆₄	16,5	3,67e-61	0,00122	0,00125	0,00128	0,0139	0,0288
42	CLPSO ₁₂₈	16,5	3,67e-61	0,00119	0,00122	0,00125	0,013	0,0288
43	CLPSO ₅₁₂	16,5	3,67e-61	0,00116	0,00119	0,00122	0,0121	0,0288
44	rand/1/bin	17,2	2,91e-66	0,00114	0,00117	0,0012	0,0111	0,0288
45	CoDE ₁₂₈	17,3	7,41e-67	0,00111	0,00114	0,00117	0,0102	0,0288
46	CoDE ₂₅₆	17,7	2,6e-70	0,00109	0,00111	0,00114	0,00928	0,0288
47	rand/1/bin ₁₂₈	17,7	2,03e-70	0,00106	0,00109	0,00112	0,00836	0,0288
48	EPSDE ₆₄	18,6	8,13e-77	0,00104	0,00107	0,0011	0,00743	0,0288
49	CoDE ₅₁₂	19,1	3,81e-81	0,00102	0,00105	0,00107	0,00651	0,0288
50	rand/1/bin ₅₁₂	19,1	2,33e-81	0,001	0,00103	0,00105	0,00558	0,0288
51	EPSDE ₅₁₂	19,1	1,37e-81	0,00098	0,00101	0,00103	0,00465	0,0288
52	rand/1/bin ₂₅₆	19,3	2,22e-83	0,000962	0,000986	0,00101	0,00372	0,0288
53	EPSDE	19,6	2,24e-85	0,000943	0,000967	0,000993	0,00279	0,0288
54	EPSDE ₁₂₈	20,2	1,23e-90	0,000926	0,000949	0,000974	0,00186	0,0288
55	EPSDE ₂₅₆	20,3	1,02e-91	0,000909	0,000932	0,000956	0,000932	0,0288

Primerjava v okolju ekonomskih sistemov

Na sliki 9.3 vidimo primerjavo rezultatov med pristopi za razporejanje proizvodnje električne energije iz [71], kjer predlagan paralelni dvostopenjski algoritem z nadomestno matriko izboljša rezultate predhodnih pristopov. Zanimivost agregacije za teste v okolju HTS je, da ta primerja le najboljše dobljene rešitve. Namreč, zaradi zahtevnih omejitev so v preteklosti za primerjavo zadostovali le najboljši rezultati posameznih zagonov, saj so navadno razlike med poročanimi dobljenimi cenami urnikov HTS bistveno različne. Pristop [71] v okolju HTS tako izračuna praktično optimalne rezultate, kar pred tem ni uspelo nobenemu pristopu. Glavni razlog za tak rezultat je učinkovita obravnava omejitev, ki je deloma predizračunana. Rezultati tega problema razporejanja na uveljavljenih testih iz literature tudi kažejo večjo izboljšavo na vseh scenarijih in vseh kriterijih v primerjavi z do tedaj znanimi pristopi [71]. Poleg tega, da je bila izboljšana učinkovitost produkcije električne energije in zmanjšanja izpustov ter ogljičnega odtisa, so hkrati bile zadovoljene vse 24-urne sistemske in operativne zahteve v razporejanju povpraševane energije.

Če navedem še kakšno okolje iz nabora ekonomskih sistemov, lahko omenim, da za primerjavo trgovalnih finančnih ekonomskih sistemov na primer lahko obravnavamo sestavo portfelja z optimizacijskimi algoritmi, kar je opisano v [230]. Nek sestavljen portfelj ovrednotimo glede na podane ekonomske vrednosti ali pa določimo še filtrirni pogoj (signal) za njegovo uporabo.

Izhodišče za nadaljnjo literaturo v sklopu ovrednotenja in primerjave kakovosti algoritmov, vključno z ORLTES, podaja pregledni prispevek [221], ki je nastal v okviru skupine za ovrednotenja (angl. *Benchmarks*) pri ImA-ppNIO in katerega izsledki so bili predstavljeni in v več delavnicah diskutirani tako na GECCO (delavnica za ovrednotenje diskretne optimizacije s črno škatlo (angl. *Black Box Discrete Optimization Benchmarking*)), PPSN (tudi delavnica BB-DOB), kot tudi CEC (posebna sekcija za BEADO) in SEMCCO & FANCCO. Kot zanimivost, na SEMCCO & FANCCO je letos potekal še panel na temo ovrednotenja EA, saj je na konferenci potekalo tekmovanje EA, katerega rezultati so bili združeni v enoten seznam, ki je rangiral uvrščene algoritme¹. Skozi seminar so na tej konferenci bili predstavljeni tudi izzivi v ORLTES [228].

¹Algoritem DISH je na konferenci SEMCCO & FANCCO izkazal vse točke, prejel je 100 točk, kar je Kenneth Price zapisal tudi v poročilu <https://github.com/P-N-Suganthan/CEC2019/blob/master/100-DigitChallengeAnalysisofResults.pdf>, ki ga je objavil Ponnuthurai Nagaratnam Suganthan na GitHub (27. 11. 2019).

Poglavje 10

Samostojno delo – primeri za preverjanje

V tem poglavju je podanih nekaj primerov za preverjanje znanja in smernice, na kak način tvoriti novo znanje iz tematik **ORLTES**. Lahko pa je namenjeno zgolj kot samopreverjanje za študente, ali so zbrano gradivo iz zgornjih poglavij dovolj pozorno predelali. Seznam primerov je naslednji:

- **Primer: zapis znanja in razumevanja **ORLTES** preko zapisa slovarskih vnosov (indeksno kazalo) s pojmi **ORLTES****

Cilj: seznaniti se z "Hamdy A. Taha. Operations Research: An Introduction (10th Edition). Pearson. 2016."

Referenca na zbrano gradivo za rešitev: študent(ka) zapiše slovar osnovnih pojmov, podobno kot je zapisano pojmovno kazalo v tem dokumentu, pri čemer namesto številke strani navede zaporedne številke prispevkov iz literature, ki jih nato navede še kot reference.

- **Primer: zapis znanja in razumevanja **ORLTES** preko povzetka članka**

Cilj: razumeti prispevek: Das, Swagatam, Sankha Subhra Mullick, and Ponnuthurai N. Suganthan. *Recent advances in differential evolution—an updated survey*. Swarm and Evolutionary Computation 27 (2016): 1-30.

Referenca na zbrano gradivo za rešitev: študent(ka) zapiše dokument v LaTeX, v katerem poda besedilo, podobno kot je v tem dokumentu v razdelku "Algoritem diferencialne evolucije" na strani 18.

- **Primer: pregled programske kode, ki je na voljo na GitHub. Zagon primera testnih funkcij na izbranem optimizatorju (kodi)**

Cilj: pregledati naslednje tri repozitorije kode:

- <https://github.com/P-N-Suganthan>.
- <https://github.com/jMetal/jMetal>.

– <https://github.com/numbbo/coco>.

Referenca na zbrano gradivo za rešitev: študent(ka) prenese repozitorije na svoj računalnik in izvede teste za CEC 2017, CEC 2019 in COCO, za algoritem [DISH](#), podobno kot je to narejeno v referencirani literaturi za algoritem [DISH](#) v prejšnjih poglavjih tega dokumenta. Izvoro kodo algoritma [DISH](#) je objavil Ponnuthurai Nagaratnam Suganthan v GitHubu na predlog in potrditev Kennetha Priceja v oktobru 2019 po zaključku tekmovanja [CEC 2019](#) in se nahaja na naslovu <https://github.com/P-N-Suganthan/CEC2019/blob/master/PaperCodes-DISHchain.zip>.

- **Primer: dopolnitev nabora testnih funkcij s primerom ORLTES**

Cilj: dodati nov tip testne funkcije za [ORLTES](#).

Referenca na zbrano gradivo za rešitev: priprava prispevka, kjer je opisano, kako dodati nabor testnih funkcij iz primera optimizacije z omejitvami (na primer, kot za [UGPP](#) – na referenci <http://dx.doi.org/10.1016/j.apenergy.2014.12.020> oz. za [HTS](#) – na referenci <http://dx.doi.org/10.1016/j.asoc.2016.01.038>).

- **Primer: Analiza in načrtovanje primernih sistemov v ORLTES za reševanje izbranega problema iz določenega področja**

Cilj: identificirati, kateri sistem [ORLTES](#) je primeren, da lahko zanj načrtujemo neko novo rešitev.

Referenca na zbrano gradivo za rešitev: <http://dx.doi.org/10.1016/j.ins.2014.04.037> – v tem prispevku najdemo problem morfološke rekonstrukcije.

- **Uporaba funkcij CEC 2011 – vesoljske misije Messenger in Cassini**

Cilj: modeliranje trajektorij v vesolju.

Referenca na zbrano gradivo za rešitev: s pomočjo <https://github.com/P-N-Suganthan> poiščemo testni nabor funkcij [CEC 2011](#), med katerimi je implementiran tudi model za načrtovanje vesoljskih misij, kot sta Messenger in Cassini. Nato poženemo optimizacijski algoritem za ta model in poročamo rezultate.

- **Aplikacija algoritma ORLTES za vodenje izbranega sistema v nepredvidljivem okolju in uporaba funkcij UGPP**

Cilj: aplicirati algoritem, ki vodi sistem za navigacijo podvodnih morskih sond (UGPP).

Referenca na zbrano gradivo za rešitev: <http://dx.doi.org/10.1016/j.eswa.2018.10.048> — izberemo algoritem L-SHADE in ga zaženemo na funkcijah UGPP ter poročamo rezultate.
- **Aplikacija funkcij z omejitvami: problem UGPP z omejitvami**

Cilj: aplicirati algoritem, ki vodi sistem za navigacijo podvodnih morskih sond, kjer so posebej obravnavane terenske omejitve (UGPP z omejitvami).

Referenca na zbrano gradivo za rešitev: <http://dx.doi.org/10.1016/j.asoc.2016.01.038> — izberemo optimizacijski algoritem diferencialne evolucije z obravnavo omejitev in ga zaženemo na funkcijah UGPP z omejitvami ter poročamo rezultate.
- **Primer: dopolnitev algoritma ORLTES za funkcije z omejitvami na problemu HTS**

Cilj: dopolniti algoritem za optimizacijo, ki obravnava omejitve pri razporejanju delovanja elektrarn.

Referenca na zbrano gradivo za rešitev: <http://dx.doi.org/10.1016/j.apenergy.2014.12.020> — algoritem diferencialne evolucije nadgradimo z dvonivojskim reševanjem modela z omejitvami in poročamo rezultate.
- **Zapis prispevka o obravnavi napak med misijami in zanesljivosti transporta — fizikalni izzivi kibernetiko-tehničnih naprav**

Cilj: nasloviti robustnost v delno avtonomnih kibernetiko-tehničnih napravah.

Referenca na zbrano gradivo za rešitev: <http://dx.doi.org/10.3389/frobt.2016.00014> — opišemo mala omrežja (angl. *small world networks*) in s tem povezane rešitve za kibernetiko-tehnične naprave.
- **Dopolnitev ORLTES za ekonomsko planiranje z novo ustvarjenim znanjem — funkcije DL/FX**

Cilj: za izbrane kriptovalute želimo opisati plan investiranja.

Referenca na zbrano gradivo za rešitev: http://dx.doi.org/10.1007/978-3-030-16272-6_12 — preučimo algoritme za sestavo portfelja, referencirane v tem prispevku in zaženemo obdelavo podatkov.

- **Primerjava kakovosti algoritmov in analiza novo načrtovanega ORLTES kot celote**

Cilj: želimo primerjati algoritme in analizirati kakovost novo načrtovanega ORLTES kot celote.

Referenca na zbrano gradivo za rešitev: <http://dx.doi.org/10.1016/j.eswa.2018.10.048> — izberemo primerjavo z rangiranjem in poročamo rezultate nad funkcijami UGPP. Modificiramo funkcije UGPP in poročamo novo modifikacijo skupaj z rangiranjem — na primer kot odana naloga na Moodle (estudij.um.si) ali kot objavljeno tehnično poročilo o raziskavi, konferenčni prispevek na konferenci CEC (za to je na primer ustrezna posebna sekcija *ovrednotenje evolucijskih algoritmov za diskretno optimizacijo* (angl. *Benchmarking of Evolutionary Algorithms for Discrete Optimization*) (BEADO), lahko pa tudi katera druga posebna sekcije za zvezno optimizacijo) ali konferencah SEMCCO & FANCCO oz. GECCO, najboljše rezultate pa lahko pošljemo v recenzijo pri revijah, kot so npr. SWEVO, ASOC in ESWA, tudi TEVC in ECJ. Pri pripravi dokumentacije obvezno upoštevamo tudi ustrezno navajanje virov.

Poglavje 11

Zaključek

Skozi poglavja je bil podan pregled zbranega študijskega gradiva za predavanja pri predmetu Operacijske raziskave logističnih, transportnih in ekonomskih sistemov, uvrščenem v tretjestopenjski bolonjski študijski program Računalništvo in informatika kot izbirna učna enota.

To učno gradivo je pripravljeno v upanju, da bo študentom v pomoč in dodatno poglobitev pri svojem delu. V osnovi to gradivo služi tudi kot osnutek izhodiščnih gradiv za izvedbo avtorskih predavanj pri predmetu [ORLTES](#), lahko pa tudi kot platforma za dopolnilo k pripravi študenta in izvajalca predmeta pri uvedbi in spremljanju študijskega dela v primeru mentorske izvedbe tega predmeta. Ker to zbrano gradivo navaja številne citate v literaturi in nekatera dela posebej izpostavi, je nadaljnje delo po preučitvi tega gradiva predvidena diskusija z izvajalcem učne enote, ki na podlagi izzvanih študentskih kritičnih vprašanj ponudi nadaljne usmeritve.

Literatura

- [1] 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*, številka 2, strani 971–978, Piscataway, New Jersey, 2001. IEEE Service Center. 26, 27
- [2] B. Alatas, E. Akin in A. Karci. MODENAR: Multi-objective differential evolution algorithm for mining numeric association rules. *Applied Soft Computing*, 8(1):646–656, 2008. 26, 29
- [3] M. M. Ali. Differential evolution with preferential crossover. *European Journal of Operational Research*, 127(3):1137–1147, 2007. 22
- [4] M. Aono in T. Kunii. Botanical tree image generation. *IEEE Computer Graphics and Applications*, 4(5):10–34, 1984. 45
- [5] B. V. Babu in M. M. L. Jehan. Differential Evolution for Multi-Objective Optimization. V *Proceedings of the 2003 Congress on Evolutionary Computation (CEC 2003)*, številka 4, strani 2696–2703, Canberra, Australia, 2003. IEEE Press. 26
- [6] B. V. Babu, J. H. S. Mubeen in P. G. Chakole. Multiobjective Optimization Using Differential Evolution. *TechGenesis – The Journal of Information Technology*, 2(2):4–12, 2005. 26
- [7] T. Bäck. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, New York, 1996. 17, 23
- [8] R. Balling, B. Theodore in R. Wilkins. Engineering Hyperstructures for Cities of the Future. V *Structures Congress 2010*, strani 3198–3206. ASCE, 2010. 7
- [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)*, uredniki: W. B. Langdon, E. Cantú-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M. A. Potter, A. C. Schultz, J. F. Miller, E. Burke in N. Jonoska, strani 995–1002, San Francisco, California, 2002. Morgan Kaufmann Publishers. 29
- [10] N. Barricelli. Numerical testing of evolution theories. *Acta Biotheoretica*, 16(1):69–98, 1962. 6, 15
- [11] D. Beaumont in S. Stepney. Grammatical Evolution of L-systems. V *The 2009 IEEE Congress on Evolutionary Computation CEC 2009*, strani 2446–2453. IEEE Press, 2009. 44, 48

- [12] G. Beni in J. Wang. Swarm intelligence in cellular robotic systems. *Robots and Biological Systems: Towards a New Bionics?*, strani 703–712, 1993. [17](#)
- [13] H.-G. Beyer in H.-P. Schwefel. Evolution strategies — A comprehensive introduction. *Natural Computing*, 1:3–52, 2002. [17](#), [23](#)
- [14] J. Bloomenthal. Modeling the mighty maple. V *SIGGRAPH '85 Conference Proceedings, San Francisco, CA, 22–26 July 1985*, urednik: B. A. Barsky, strani 305–311, 1985. [45](#)
- [15] B. Bošković, J. Brest, A. Zamuda, S. Greiner in V. Žumer. History Mechanism Supported Differential Evolution for Chess Evaluation Function Tuning. *Soft Computing – A Fusion of Foundations, Methodologies and Applications*, 15(4):667–682, 2011. [29](#)
- [16] 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*, strani 6742–6747. IEEE Press, 2006. [29](#)
- [17] B. Brenčič. Optimizacija s čebelami. Diplomaska naloga 1. bolonjske stopnje, mentor: J. Brest, somentor: A. Zamuda, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, 2011. [17](#)
- [18] J. Brest. Constrained Real-Parameter Optimization with ϵ -Self-Adaptive Differential Evolution. V *Constraint-Handling in Evolutionary Optimization*, strani 73–93. Springer, 2009. [24](#), [25](#), [65](#)
- [19] J. Brest. Differential Evolution with Self-Adaptation. *Encyclopedia of Artificial Intelligence*, strani 488–493, 2009. [23](#), [25](#)
- [20] 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*, 11(7):617–629, 2007. [23](#), [25](#)
- [21] 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*, 10(6):646–657, 2006. [xii](#), [18](#), [19](#), [22](#), [23](#), [24](#), [25](#), [44](#), [51](#)
- [22] J. Brest, P. Korošec, J. Šilc, A. Zamuda, B. Bošković in M. S. Maučec. Differential evolution and differential ant-stigmergy on dynamic optimisation problems. *International Journal of Systems Science*, 44(4):663–679, 2013. [17](#), [25](#), [26](#), [29](#), [44](#)
- [23] J. Brest in M. S. Maučec. Population Size Reduction for the Differential Evolution Algorithm. *Applied Intelligence*, 29(3):228–247, 2008. [25](#), [60](#)
- [24] 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*, strani 919–926. IEEE Press, 2006. [22](#), [23](#), [25](#)
- [25] J. Brest, V. Žumer in M. S. Maučec. Population size in differential evolution algorithm. *Elektrotehniški vestnik*, 74(1-2):55–60, 2007. [23](#)

- [26] 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*, strani 2032–2039. IEEE Press, 2008. [25](#)
- [27] J. Brest, A. Zamuda, B. Bošković, M. S. Maučec in V. Žumer. Dynamic Optimization using Self-Adaptive Differential Evolution. V *IEEE Congress on Evolutionary Computation 2009*, strani 415–422. IEEE Press, 2009. [25](#), [44](#)
- [28] J. Brest, A. Zamuda, B. Bošković in V. Žumer. Večkriterijska optimizacija: primerjava algoritmov *MOjDE* in *DEMO*. V *Zbornik šestnajste mednarodne Elektrotehniške in računalniške konference ERK 2007, vol. B*, strani 85–88, 2007. [25](#)
- [29] J. Brest, A. Zamuda, B. Bošković in V. Žumer. An Analysis of the Control Parameters' Adaptation in DE. V *Advances in Differential Evolution, Studies in Computational Intelligence*, urednik: U. K. Chakraborty, številka 143, strani 89–110. Springer, 2008. [22](#), [25](#)
- [30] J. Brest, A. Zamuda, I. Fister, B. Bošković in M. S. Maučec. Constrained real-parameter optimization using a differential evolution algorithm. V *IEEE SSCI2011 symposium series on computational intelligence: proceedings. Piscataway: IEEE*, strani 9–16, 2011. [25](#), [65](#)
- [31] J. Brest, A. Zamuda, I. Fister in M. S. Maučec. Large Scale Global Optimization using Self-adaptive Differential Evolution Algorithm. V *IEEE World Congress on Computational Intelligence 2010, July 18 - 23, Barcelona, Spain*, strani 3718–3725, 2008. [25](#)
- [32] D. Brown. *Da Vincijeva šifra*. Mladinska knjiga, 2005. [7](#)
- [33] 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*, 32(4):460–473, 2002. [26](#), [29](#)
- [34] E. A. W. Budge. *The Papyrus of Ani in the British Museum; the Egyptian Text with Interlinear Transliteration and Translation, a Running Translation, Introduction, etc.* British Museum (reprint Dover Publications, 1967), 1895. [14](#)
- [35] A. Caponio, F. Neri in V. Tirronen. Super-fit control adaptation in memetic differential evolution frameworks. *Soft Computing—A Fusion of Foundations, Methodologies and Applications*, 13(8):811–831, 2009. [22](#)
- [36] B. Chamberlain. *The Kojiki: Records of ancient matters*. Tuttle Publishing, 2005. [14](#)
- [37] S. Y. Chong, P. Tino in X. Yao. Measuring generalization performance in co-evolutionary learning. *IEEE Transactions on Evolutionary Computation*, 12(4):479–505, 2008. [29](#)
- [38] C. A. Coello Coello. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, 191(11-12):1245–1287, 2002. [14](#)

- [39] 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*, urednika: G. Y. Yen in D. B. Fogel, poglavje 4, strani 73–88. IEEE Computational Intelligence Society, Vancouver, Canada, 2006. [10](#), [26](#)
- [40] C. A. Coello Coello. Evolutionary multi-objective optimization: a historical view of the field. *Computational Intelligence Magazine, IEEE*, 1(1):28–36, Feb. 2006. [10](#), [26](#)
- [41] 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. [7](#), [14](#), [20](#)
- [42] C. Darwin in A. R. Wallace. On the Tendency of Species to form Varieties; and on the Perpetuation of Varieties and Species by Natural Means of Selection. *Journal of the Proceedings of the Linnean Society of London, Zoology* 3:46–50, 1858. [14](#)
- [43] E. Darwin. *Zoonomia; or, the laws of organic life*, številka 2. Byrnie, 1800. [14](#)
- [44] S. Das, A. Abraham, U. Chakraborty in A. Konar. Differential Evolution Using a Neighborhood-based Mutation Operator. *IEEE Transactions on Evolutionary Computation*, 13(3):526–553, 2009. [22](#)
- [45] S. Das in P. N. Suganthan. Differential Evolution: A Survey of the State-of-the-art. *IEEE Transactions on Evolutionary Computation*, 15(1):4–31, 2011. [18](#), [23](#)
- [46] R. Dawkins. *The Selfish Gene – with a new Introduction by the Author*. Oxford University Press, USA, 2006. [7](#), [14](#)
- [47] 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*, številka 1, strani 566–573, Lisbon, Portugal, 2003. [29](#)
- [48] K. Deb in H. Gupta. Introducing robustness in multi-objective optimization. *Evolutionary Computation*, 14(4):463–494, 2006. [11](#)
- [49] 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*, 13(4):501–525, 2005. [11](#)
- [50] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK, 2001. ISBN 0-471-87339-X. [9](#), [11](#), [26](#), [29](#)
- [51] 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*, uredniki: M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo in H.-P. Schwefel, strani 849–858, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917. [26](#)
- [52] K. Deb, S. Agrawal, A. Pratab in T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. KanGAL report 200001, Indian Institute of Technology, Kanpur, India, 2000. [26](#)
- [53] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006. [70](#)

- [54] 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*, 11(1):17–45, 2007. [26](#), [27](#)
- [55] M. Dorigo, V. Maniezzo in A. Coloni. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 26(1):29–41, 1996. [17](#)
- [56] W. Duch in J. Mandziuk. Quo Vadis Computational Intelligence? V *Machine Intelligence: Quo Vadis?*, urednik: K. H. P. Sincak, J. Vascak, številka 21 iz *Advances in Fuzzy Systems – Applications and Theory*, strani 3–28. World Scientific, 2004. [6](#)
- [57] P. Ehrenfreund, W. Irvine, L. Becker, J. Blank, J. R. Brucato, L. Colangeli, S. Derenne, D. Despois, A. Dutrey, H. Fraaije, A. Lazcano, T. Owen, F. Robert in I. S. S. I. ISSI-Team. Astrophysical and astrochemical insights into the origin of life. *Reports on Progress in Physics*, 65(10):1427, 2002. [7](#)
- [58] A. E. Eiben in J. E. Smith. *Introduction to Evolutionary Computing (Natural Computing Series)*. Springer, 2003. [14](#), [23](#)
- [59] A. E. Eiben, R. Hinterding in Z. Michalewicz. Parameter Control in Evolutionary Algorithms. *IEEE Trans. on Evolutionary Computation*, 3(2):124–141, 1999. [11](#), [23](#)
- [60] A. Einstein. Die grundlage der allgemeinen relativitätstheorie. *Annalen der Physik*, 354(7):769–822, 1916. [7](#)
- [61] 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*, strani 416–423, Nice, France, 1997. American Society of Mechanical Engineering. [29](#)
- [62] H.-Y. Fan in J. Lampinen. A Trigonometric Mutation Operation to Differential Evolution. *Journal of Global Optimization*, 27(1):105–129, 2003. [21](#)
- [63] V. Feoktistov. *Differential Evolution: In Search of Solutions (Springer Optimization and Its Applications)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. [23](#), [29](#)
- [64] D. B. Fogel. *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence, Third Edition (IEEE Press Series on Computational Intelligence)*. Wiley-IEEE Press, Hoboken, New Jersey, 2006. [15](#), [23](#)
- [65] L. J. Fogel, A. J. Owens in M. J. Walsh. *Artificial Intelligence Through Simulated Evolution*. John Wiley & Sons Inc, 1966. [6](#), [15](#), [16](#)
- [66] A. Fraser. Simulation of Genetic Systems by Automatic Digital Computers VI. Epistasis. *Australian Journal of Biological Sciences*, 13(2):150–162, 1957. [6](#), [9](#), [15](#)
- [67] T. Freeth, Y. Bitsakis, X. Moussas, J. Seiradakis, A. Tselikas, H. Mangou, M. Zafeiropoulou, R. Hadland, D. Bate, A. Ramsey et al. Decoding the ancient greek astronomical calculator known as the antikythera mechanism. *Nature*, 444(7119):587, 2006. [8](#)
- [68] S. García, D. Molina, M. Lozano in F. Herrera. A study on the use of non-parametric tests for analyzing the evolutionary algorithms’ behaviour: a case study on the CEC’2005 Special Session on Real Parameter Optimization. *Journal of Heuristics*, 15(6):617–644, 2009. [67](#), [69](#)

- [69] Z. Geem in J. Choi. Music composition using harmony search algorithm. *Applications of Evolutionary Computing*, strani 593–600, 2007. 18
- [70] S. Ghosh, S. Roy, S. Islam, S. Das in P. Suganthan. A differential covariance matrix adaptation evolutionary algorithm for global optimization. V *2011 IEEE Symposium on Differential Evolution (SDE)*, strani 1 –8, 2011. 17
- [71] A. Glotić in A. Zamuda. Short-term combined economic and emission hydrothermal optimization by surrogate differential evolution. *Applied Energy*, 141:42–56, 1 March 2015. x, 29, 64, 66, 71, 73
- [72] F. Glover. Tabu Search – Part I. *ORSA Journal on computing*, 1(3):190–206, 1989. 9, 26
- [73] C. K. Goh in K. C. Tan. An Investigation on Noisy Environments in Evolutionary Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, 11(3):354–381, 2007. 29
- [74] D. Goldberg. What every computer scientist should know about floating-point arithmetic. *ACM Computing Surveys (CSUR)*, 23(1):5–48, 1991. 18
- [75] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1989. 16
- [76] 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*, 1(3):296–297, 1971. 12
- [77] H. Hamann, Y. Khaluf, J. Botev, M. D. Soorati, E. Ferrante, O. Kosak, J.-M. Montanier, S. Mostaghim, R. Redpath, J. Timmis, F. Veenstra, M. Wahby in A. Zamuda. Hybrid Societies: Challenges and Perspectives in the Design of Collective Behavior in Self-organizing Systems. *Frontiers in Robotics and AI*, 3(14):1–8, 2016. 49
- [78] M. P. Hansen in A. Jaskiewicz. Evaluating the quality of approximations to the non-dominated set. Technical Report IMM-REP-1998-7, Technical University of Denmark, 1998. 67
- [79] N. Hansen, S. Muller in P. Koumoutsakos. Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). *Evolutionary Computation*, 11(1):1–18, 2003. 23
- [80] N. Hansen in A. Ostermeier. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation*, 9(2):159–195, 2001. 17, 23
- [81] R. P. Hardie in R. K. Gaye. *English translation of Aristotle: Physics, 350 AC*. Oxford: The Clarendon Press, 1930. 14
- [82] S. Hawking, C. Sagan, R. Miller et al. *A brief history of time: From the big bang to black holes*. Bantam Books, 1988. 7
- [83] S. Heath. *The Method of Archimedes, recently discovered by Heiberg: A supplement to the Works of Archimedes*. Cosimo Classics, 2007. 14
- [84] J. Holland. *Adaptation In Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, 1975. 6, 15
- [85] M. Holton. Strands, gravity, and botanical tree imagery. *Comput. Graph. Forum*, 13(1):57–67, 1994. 45

- [86] 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. Technical Report TR-07-01, Nanyang Technological University et. al., Singapore, 2007. [28](#)
- [87] 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)*, strani 17–24. IEEE Service Center, 2006. [22](#)
- [88] 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*, strani 3601–3608. IEEE Press, 2007. [28](#)
- [89] V. L. Huang, P. N. Suganthan, A. K. Qin in S. Baskar. Multiobjective Differential Evolution with External Archive and Harmonic Distance-Based Diversity Measure. Technical Report TR-07-01, Nanyang Technological University, Singapore, 2006. [26](#)
- [90] E. J. Hughes. Swarm Guidance using a Multi-Objective Co-evolutionary On-Line Evolutionary Algorithm. V *2004 Congress on Evolutionary Computation (CEC 2004)*, številka 2, strani 2357–2363, Portland, Oregon, USA, 2004. IEEE Service Center. [29](#)
- [91] J. Huxley. *Evolution: The Modern Synthesis*. Allen & Unwin, London, 1942. [14](#)
- [92] J. Huxley. The emergence of Darwinism. *Perspectives in biology and medicine*, 3:321, 1960. [15](#)
- [93] T. H. Huxley. ART. VIII. - Darwin on the origin of Species. *Westminster Review*, 3:541–570, 1860. [14](#)
- [94] C. Igel, N. Hansen in S. Roth. Covariance Matrix Adaptation for Multi-objective Optimization. *Evolutionary Computation*, 15(1):1–28, 2007. [17](#), [23](#)
- [95] 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)*, urednik: M. K. et al., številka 1, strani 691–697, Seattle, Washington, USA, 2006. ACM Press. ISBN 1-59593-186-4. [18](#)
- [96] W. D. Jones. How to build a mile-high skyscraper. *Spectrum, IEEE*, 44(6):52–53, 2007. [7](#)
- [97] R. Joshi in A. Sanderson. Minimal representation multisensor fusion using differential evolution. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 29(1):1083–4427, 1999. [29](#)
- [98] D. Karaboga in B. Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39(3):459–471, 2007. [17](#)
- [99] J. Kennedy in R. Eberhart. Particle swarm optimization. V *Neural Networks, 1995. Proceedings., IEEE International Conference on*, številka 4, strani 1942–1948. IEEE, 1995. [17](#)
- [100] J. Kephart. A biologically inspired immune system for computers. V *Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*, strani 130–139, 1994. [17](#)

- [101] S. Kirkpatrick, C. Gelatt in M. Vecchi. Optimization by simulated annealing. *science*, 220(4598):671, 1983. 9, 26
- [102] M. Kitti. History of Optimization. V *Systeemianalyysin laboratorio*, 3. 10. 2011. URL <http://www.mitrikitti.fi/opthist.html>. 8, 10
- [103] J. Knowles, L. Thiele in E. Zitzler. A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers. 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland, 2006. revised version. 67
- [104] D. Knuth. Computer programming as an art. *Communications of the ACM*, 17(12):667–673, 1974. 8
- [105] P. Korošec, J. Šilc in B. Filipič. The differential ant-stigmergy algorithm. *Information Sciences*, 192:82–97, 2012. 25, 29
- [106] 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*, strani 157–164. IEEE Press, 2007. 17
- [107] J. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992. 16
- [108] S. Koziel in Z. Michalewicz. Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization. *Evolutionary Computation*, 7(1):19–44, 1999. 13
- [109] S. Kukkonen in J. Lampinen. GDE3: The third Evolution Step of Generalized Differential Evolution. V *2005 IEEE Congress on Evolutionary Computation (CEC 2005)*, številka 1, strani 443–450, Edinburgh, Scotland, 2005. IEEE Service Center. 26
- [110] 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*, strani 3593–3600. IEEE Press, 2007. 26, 27
- [111] D. Kundu, K. Suresh, S. Ghosh, S. Das, B. K. Panigrahi in S. Das. Multi-objective optimization with artificial weed colonies. *Information Sciences*, 181(12):2441–2454, 2011. 17
- [112] J. A. Lampinen. A Bibliography of Differential Evolution Algorithm. Technical report, Lappeenranta University of Technology, Department of Information Technology, Laboratory of Information Processing, 2001. URL <http://www2.lut.fi/~jlampine/debiblio.htm>. 21
- [113] W. Langdon, S. Gustafson in J. Koza. The genetic programming bibliography. V *Collection of Computer Science Bibliographies*, 2011. URL <http://www.cs.bham.ac.uk/~wbl/biblio/>. 16
- [114] 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*, 169(3):932–942, 2006. 29
- [115] P. Lermusiaux, D. Subramani, J. Lin, C. Kulkarni, A. Gupta, A. Dutt, T. Lolla, P. Haley, W. Ali, C. Mirabito in S. Jana. A future for intelligent autonomous ocean observing systems. *Journal of Marine Research*, 75(6):765–813, 2017. 62, 71

- [116] J. J. Liang, T. P. Runarsson, E. Mezura-Montes, M. Clerc, P. N. Suganthan, C. A. C. Coello in K. Deb. Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-Parameter Optimization. Technical Report Technical Report Report 2006005, Nanyang Technological University, Singapore, 2005. [14](#)
- [117] K. Liang, X. Yao in C. Newton. Adapting Self-adaptive Parameters in Evolutionary Algorithms, 2001. [21](#), [23](#)
- [118] E. Limpert, W. Stahel in M. Abbt. Log-normal Distributions across the Sciences: Keys and Clues. *BioScience*, 51(5):341–352, 2001. [23](#)
- [119] C. Linnæus. *Systema naturæ per regna tria naturæ, secundum classes, ordines, genera, species, cum characteribus, differentiis, synonymis, locis*. Laurentii Salvii, Stockholm, 1758. [7](#)
- [120] J. Liu in J. Lampinen. A Fuzzy Adaptive Differential Evolution Algorithm. *Soft Comput.*, 9(6):448–462, 2005. [21](#), [23](#)
- [121] M. Livio. *The golden ratio: The story of phi, the world's most astonishing number*. Broadway, 2003. [7](#)
- [122] N. K. Madavan. Multiobjective Optimization Using a Pareto Differential Evolution Approach. V *Congress on Evolutionary Computation*, številka 2, strani 1145–1150, Piscataway, New Jersey, 2002. IEEE Service Center. [26](#), [27](#)
- [123] R. Mallipeddi, P. N. Suganthan, Q. K. Pan in M. F. Tasgetiren. Differential evolution algorithm with ensemble of parameters and mutation strategies. *Applied Soft Computing*, 11(2):1679–1696, 2011. [23](#)
- [124] U. Maulik in I. Saha. Modified differential evolution based fuzzy clustering for pixel classification in remote sensing imagery. *Pattern Recognition*, 42(9):2135–2149, 2009. [29](#)
- [125] J. McCarthy. Recursive functions of symbolic expressions and their computation by machine, Part I. *Communications of the ACM*, 3(4):184–195, 1960. [16](#)
- [126] J. McCarthy, M. Minsky, N. Rochester in C. Shannon. A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955. *AI Magazine*, 27(4):12, 2006. [6](#), [15](#)
- [127] G. Mendel. *Experiments in Plant Hybridization*. Cosimo, Inc., 1865. [14](#)
- [128] E. Mezura-Montes. *Alternative Techniques to Handle Constraints in Evolutionary Optimization*. Doktorska disertacija, Computer Science Section, Electrical Eng. Department, 2004. [14](#)
- [129] E. Mezura-Montes in B. C. Lopez-Ramirez. Comparing bio-inspired algorithms in constrained optimization problems. *The 2007 IEEE Congress on Evolutionary Computation*, strani 662–669, 25-28 Sept. 2007. [18](#), [23](#)
- [130] E. Mezura-Montes, J. Velázquez-Reyes in C. A. C. Coello. A comparative study of differential evolution variants for global optimization. V *GECCO 2006: Proceedings of the 8th annual conference on Genetic and evolutionary computation*, strani 485–492, New York, NY, USA, 2006. ACM Press. [23](#)

- [131] Z. Michalewicz in D. B. Fogel. *How to Solve It: Modern Heuristics*. Springer, Berlin, 2000. 14
- [132] Z. Michalewicz in M. Schoenauer. Evolutionary Algorithms for Constrained Parameter Optimization Problems. *Evolutionary Computation*, 4(1):1–32, 1996. 13
- [133] E. Mininno, F. Neri, F. Cupertino in D. Naso. Compact Differential Evolution. *IEEE Transactions on Evolutionary Computation*, 15(1):32–54, 2011. 23
- [134] S. Mirjalili in A. Lewis. S-shaped versus v-shaped transfer functions for binary particle swarm optimization. *Swarm and Evolutionary Computation*, 9:1–14, 2013. 36
- [135] T. I. Museum. Kumranski rokopisi. V *The Digital Dead Sea Scrolls*, Digitalizirano (Google), 2011. URL <http://dss.collections.imj.org.il/>. 14
- [136] J. Nelder in R. Mead. A simplex method for function minimization. *The computer journal*, 7(4):308, 1965. 9, 18
- [137] F. Neri in E. Mininno. Memetic compact differential evolution for cartesian robot control. *IEEE Computational Intelligence Magazine*, 5(2):54–65, 2010. 29
- [138] F. Neri in V. Tirronen. Recent Advances in Differential Evolution: A Survey and Experimental Analysis. *Artificial Intelligence Review*, 33(1–2):61–106, 2010. 18, 23
- [139] F. Neri in C. Cotta. Memetic algorithms and memetic computing optimization: A literature review. *Swarm and Evolutionary Computation*, 2:1–14, 2012. 17
- [140] B. Neubert, T. Franken in O. Deussen. Approximate image-based tree-modeling using particle flows. *ACM Trans. Graph.*, 26(3):88, 2007. 45
- [141] I. Newton. *Philosophiæ naturalis principia mathematica*. Royal Society, London, 1687. 8
- [142] E. Nisbet in N. Sleep. The habitat and nature of early life. *Nature*, 409(6823):1083–1091, 2001. 7
- [143] J. A. Oaks in H. M. Alkhateeb. Simplifying equations in arabic algebra. *Historia Mathematica*, 34(1):45–61, 2007. 6
- [144] J. J. O’Connor in E. F. Robertson. A chronology of pi. Technical report, The MacTutor History of Mathematics archive, 2000. URL http://www-history.mcs.st-andrews.ac.uk/HistTopics/Pi_chronology.html. 7
- [145] J. J. O’Connor in E. F. Robertson. George dantzig. Technical report, The MacTutor History of Mathematics archive, 2003. URL http://www-history.mcs.st-andrews.ac.uk/Biographies/Dantzig_George.html. 6, 8
- [146] M. G. Omran, A. Salman in A. P. Engelbrecht. Self-adaptive Differential Evolution. V *Computational intelligence and security*, strani 192–199, 2005. 21, 23
- [147] K. Passino. Bacterial foraging optimization. *International Journal of Swarm Intelligence Research*, 1(1):1–16, 2010. 17
- [148] D. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim in M. Zaidi. The Bees Algorithm – A Novel Tool for Complex Optimisation Problems. V *Proceedings of IPROMS 2006 conference*, strani 454–461, 2006. 17

- [149] A. Posamentier in N. Gordan. An astounding revelation on the history of pi. *Mathematics Teacher*, 1984. [7](#)
- [150] K. V. Price in R. M. Storn. Differential Evolution: A Simple Evolution Strategy for Fast Optimization. *Dr. Dobb's Journal of Software Tools*, 22(4):18–24, 1997. [20](#)
- [151] K. V. Price, R. M. Storn in J. A. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization*. Natural Computing Series. Springer-Verlag, Berlin, Germany, 2005. [23](#), [29](#)
- [152] W. Price. A controlled random search procedure for global optimisation. *The Computer Journal*, 20(4):367–370, 1977. [9](#), [18](#)
- [153] A. K. Qin, V. L. Huang in P. N. Suganthan. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation*, 13(2):398–417, 2009. [22](#), [28](#)
- [154] A. K. Qin in P. N. Suganthan. Self-adaptive Differential Evolution Algorithm for Numerical Optimization. V *Proceedings of the 2005 Congress on Evolutionary Computation*, številka 2, strani 1785–1791. IEEE Press, 2005. [22](#), [23](#)
- [155] L. Quan. *Image-Based Modeling*. Springer Publishing Company, Incorporated, 1st edition, 2010. [44](#)
- [156] L. Quintana-Murci, O. Semino, H. Bandelt, G. Passarino, K. McElreavey in A. Santachiara-Benerecetti. Genetic evidence of an early exit of Homo sapiens sapiens from Africa through eastern Africa. *Nat Genet*, 23(4):437–441, 1999. [7](#)
- [157] I. Rechenberg. *Evolutionsstrategie–Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Doktorska disertacija, Frommann-Holzboog, Stuttgart, 1973. [6](#), [15](#)
- [158] W. Reeves. Approximate and probabilistic algorithms for shading and rendering structured particle systems. *Proceedings of SIGGRAPH'85*, strani 313–322, 1985. [45](#)
- [159] Rigveda. *Vede*. Indija, 2000 AC. [14](#)
- [160] 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*, številka B, strani 121–124, 2005. [26](#)
- [161] 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*, številka 3410 iz *Lecture Notes in Computer Science*, strani 520–533. Springer, 2005. [26](#), [27](#)
- [162] G. Romanes. Post-darwinian questions: heredity and utility. V *Darwin, and after Darwin*. Longmans, Green: London, 1895. [14](#)
- [163] L. V. Santana-Quintero in C. A. Coello Coello. An Algorithm Based on Differential Evolution for Multi-Objective Problems. *International Journal of Computational Intelligence Research*, 1(2):151–169, 2005. [26](#)
- [164] J. D. Schaffer. Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. V *Proceedings of the 1st International Conference on Genetic Algorithms*, strani 93–100, Mahwah, NJ, USA, 1985. Lawrence Erlbaum Associates, Inc. [26](#)

- [165] H.-P. Schwefel. *Evolution and Optimum Seeking*. Sixth-Generation Computer Technology. Wiley Interscience, New York, 1995. [17](#), [23](#)
- [166] 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*, strani 37–45, Washington, D.C., July 1999. IEEE Service Center. [29](#)
- [167] M. Smid. Closest-Point Problems in Computational Geometry. *Handbook on Computational Geometry*, edited by J.-R. Sack, North Holland, Amsterdam, strani 877–935, 2000. [27](#)
- [168] K. Sörensen. Metaheuristics—the metaphor exposed. *International Transactions in Operational Research*, 22(1):3–18, 2015. [16](#)
- [169] N. Srinivas in K. Deb. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248, 1994. [26](#)
- [170] A. P. Stakhov. The golden section in the measurement theory. *Computers & Mathematics with Applications*, 17(4-6):613–638, 1989. [8](#)
- [171] R. M. Storn in K. V. Price. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11:341–359, 1997. [xii](#), [18](#), [21](#)
- [172] T. Takahama, S. Sakai in N. Iwane. Solving Nonlinear Constrained Optimization Problems by the ϵ Constrained Differential Evolution. *IEEE International Conference on Systems, Man and Cybernetics 2006 (SMC 2006)*, 3:2322–2327, 2006. [13](#), [22](#), [25](#)
- [173] K. Tan, Y. Yang in C. Goh. A distributed Cooperative coevolutionary algorithm for multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 10(5):527–549, 2006. [26](#), [29](#)
- [174] R. Tanabe in A. S. Fukunaga. Improving the search performance of SHADE using linear population size reduction. V *2014 IEEE Congress on Evolutionary Computation*, strani 1658–1665. IEEE, 2014. [31](#)
- [175] J. Teo. Exploring dynamic self-adaptive populations in differential evolution. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 10(8):673–686, 2006. [22](#), [23](#)
- [176] J. H. Tigay. *The evolution of the Gilgamesh epic*. University of Pennsylvania Press, 1982. [14](#)
- [177] V. Tirronen, F. Neri, T. Kärkkäinen, K. Majava in T. Rossi. An enhanced memetic differential evolution in filter design for defect detection in paper production. *Evolutionary Computation*, 16(4):529–555, 2008. [29](#)
- [178] L.-Y. Tseng in C. Chen. Multiple Trajectory Search for Multiobjective Optimization. V *The 2007 IEEE Congress on Evolutionary Computation CEC 2007*, strani 3609–3616. IEEE Press, 2007. [27](#)
- [179] A. Turing. Intelligent machinery (1948). *The essential Turing: seminal writings in computing, logic, philosophy, artificial intelligence, and artificial life, plus the secrets of Enigma*, stran 395, 2004. [6](#), [15](#)

- [180] A. Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950. [6](#), [15](#)
- [181] 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*, številka 4403 iz *Lecture Notes in Computer Science*, strani 257–271. Springer, 2007. [26](#)
- [182] T. Tušar, P. Korošec, G. Papa, B. Filipič in J. Šilc. A comparative study of stochastic optimization methods in electric motor design. *Applied Intelligence*, 2(27):101–111, 2007. [29](#)
- [183] J. Tvrđík. Adaptation in differential evolution: A numerical comparison. *Applied Soft Computing*, 9(3):1149–1155, 2009. [23](#)
- [184] N. Veček, M. Mernik in M. Črepinšek. A chess rating system for evolutionary algorithms: A new method for the comparison and ranking of evolutionary algorithms. *Information Sciences*, 277:656–679, 2014. [67](#)
- [185] A. Viktorin, R. Senkerik, M. Pluhacek, T. Kadavy in A. Zamuda. Distance Based Parameter Adaptation for Success-History based Differential Evolution. *Swarm and Evolutionary Computation*, Available online 12 November 2018. [36](#)
- [186] S. von Mammen in C. Jacob. Genetic Swarm Grammar Programming: Ecological Breeding Like a Gardener. V *2007 IEEE Congress on Evolutionary Computation*, urednika: D. Srinivasan in L. Wang, strani 851–858, Singapore, 25–28 September 2007. IEEE Computational Intelligence Society, IEEE Press. [44](#)
- [187] S. von Mammen in C. Jacob. The Evolution of Swarm Grammars: Growing Trees, Crafting Art and Bottom-Up Design. *IEEE Computational Intelligence Magazine*, 4(3):10–19, 2009. [44](#)
- [188] A. R. Wallace. On the Tendency of Species to form Varieties. *Journal of the Proceedings of the Linnean Society of London, Zoology* 3:53–62, 1858. [14](#)
- [189] 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*, uredniki: K. Giannakoglou, D. Tsahalis, J. Periaux, K. Papailiou in T. Fogarty, strani 167–172, Barcelona, Spain, 2001. International Center for Numerical Methods in Engineering (CIMNE). [29](#)
- [190] M. Weber, F. Neri in V. Tirronen. A Study on Scale Factor in Distributed Differential Evolution. *Information Sciences*, 181(12), 2011. [29](#)
- [191] A. Weismann. *The Germ-Plasm: A Theory of Heredity*. Charles Scribner’s Sons, 1893. [14](#)
- [192] Wikimedia. Wikipedia, the free encyclopedia. V *Wikimedia Foundation*, 2011. URL <http://en.wikipedia.org/>. [7](#)
- [193] J. Wojtusiak in R. Michalski. The lem3 implementation of learnable evolution model and its testing on complex function optimization problems. V *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, strani 1281–1288. ACM, 2006. [18](#)

- [194] F. Xue, A. C. Sanderson in R. J. Graves. Pareto-based Multi-Objective Differential Evolution. V *Proceedings of the 2003 Congress on Evolutionary Computation*, številka 2, strani 862–869, Canberra, Australia, 2003. IEEE Press. 21
- [195] X. Yang. Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-Inspired Computation*, 2(2):78–84, 2010. 18
- [196] X. Yang in S. Deb. Cuckoo search via Lévy flights. V *NaBIC 2009 World Congress on Nature & Biologically Inspired Computing*, strani 210–214. IEEE, 2009. 17
- [197] Z. Yang, J. He in X. Yao. Making a Difference to Differential Evolution. V *Advances in Metaheuristics for Hard Optimization*, urednika: Z. Michalewicz in P. Siarry, Lecture Notes in Computer Science, strani 397–414, Berlin, 2008. Springer. 22, 28
- [198] 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*, strani 3523–3530, Singapore, 25–28 September 2007. 22, 28
- [199] X. Yao, Y. Liu in G. Lin. Evolutionary Programming Made Faster. *IEEE Transactions on Evolutionary Computation*, 3(2):82–102, 1999. 16, 28
- [200] O. Yogev, A. A. Shapiro in E. K. Antonsson. Computational Evolutionary Embryogeny. *IEEE Transactions on Evolutionary Computation*, 14(2):301–325, 2010. 45
- [201] E. L. Yu in P. N. Suganthan. Ensemble of niching algorithms. *Information Sciences*, 180(15), 2010. 23
- [202] D. Zaharie. Influence of crossover on the behavior of Differential Evolution Algorithms. *Applied Soft Computing*, 9(3):1126–1138, 2009. 23
- [203] E. N. Zalta (glavni urednik). The Stanford Encyclopedia of Philosophy. V *The Metaphysics Research Lab, Center for the Study of Language and Information, Stanford University, Stanford, CA 94305-4115*, 2011. URL <http://plato.stanford.edu/>. 7
- [204] A. Zamuda. Samoprilagajanje krmilnih parametrov pri algoritmu diferencialne evolucije za večkriterijsko optimizacijo. Mag. naloga, Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru, 2008. 7
- [205] A. Zamuda. Diferencialna evolucija realnih industrijskih izzivov CEC 2011. V *Zbornik dvajsete mednarodne Elektrotehniške in računalniške konference ERK 2011, vol. B*, strani 185–188, 2011. 29
- [206] A. Zamuda. *Differential Evolution for Parameterized Procedural Woody Plant Models Reconstruction*. Ph.D. Thesis (in Slovene), University of Maribor, Faculty of Electrical Engineering and Computer Science, 2012. 43, 44, 45, 47
- [207] A. Zamuda in J. Brest. Večkriterijska rekonstrukcija numerično kodiranih proceduralnih modelov dreves z diferencialno evolucijo. V *Zbornik devetnajste mednarodne Elektrotehniške in računalniške konference ERK 2010, vol. B*, strani 169–172, 2010. 44
- [208] A. Zamuda in J. Brest. Vectorized procedural models for animated trees reconstruction using differential evolution. *Information Sciences*, 278:1–21, 2014. 29, 43

- [209] A. Zamuda in J. Brest. Self-adaptive control parameters' randomization frequency and propagations in differential evolution. *Swarm and Evolutionary Computation*, 25:72–99, 2015. [x](#), [44](#), [68](#), [69](#)
- [210] A. Zamuda, J. Brest, B. Bošković in V. Žumer. Differential Evolution for Multiobjective Optimization with Self Adaptation. V *The 2007 IEEE Congress on Evolutionary Computation CEC 2007*, strani 3617–3624. IEEE Press, 2007. [26](#), [28](#)
- [211] A. Zamuda, J. Brest, B. Bošković in V. Žumer. Večkriterijska optimizacija: eksperimentalni rezultati algoritmov *MOjDE* in *DEMO*. V *Zbornik šestnajste mednarodne Elektrotehniške in računalniške konference ERK 2007, vol. B*, strani 89–92, 2007. [25](#), [28](#)
- [212] 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*, strani 3719–3726. IEEE Press, 2008. [22](#), [28](#)
- [213] A. Zamuda, J. Brest, B. Bošković in V. Žumer. Študija samoprilagajanja krmilnih parametrov pri algoritmu DEMOWSA. *Elektrotehniški vestnik*, 75(4):223–228, 2008. [7](#), [26](#)
- [214] A. Zamuda, J. Brest, B. Bošković in V. Žumer. Differential Evolution with Self-adaptation and Local Search for Constrained Multiobjective Optimization. V *IEEE Congress on Evolutionary Computation 2009*, strani 195–202. IEEE Press, 2009. [26](#), [28](#), [65](#)
- [215] A. Zamuda, J. Brest, B. Bošković in V. Žumer. Woody Plants Model Recognition by Differential Evolution. V *The Fourth International Conference on Bioinspired Optimization Methods and their Applications, May 20 - 21 2010, Ljubljana, Slovenia*, strani 205–215, 2010. [44](#), [45](#)
- [216] A. Zamuda, J. Brest, B. Bošković in V. Žumer. Differential Evolution for Parameterized Procedural Woody Plant Models Reconstruction. *Applied Soft Computing*, 11(8):4904–4912, 2011. [29](#), [43](#), [44](#), [45](#), [47](#)
- [217] A. Zamuda, J. Brest, N. Guid in V. Žumer. Construction of Virtual Trees within Ecosystems with EcoMod Tool. V *Proceedings of IPSI-2006 Slovenia, International Conference on Advances in the Internet, Processing, Systems, and Interdisciplinary Research*, stran 15, 2006. [44](#), [45](#), [46](#), [51](#)
- [218] A. Zamuda, J. Brest, N. Guid in V. Žumer. Modelling, Simulation, and Visualization of Forest Ecosystems. V *The IEEE Region 8 EUROCON 2007: International conference on "Computer as a tool", September 9-12, 2007, Warsaw, Poland*, strani 2600–2606. IEEE Press, 2007. [45](#)
- [219] A. Zamuda, J. Brest in V. Žumer. Razpoznavna numerično kodiranih proceduralnih modelov iz slik naravnih dreves z uporabo diferencialne evolucije. V *Zbornik osemnajste mednarodne Elektrotehniške in računalniške konference ERK 2009, vol. B*, strani 171–174, 2009. [44](#)
- [220] A. Zamuda in J. D. Hernández Sosa. Differential Evolution and Underwater Glider Path Planning Applied to the Short-Term Opportunistic Sampling of Dynamic Mesoscale Ocean Structures. *Applied Soft Computing*, 24:95–108, 2014. [70](#)

- [221] A. Zamuda, M. Nicolau in C. Zarges. A black-box discrete optimization benchmarking (bb-dob) pipeline survey: taxonomy, evaluation, and ranking. V *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO 2018)*, strani 1777–1782, 2018. [73](#)
- [222] A. Zamuda, J. D. H. Sosa in L. Adler. Constrained Differential Evolution Optimization for Underwater Glider Path Planning in Sub-mesoscale Eddy Sampling. *Applied Soft Computing*, 42:93–118, 2016. [56](#), [57](#), [60](#)
- [223] A. Zamuda, J. D. H. Sosa in L. Adler. Improving Constrained Glider Trajectories for Ocean Eddy Border Sampling within Extended Mission Planning Time. V *2016 IEEE Congress on Evolutionary Computation*, strani 1727–1734, 2016. [62](#)
- [224] A. Zamuda in D. Strnad. Interaktivni modelirnik realističnih animiranih dreves. *Zbornik trinajste mednarodne Elektrotehniške in računalniške konference ERK 2004*, strani 11–14, 2004. [44](#)
- [225] A. Zamuda, A. Čep in J. Brest. Optimizacija medatomskega energijskega potenciala Lennard-Jones z diferencialno evolucijo na arhitekturi CUDA. V *Zbornik dvajsete mednarodne Elektrotehniške in računalniške konference ERK 2011, vol. B*, strani 197–200, 2011. [11](#)
- [226] A. Zamuda, C. Zarges, G. Stiglic in G. Hrovat. Stability selection using a genetic algorithm and logistic linear regression on healthcare records. V *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO 2017)*, strani 143–144, 2017. [62](#), [68](#)
- [227] A. Zamuda. Binary 100-digit challenge using IEEE-754 coded numerical optimization scenarios (100b-digit) and V-shape binary distance-based success history differential evolution (DISHv). V *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, strani 1821–1828. ACM, 2019. [36](#)
- [228] A. Zamuda. Differential evolution applicability (tutorial at 7-th joint international conferences on swarm, evolutionary and memetic computing conference (semcco 2019) & fuzzy and neural computing conference (fancco 2019), maribor, slovenia, eu, 10-12 july 2019). V *Swarm, Evolutionary, and Memetic Computing and Fuzzy and Neural Computing*, stran v, 2019. [73](#)
- [229] A. Zamuda in J. Brest. On Tenfold Execution Time in Real World Optimization Problems with Differential Evolution in Perspective of Algorithm Design. V *2018 25th International Conference on Systems, Signals and Image Processing (IWSSIP)*, strani 1–5. IEEE, 2018. [62](#)
- [230] A. Zamuda, V. Crescimanna, J. C. Burguillo, J. M. Dias, K. Wegrzyn-Wolska, I. Rached, H. González-Vélez, R. Senkerik, C. Pop, T. Cioara et al. Forecasting cryptocurrency value by sentiment analysis: An hpc-oriented survey of the state-of-the-art in the cloud era. V *High-Performance Modelling and Simulation for Big Data Applications*, strani 325–349. Springer, 2019. [49](#), [65](#), [71](#), [73](#)
- [231] A. Zamuda in J. D. H. Sosa. Success history applied to expert system for underwater glider path planning using differential evolution. *Expert Systems with Applications*, 119(1 April 2019):155–170, 2019. [x](#), [29](#), [49](#), [53](#), [55](#), [62](#), [69](#), [70](#), [71](#)
- [232] J. Zhang in A. C. Sanderson. JADE: adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation*, 13(5):945–958, 2009. [22](#), [31](#), [33](#), [36](#), [39](#)

- [233] J. Zhang in A. Sanderson. JADE: Self-adaptive differential evolution with fast and reliable convergence performance. V *Proceedings of the 2007 IEEE Congress on Evolutionary Computation (CEC)*, strani 2251–2258, 25-28 Sept. 2007. [22](#)
- [234] Q. Zhang in H. Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007. [27](#)
- [235] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu in S. Tiwari. Multiobjective optimization Test Instances for the CEC 2009 Special Session and Competition. Technical Report CES-487, University of Essex, Colchester, UK and Nanyang Technological University, Singapore, Special Session on Performance Assessment of Multi-Objective Optimization Algorithms, Technical Report, 2008. [27](#)
- [236] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan in Q. Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49, 2011. [28](#)
- [237] 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*, strani 3585–3592. IEEE Press, 2007. [26](#)
- [238] 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. [17](#)
- [239] K. Zielinski, P. Weitekemper, R. Laur in K.-D. Kammeyer. Optimization of Power Allocation for Interference Cancellation With Particle Swarm Optimization. *IEEE Transactions on Evolutionary Computation*, 13(1):128–150, 2008. [29](#)
- [240] 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*, strani 3019–3026. IEEE Press, 2007. [17](#), [27](#)
- [241] 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*, uredniki: K. C. Giannakoglou, D. T. Tsahalis, J. Périaux, K. D. Papailiou in T. Fogarty, strani 95–100, Athens, Greece, 2001. International Center for Numerical Methods in Engineering (CIMNE). [26](#)
- [242] E. Zitzler in S. Künzli. Indicator-Based Selection in Multiobjective Search. V *Parallel Problem Solving from Nature (PPSN VIII)*, urednika: X. Yao et al., strani 832–842, Berlin, Germany, 2004. Springer-Verlag. [26](#)
- [243] E. Zitzler, M. Laumanns in L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, 2001. [26](#)