

RĪGAS TEHNISKĀ UNIVERSITĀTE
Datorzinātnes un informācijas tehnoloģijas fakultāte
Informācijas tehnoloģijas institūts

Vitālijs Boļšakovs

Doktora studiju programmas „Informācijas tehnoloģija” doktorants

**SAREŽĢĪTU SISTĒMU IMITĀCIJAS MODELĒŠANĀ
SAKŅOTA PIEMĒROTĪBAS AINAVAS ANALĪZE UN
OPTIMIZĀCIJA**

Promocijas darba kopsavilkums

Zinātniskā vadītāja
Dr.habil.sc.ing., prof.
G. MERKURJEVA

RTU Izdevniecība
Rīga 2013

UDK 004.94(043.2)

Bo 385 s

Boļšakovs, V. Sarežģītu sistēmu imitācijas modelēšanā saknota piemērotības ainavas analīze un optimizācija. Promocijas darba kopsavilkums.-R.: RTU Izdevniecība, 2013.-50.lpp.

Iespiests saskaņā ar Informācijas tehnoloģijas institūta 2013.gada 25. jūnija Padomes sēdes lēmumu, protokols Nr. 13-05.



Šis darbs ir izstrādāts ar Eiropas Sociālā fonda atbalstu projekta „Atbalsts doktora studiju programmu īstenošanai” projekta „Atbalsts RTU doktora studiju īstenošanai” ietvaros.

ISBN 978-9934-10-475-6

PROMOCIJAS DARBS
IZVIRZĪTS RĪGAS TEHNISKAJĀ UNIVERSITĀTĒ
INŽENIERZINĀTŅU DOKTORA GRĀDA IEGŪŠANAI

Promocijas darbs inženierzinātņu doktora grāda iegūšanai tiek publiski aizstāvēts 2013.g. 9. decembrī plkst. 14³⁰ Rīgas Tehniskās universitātes Datorzinātnes un informācijas tehnoloģijas fakultātē, Meža ielā 1/3, 202. auditorijā.

OFICIĀLIE RECENZENTI

Dr. habil. sc. ing, profesors Zigurds Markovičs
Rīgas Tehniskā universitāte, Latvija

Dr. sc. ing., asociētais profesors Pēteris Grabusts
Rēzeknes Augstskola, Latvija

Dipl.-Ing. Dr., profesors Stefan Wagner
Upper Austria University of Applied Sciences, Austrija

APSTIPRINĀJUMS

Apstiprinu, ka esmu izstrādājis šo promocijas darbu, kas iesniegts izskatīšanai Rīgas Tehniskajā universitātē inženierzinātņu doktora grāda iegūšanai. Promocijas darbs nav iesniegts nevienā citā universitātē zinātniskā grāda iegūšanai.

Vitālijs Boļšakovs.....(Paraksts)

Datums:

Promocijas darbs ir uzrakstīts angļu valodā, satur ievadu, 4 nodaļas, secinājumus, literatūras sarakstu, 2 pielikumus, 73 attēlus un 14 tabulas, kopā 135 lappuses. Literatūras sarakstā ir 94 avoti.

VISPĀRĪGS DARBA RAKSTUROJUMS

Tēmas aktualitāte

Tradicionālās optimizācijas metodes (lineārā programmēšana, veselo skaitļu programmēšana, stohastiskā optimizācija, u.c.) dažos gadījumos nevar pielietot sarežģītu problēmu risināšanai. Tās var novest pie šādu problēmu neefektīviem risinājumiem, kam par iemeslu var būt liels optimizācijas problēmas parametru skaits, stohastisku parametru pastāvēšana un liela pārmeklēšanas telpa. Vairākas metodes tiek pielietotas kombinatorikas uzdevumu optimizācijā. Lai izvēlētos piemērotākās metodes šādu uzdevumu risināšanai, tiek pielietota piemērotības ainavas analīze. Turklāt, sistēmas imitācijas modelēšana ļauj novērtēt sistēmas veiktspēju, nepielietojot analītisko skaitļošanu. Mūsdienās imitācijas modelēšanā sakņotas optimizācijas tehnoloģija ir kļuvusi par svarīgu instrumentu sarežģītu sistēmu optimizācijā, kurām risinājuma novērtēšana var būt apgrūtināta, (Kleijnen, Merkurjev, Gosavi, Merkurjeva, Carson, Azadivar, etc.). Imitācijas modelēšanā sakņota piemērotības ainavas analīze nodrošina efektīvu pieeju optimizācijas algoritmu lietojamības noteikšanai.

Mūsdienās piemērotības ainavas analīzes metodes pielieto optimizācijas problēmu sarežģītības noteikšanai evolucionāriem algoritmiem (piemēram, Vassilev, Fogarty, Miller, Affenzeller, u.c.). Taču joprojām trūkst informācijas par esošo metožu lietošanu. Pie tam, literatūrā ir ļoti maz pētījumu par imitācijas modelēšanas pielietošanu piemērotības ainavas analīzē un sarežģītu sistēmu optimizācijā. Imitācijas modelēšanā sakņota piemērotības ainavas analīze ļaus izvēlēties labāku algoritmu sarežģītu sistēmu optimizācijai un ļaus konstruēt un pielāgot piemērotāko optimizācijas algoritmu.

Darba mērķis un uzdevumi

Darba mērķis ir izstrādāt metodes un algoritmus sarežģītu sistēmu imitācijas modelēšanā sakņotai piemērotības ainavas analīzei.

Promocijas darba mērķa sasniegšanai ir izvirzīti šādi uzdevumi:

1. Veikt eksistējošo piemērotības ainavas analīzes metožu un mēru apskatu to lietošanai NP-sarežģītu problēmu imitācijas modelēšanā sakņotai optimizācijai.
2. Izstrādāt formalizētu shēmu ar piemērotības ainavas analīzi uzlabotai imitācijas modelēšanā sakņotai optimizācijai.
3. Veikt etalonuzdevumu piemērotības ainavu analīzi, lai noteiktu ainavas tipisko struktūru ietekmi uz statistiskiem un informācijas piemērotības ainavas mēriem.
4. Izstrādāt procedūru NP-sarežģītu problēmu imitācijas modelēšanā piemērotības ainavas analīzei un optimizācijai.

5. Aprobēt izstrādātās metodes piegāžu plānošanas optimizācijas uzdevumā.

Aizstāvēšanai izvirzāmās tēzes

1. Piemērotības ainavas un to struktūru formālo definīciju noteikšana un piemērotības ainavas statistisko un informācijas analīzes metožu apskats ļauj izstrādāt formālu shēmu ar piemērotības ainavas analīzi papildinātai imitācijas modelēšanā sakņotas optimizācijas procedūrai.
2. Paplašinot piemērotības ainavas analīzes metodes uz imitācijas modelēšanā sakņotas optimizācijas uzdevumiem, ir iespējams realizēt algoritmus un programmlīdzekļus piemērotības ainavas pielietošanai sarežģītu optimizācijas uzdevumu risināšanā.
3. Etalonuzdevumu piemērotības ainavu eksperimentālā analīze ļauj noteikt sakarības starp piemērotības ainavas strukturālām īpašībām, mēriem un optimizācijas algoritmu uzvedību.
4. Lietojot izstrādātās metodes transportlīdzekļu maršrutu grafikplāna optimizācijā, ir iespējams uzlabot piegādes plānošanas operatīvā līmeņa uzdevumu risinājumus.

Pētījuma objekts un priekšmets

Darba pētījumu objekts ir NP-sarežģītu uzdevumu metaheiristiskā optimizācija.

Darba pētījumu priekšmets ir metožu un algoritmu izveidošana imitācijas modelēšanā sakņotai piemērotības ainavas analīzei un to lietošanai sarežģītu sistēmu optimizācijā.

Pētījuma metodes un teorētiskais (metodoloģiskais) pamatojums

Pētījumu veikšanai ir izmantota piemērotības ainavas statistiskā un informācijas analīze, diskrešu notikumu imitācijas modelēšana, ģenētiskais algoritms, evolūcijas stratēģija, imitācijas modelēšanā sakņota optimizācija, metaheiristiskā optimizācija un sistēmanalīze.

Promocijas darba zinātniskā novitāte

Darba zinātniskais jaunieguvums ir šāds:

1. Imitācijas modelēšanā sakņota piemērotības ainavas analīzes algoritma izstrāde.
2. Piemērotības ainavas lietošanas apraksts lietišķā pētījumā.
3. Izstrādāto metožu un algoritmu lietošana piegāžu maršrutu plānošanā un grafikplānošanā.

Darba praktiskā nozīmība

Izstrādātās metodes ir izmantotas reģionāla sadales centra produkcijas piegāžu maršrutu plānošanas un grafikplānošanas problēmas atrisināšanā. Optimizācijas problēmas, kas ir definētas integrētā plānošanas pieejā,

ir analizētas ar piemērotības ainavas analīzes tehnikām un ir sniegti secinājumi par problēmu pārmeklēšanas telpām.

Piemērotības ainavas analīze un metaheiristiski optimizācijas algoritmi ir izmantoti SIA „Havi Logistics” piegāžu no viena sadales centra uz mazumtirdzniecības veikalu lielo tīklu biznesa procesu optimizācijā. Apliecinājums, kas saņemts no SIA „Havi Logistics” valdes priekšsēdētāja, apliecina, ka algoritmi un metodes, kas ir izstrādāti promocijās darbā ir lietderīgi un izmantojami Latvijas lielajiem loģistikas uzņēmumiem.

Darba rezultātu aprobācija

Promocijas darba rezultāti tika prezentēti **12 starptautiskās zinātniskajās konferencēs:**

1. Rīgas Tehniskās universitātes 150 gadu jubilejai veltītā 53. starptautiskā zinātniskā konference un 1. pasaules inženieru un RPI/RTU absolventu kongress, apakšsekcija „Informācijas tehnoloģija un vadības zinātne”. Rīgā, Latvijā, 2012. g. 11.-12. oktobrī.
2. Starptautiskā konference “*The 24th European Modeling & Simulation Symposium*” (EMSS-2012). Vīnē, Austrijā, 2012. g. 19.-21. septembrī.
3. Starptautiskā konference “*25th European Conference on Operational Research EURO-2012*”. Viļņā, Lietuvā, 2012. g. 8.-11. jūlijā.
4. Starptautiskā konference “*1st Australian Conference on the Application of Systems Engineering*” (ACASE'12). Sidnejā, Austrālijā, 2012. g. 6.-8. februārī.
5. Starptautiskā konference “*UKSim 5th European Symposium on Computer Modeling and Simulation*” (EMS2011). Madridē, Spānijā, 2011. g. 16.-18. novembrī.
6. Starptautiskā konference “*13th European Conference on Computer Aided System Theory Eurocast-2011*”. Las Palmas de Gran Canaria, Spānijā, 2011. g. 6.-11. februārī.
7. Starptautiskā konference “*UKSIM Fourth European Modelling Symposium on Computer Modelling and Simulation*” (EMS2010). Pizā, Itālijā, 2010. g. 17.-19. novembrī.
8. Rīgas Tehniskās universitātes 51. starptautiskā zinātniskā konference. Rīgā, Latvijā, 2010. g. 13.-15. oktobrī.
9. Starptautiskā konference “*The 7th EUROSIM Congress on Modelling and Simulation*”. Prāgā, Čehijā, 2010. g. 6.-10. septembrī.
10. Starptautiskā konference “*12th International Conference on Computer Modelling and Simulation*” (UKSim2010). Kembridžā, Lielbritānijā, 2010. g. 24.-26. martā.
11. Starptautiskā konference “*1st International Conference on Intelligent Systems, Modelling and Simulation*” (ISMS2010). Liverpūlē, Lielbritānijā, 2010. g. 27.-29. janvārī.
12. Rīgas Tehniskās universitātes 50. starptautiskā zinātniskā konference. Rīgā, Latvijā, 2009. g. 14.-16. oktobrī.

Pētījumu rezultāti ir atspoguļoti **13 zinātniskajās publikācijās**, ieskaitot **1** grāmatas nodaļu izdevniecībā „Springer”, **1** publikāciju zinātniskajā žurnālā un **11** publikācijas starptautiski atzītos rakstu krājumos. Raksts „Simulation Optimisation and Monitoring in Tactical and Operational Planning of Deliveries” ieguva diplomu „The 24th European Modeling and

Simulation Symposium Best Paper Award” kā labākais konferences raksts.
Publikāciju pilnais saraksts:

1. Merkurjeva G., Bolshakov V. Simulation Optimisation and Monitoring in Tactical and Operational Planning of Deliveries // Proceedings of the European Modeling and Simulation Symposium, 2012, Austria, Vienna, 19.-21. September, 2012. - pp 226-231. Indexed in: Scopus.
2. Pitzer E., Vonolfen S., Beham A., Affenzeller M., Bolshakov V., Merkurjeva G. Structural Analysis of Vehicle Routing Problems using General Fitness Landscape Analysis and Problem Specific Measures // 1st Australian Conference on the Application of Systems Engineering (ACASE'12), Australia, Sydney, 6.-8. February, 2012. - pp 36-38.
3. Merkurjeva G., Bolshakov V. Simulation-Based Fitness Landscape Analysis and Optimisation for Vehicle Scheduling Problem // EUROCAST 2011, Part I, LNCS 6927: Springer-Verlag Berlin Heidelberg, 2012. - pp 280-286. Indexed in: SpringerLink, Scopus.
4. Merkurjeva G., Bolshakov V., Kornevs M. An Integrated Approach to Product Delivery Planning and Scheduling // Scientific Journal of RTU. 5. series., Computer Science. - 49. vol. (2011), pp 97-103. Indexed in: EBSCO, CSA/ProQuest, VINITI.
5. Bolshakov V., Pitzer E., Affenzeller M. Fitness Landscape Analysis of Simulation Optimisation Problems with HeuristicLab // Proceedings of the UKSim 5th European Symposium on Computer Modeling and Simulation, Spain, Madrid, 16.-18. November, 2011. - pp 107-112. Indexed in: IEEE CS Digital Library, Scopus.
6. Merkurjeva G., Bolshakovs V. Benchmark Fitness Landscape Analysis // International Journal of Simulation Systems, Science and Technology. - Vol.12, No.2. (2011) pp 38-45. Indexed in: Inspec, Scopus.
7. Merkurjeva G., Bolshakov V. Simulation-Based Fitness Landscape Analysis and Optimisation for Vehicle Scheduling Problem // EUROCAST 2011 Computer Aided Systems Theory Extended Abstracts. IUCTC Universidad de Las Palmas de Gran Canaria, Spain, 2011. – pp. 254-255.
8. Merkurjeva G., Bolshakovs V. Structural Analysis of Benchmarking Fitness Landscapes // Scientific Journal of RTU. 5. series., Computer Science. - 44. vol. – Riga: “RTU Publishing House”, 2010. - pp 81-86. Indexed in: EBSCO, CSA/ProQuest, VINITI.
9. Merkurjeva G., Bolshakovs V. Comparative Analysis of Statistical and Information Measures for Benchmark Fitness Landscapes // Proceedings of the UKSIM Fourth European Modelling Symposium on Computer Modelling and Simulation, Italy, Pisa, 2010. - pp 96-101. Indexed in: IEEE CS Digital Library, Scopus.
10. Merkurjeva G., Merkurjev Yu., Bolshakovs V. Simulation-Based Fitness Landscape Analysis for Vehicle Scheduling Problem // Proceedings of the 7th EUROSIM Congress on Modelling and Simulation. Czech Republic, Prague, 6.-10. September, 2010. – p. 88.
11. Merkurjeva G., Bolshakovs V. Vehicle Schedule Simulation with AnyLogic// Proc. of the 12th International Conference on Computer Modelling and Simulation UKsim-2010. Los Alamitos: “IEEE Conference Publication Service”, 2010. – pp. 169-174. Indexed in: IEEE CS Digital Library, Scopus.
12. Merkurjeva G., Bolshakovs V. Simulation-Based Vehicle Scheduling with Time Windows// Proc. of the 1st UKSim/AMSS International conference on Intelligent

Systems, Modelling and Simulation. Los Alamitos: "IEEE Conference Publication Service", 2010. – pp. 134-139. Indexed in: IEEE CS Digital Library, Scopus.

13. Merkurjeva G., Bolshakovs V. Simulation-based analysis of fitness landscape in optimisation // Scientific Journal of RTU. 5. series., Computer Science. - 44. Vol. – Rīga: "RTU Publishing House", 2009. – pp. 39-44 Indexed in: EBSCO, CSA/ProQuest, VINITI.

Promocijas darbā iegūtie rezultāti tika izmantoti šādos Latvijas Zinātnes Padomes pētījumu grantos:

1. Pētījumu grants Nr. 09.1201 "Imitācijas modelēšanā bāzēta optimizācija pielietojot skaitļošanas intelektu". Projekta vadītājs: Dr. habil. sc. ing., prof. J. Merkurjevs (2009);
2. Pētījumu grants Nr. 09.1564 "Imitācijas modelēšanas un skaitļošanas intelekta metodes loģistikas un elektronisko pakalpojumu optimizācijai". Projekta vadītājs: Dr. habil. sc. ing., prof. J. Merkurjevs (2010-2012).

Iegūtie rezultāti tika izmantoti arī RTU fundamentālo un lietišķo pētījumu projektā Nr. FLPP-2011/6 "Imitācijas modelēšanā bāzēta transportlīdzekļu kustības grafikplāna klasteru analīze un optimizācija", kur tos izmantoja SIA „Havi Logistics” integrētai piegāžu plānošanai un grafikplānošanai: transportlīdzekļu maršrutēšanai un grafikplānošanai. Projekta vadītājs: Dr. habil., Prof. G. Merkurjeva (2011).

Promocijas darbā izstrādāto metožu aprobācija tika veikta pētījumos 3 mēnešu mobilitātes programmā Augšaustrijas Lietišķo Pētījumu universitātē, Programmatūras inženierijas nodaļā (no 1. jūnija, līdz 31. augustam, Hagenbergā, Austrijā), FP7 ICT projekta Nr. FP7-248583 "UNITE – UpgradiNg ICT excellence by strengthening cooperation between research Teams in an enlarged Europe" ietvaros.

Promocijas darba struktūra

Promocijas darbs satur ievadu, 4 nodaļas, secinājumus, literatūras sarakstu, 2 pielikumus, 73 attēlus, un 14 tabulas. Promocijas darba pamattekstis ir izklāstīts uz 135 lappusēm. Literatūras sarakstā ir 94 avoti. Promocijas darba struktūra ir šāda:

Ievads satur tēmas aktualitātes pamatojumu, darba mērķa un uzdevumu formulēšanu, pētījuma objekta, priekšmeta un izmantoto pētījumu metožu aprakstu, darba zinātniskā jaunieguvuma un praktiskās nozīmības aprakstu, kā arī darba rezultātu aprobācijas izklāstu.

Pirmajā nodaļā „Literatūras apskats un problēmas nostādne” ir aprakstītas sarežģītu optimizācijas problēmu imitācijas modelēšanā sakņotas optimizācijas metodes un ir sniegts apskats piemērotības ainavas analīzes koncepcijai un tās mērķim. Nodaļā ir sniegtas piemērotības ainavas un tās struktūru formalizētas definīcijas un interpretācijas. Ir apskatītas piemērotības ainavas īpašības, kas ietekmē optimizācijas algoritma uzvedību. Nodaļā ir aprakstītas piemērotības ainavas statistikas un

informācijas analīzes tehnikas, ir piedāvāta un definēta problēmas nostādne un formalizēta shēma imitācijas modelēšanā sakņotai piemērotības ainavas analīzei.

Otrajā nodaļā „Etalonuzdevumos bāzēta piemērotības ainavas analīze” ir aprakstītas etalonuzdevumu piemērotības ainavas un veids kā piemērotības ainavas struktūras ietekmē piemērotības ainavas analīzes statistikas un informācijas analīzes metožu mērus. Nodaļā ir definētas etalonuzdevumu ainavas un ir veikta šo ainavu statistiskā un informācijas analīze. Lai noteiktu sakarību starp piemērotības ainavas mēriem un optimizācijas algoritma veiktspēju, tika veikti papildus analīzes eksperimenti, un ir sniegta etalonuzdevumu piemērotības ainavu struktūru detalizēta izpēte.

Trešajā nodaļā „Imitācijas modelēšanā sakņota piemērotības ainavas analīze un optimizācija” ir piedāvāta imitācijas modelēšanā sakņotas piemērotības ainavas analīzes procedūra un ir aprakstīti tās algoritmi. Nodaļā ir sniegta transportlīdzekļu kustības plānošanas problēmas gadījuma izpēte, kā arī ir aprakstīta tā izstrāde. Ir aprakstīti trīs transportlīdzekļu grafikplāna optimizācijas problēmas risināšanas scenāriji. Imitācijas modelēšanā sakņota piemērotības ainavas analīze un optimizācija ir veikta gan izstrādātā rīkā, gan arī pielāgota optimizācijas struktūrietvarā. Nodaļā ir piedāvātas rekomendācijas transportlīdzekļu grafikplāna ar piemērotības ainavas analīzi pastiprinātai optimizācijai.

Ceturtajā nodaļā „Metožu izmantošana preču piegādes plānošanā” ir aprakstīta metaheiristisko optimizācijas metožu lietošana apvienotās transportlīdzekļu maršrutēšanas un grafikplāna optimizācijas uzdevuma risināšanā. Nodaļā ir sniegti transportlīdzekļu maršrutēšanas problēmas formāla nostādne, optimizācijā izmantotās metodes un apskatītā gadījuma pētījuma optimizācijas eksperimentu rezultāti. Transportlīdzekļu kustības plānošanas problēmas nostādne ir pārveidota par transportlīdzekļu maršrutu grafikplāna optimizācijas uzdevumu, kas papildina klasiskās transportlīdzekļu maršrutēšanas problēmas risinājumu. Nodaļā ir aprakstītas transportlīdzekļu maršrutu grafikplāna optimizācijas metodes un eksperimenti.

Darba rezultāti un secinājumi

Literatūras saraksts

Pielikumi

PROMOCIJAS DARBA NODAĻU ĪSS IZKLĀSTS

1. Literatūras apskats un problēmas nostādne

Pirmajā nodaļā ir sniegts apskats par piemērotības ainavas analīzi metaherisitiskā optimizācijā. Aprakstītās sarežģītu sistēmu imitācijas modelēšanā sakņotas optimizācijas metodes un metaheiristiskas metodes ir identificētas kā piemērotākās promocijas darba tēmai. Ir noteikta piemērotības ainavas vieta tādā optimizācijā. Nodaļā ir aprakstītas piemērotības ainavas formalizētas definīcijas un struktūras. Nodaļa beidzas ar piedāvātu imitācijas modelēšanā sakņotas piemērotības ainavas analīzes un optimizācijas formalizētu shēmu.

NP-sarežģītu problēmu imitācijas modelēšanā sakņota optimizācija

Mūsdienīgas optimizācijas problēmas loģistikā un ražošanā ir raksturojamas ar lielu dimensiju skaitu, nenoteiktību un nelinearitāti. Līdz ar to tās prasa stohastiskas optimizācijas metodes, kas ir jaudīgākas par tradicionālajām metodēm, tādām kā nelineāras programmēšanas metodes vai klasiskie algoritmi stohastiskā programmēšanā. Promocijas darbā ir pētītas loģistikā visizplatītākās parametriskās optimizācijas problēmas. Parametriska optimizācija tiek pielietota, lai atrastu parametru kopas vērtības, kas optimizē kādu no veiktspējas mēriem, piemēram, minimizē izmaksas vai maksimizē peļņu [15]. Tā var būt definēta matemātiski šādā veidā: optimizēt $f(x_1, x_2, \dots, x_k)$, saskaņā ar kādiem lineāriem vai nelineāriem ierobežojumiem iesaistot lēmuma mainīgos x_1, x_2, \dots , un x_k , kur f ir lēmuma mainīgo funkcija.

Viens no faktoriem, kas ievērojami ietekmē optimizācijas problēmas sarežģītību, ir problēmas skaitļošanas sarežģītība. Vairākas sarežģītības klases raksturo to, kādā veidā problēmas skaitļošanas laiks vai atmiņa ir atkarīgi no ieejas datu izmēra. Problēmas NP (nedeterminēta polinoma) klasē labākos zināmos risināšanas algoritmus var atrisināt polinoma laikā ar nedeterminētu Tjūringa skaitļotāju (un savukārt tikai eksponenciālā laikā ar reāliem datoriem). Promocijas darbā galvenais uzsvars tiek likts uz NP-sarežģītām problēmām [42], kuras ir vismaz tikpat sarežģītas kā NP-pilnas problēmas, bet NP-pilnas problēmas ir sarežģītākās problēmas NP klasē.

Citi faktori, kas ievērojami ietekmē optimizācijas problēmas sarežģītību, var būt optimizētās sistēmas stohastiskā būtība un sarežģītība iegūt mērķa funkcijas analītisko formu, kas, savukārt, samazina lietojamo optimizācijas metožu skaitu.

Imitācijas modelēšanā bāzēta optimizācija ir jaudīga metodoloģija risinājumu meklēšanai sarežģītām, liela mēroga stohastiskām optimizācijas problēmām. Imitācijas modelēšanā bāzēta mērķa funkcijas novērtēšana kļūst efektīva gadījumos, kad mērķa funkcija ir nelineāra, satur varbūtiskus elementus un mērķa funkciju ir sarežģīti izteikt algebriskās izteiksmes formā.

Literatūrā ir aprakstītas vairākas skaitliskās metodes, kurām ir nepieciešama tikai mērķa funkcijas skaitliskā vērtība jebkuram risinājuma kandidātam. Šīs metodes veido dabisku izvēli reālas pasaules sarežģītu stohastisku optimizācijas problēmu risināšanai, kur mērķa funkcijas slēgta forma parasti nav zināma, bet funkcija var būt novērtēta skaitliski [15]. Skaitliskās metodes iekļauj metaheiristiskās metodes, kas ļauj atrast labus risinājumus lielām un sarežģītām optimizācijas problēmām pieņemamā laikā, pielietojot dažādas heiristiskas un stohastiskas metodes. Kaut gan metaheiristiskās metodes negarantē, ka problēmas optimālais risinājums būs atrasts, tomēr reālo problēmu lietišķajā optimizācijā ir liela interese pēc šādām metodēm [12]. Heiristikas ir metodes, kas piedāvā likumus, kā algoritmam meklēt labus risinājumus un izvairīties no sliktiem. Liels skaits metaheiristisko metožu ir īstenotas jaudīgā un elastīgā optimizācijas struktūrietvarā HeuristicLab [49].

Tradicionālās un skaitliskās optimizācijas metodes, kuras ir lietotas imitācijas modelēšanā sakņotā optimizācijā var būt sadalītas vairākās grupās:

- Gradientā bāzētas meklēšanas metodes;
- Stohastiskā optimizācija;
- Atsauksmes virsmas metodoloģija;
- Statistiskās metodes;
- Metaheiristiskās metodes.

Gradientā bāzētās meklēšanas metodes ir balstītas uz atsauksmes funkcijas gradientu. Gradients ir izmantots, lai noteiktu mērķa funkcijas formu, un lietotu deterministiskas matemātiskās programmēšanas pieejas. Stohastiskā optimizācija ietver metodes mērķa funkcijas lokālo optimumu meklēšanai ar vērtībām, kas ir stohastiskas un nav analītiskas. Statistiskās imitācijas modelēšanas optimizācijas metodes izmanto papildus informāciju par problēmu un imitācijas modeļa struktūru. Atsauksmes virsmas metodoloģija ir balstīta uz regresijas modeļa meklēšanu, kas atbilst modeļa izejas vērtībām. Lietojot regresijas analīzi, tiek iegūta regresijas funkcija, kas apraksta imitācijas modeļa izejas un izejas datu atkarību un kuru ir vieglāk optimizēt [5]. Vairākas no augstākminētajām metodēm prasa iespēju novērtēt mērķa funkcijas gradientu. Vairums no metodēm ir paredzētas nepārtrauktām optimizācijas problēmām, un tās var būt sarežģīti lietot kombinatorikas problēmām.

Promocijas darba nozīmīgākais pētījums ir metaheiristisko metožu grupa. Šīs metodes paļaujas tikai uz kandidātu risinājumu novērtēšanu un heiristiskiem likumiem, bez papildus informācijas par optimizējamo sistēmu. Šī grupa iekļauj tādas metodes kā ģenētiskais algoritms (ĢA) [14, 17], evolūcijas stratēģija (ES) [41], imitēta atkvēlināšana (angl.: *simulated annealing*) [24] un tabu meklēšanu (angl.: *tabu search*) [13].

Ģenētiskais algoritms un evolūcijas stratēģija ir populācijās bāzētu evolūcijas algoritmu pārstāvji. Tie abi ir balstīti uz dabiskas evolūcijas

konceptiju. Ģenētiskais algoritms strādā ar indivīdu populāciju, kur indivīdi attēlo risinājuma kandidātus gēnu virknes formā (hromosomā), kurā katrs gēns kodē atbilstošus risinājuma parametrus. Ģenētiskajā algoritmā iteratīvi tiek izpildīti trīs ģenētiskie operatori: selekcija, mutācijas un krustošana. Selekcijas operators ir atbildīgs par slikto indivīdu izdzēšanu no populācijas un labo indivīdu klonēšanu populācijā, balstoties uz to piemērotību, kas ir aprēķināta ar piemērotības (mērķa) funkciju. Krustošanas operators nejausā veidā apmaina hromosomu daļas diviem nejauši izvēlētiem indivīdiem, lai iegūtu jaunus risinājumu kandidātus. Mutācija nejausā veidā izmaina gēnu vērtības daļā no nejauši izvēlētiem indivīdiem.

Evolūcijas stratēģija ir līdzīga ĢA, bet tai atšķiras indivīdu selekcija jaunajā populācijā un ir lielāka loma mutācijas operācijai. Tā saucamajā (μ, λ) evolūcijas stratēģijā sākuma populācija katrā iterācijā satur μ indivīdus, kuri tiek izmantoti, lai ar evolūcijas (mutācijas) operāciju iegūtu pēcnācēju populāciju ar λ risinājumiem. Tad pēcnācēju populācija ir samazināta līdz μ labākajiem risinājumiem, un jaunā iterācija ir izpildīta. $(\mu+\lambda)$ stratēģijā pēcnācēju populācijas λ indivīdi ir apvienoti vienā populācijā ar priekšnācējiem, un jau kopējā populācija ir samazināta līdz μ vislabākajiem risinājumiem.

Imitētā atkvēlināšana un tabu meklēšana ir lokālas pārmeklēšanas metaheiristiskās metodes. Abas metodes ir balstītas uz kalnā kāpšanas (angl.: *hill-climbing*) meklēšanas metodi. Tā kā kalnā kāpšanas metodes var iestrēgt uz lokāliem optimāliem risinājumiem, tad šīm metaheiristikām ir papildus īpatnība, kas ļauj izvairīties no šādas problēmas. Imitētā atkvēlināšana ļauj izvēlēties sliktākus risinājumus, piemērotības vērtības samazināšanas atļautā starpība algoritma darbības laikā saruks līdz nullei. Tabu meklēšanai ir speciāli aizliegumu saraksti, kas aizliedz pārmeklēt risinājumus, kas jau bija apmeklēti un līdz ar to piespiež algoritmu meklēt risinājumus jaunās vietās.

Literatūrā metaheiristiskās metodes rāda labus rezultātus kombinatorikas, analītisku optimizācijas problēmu risināšanā, un tās ir viegli kombinēt ar imitācijas modelēšanu. Līdz ar to arī galvenā uzmanība promocijas darbā tiek vērsta uz tām. Metaheiristisku un citu skaitlisko metožu pielietošana kļūst svarīgāka īpaši sarežģītām problēmām, tādām kā NP-sarežģītas optimizācijas problēmas [10].

Visi sarežģītības faktori noved pie laikietilpīgas optimizācijas un bieži vien sarežģītas optimizācijas problēmas nevar atrisināt saprātīgā laikā. Neparastām un pietiekami neizpētītām optimizācijas problēmām tas var novest pie komplicētu lēmumu pieņemšanas labākas optimizācijas metodes izvēlei un konfigurēšanai. Eksperimentālā algoritma pielāgošana var būt neefektīva, jo iepriekš minēto sarežģīto problēmu optimizācija ir laikietilpīga.

Lai optimizācijas metodes izvēli un pielāgošanu padarītu pamatotāku, piemērotības ainavas analīze piedāvā metodes problēmas pārmeklēšanas

telpas izpētei. Šīs analīzes metodes arī var pielietot dažādu metaheiristisku algoritmu optimizācijas uzvedības izpētei.

Piemērotības ainavas analīzes koncepcija

Piemērotības ainavas analīze piedāvā metodes un pieejas sarežģītu kombinatorisku un nepārtrauktu optimizācijas problēmu pārmeklēšanas telpas matemātiskai analīzei. Tās var lietot komplicētas sistēmas optimizācijas atbalstam un uzlabošanai, un ir plaši aprakstītas literatūrā par mākslīgo intelektu [19, 21, 27, 44, 50]. Vispārīgā gadījumā piemērotības ainava ir interpretēta [20, 39, 45] kā optimizācijas problēmas piemērotības funkcijas kombinācija ar saitēm vai ar attāluma metriku starp pārmeklēšanas telpas risinājumiem. Tādējādi, piemērotības ainava nosaka pārmeklēšanas telpas struktūru.

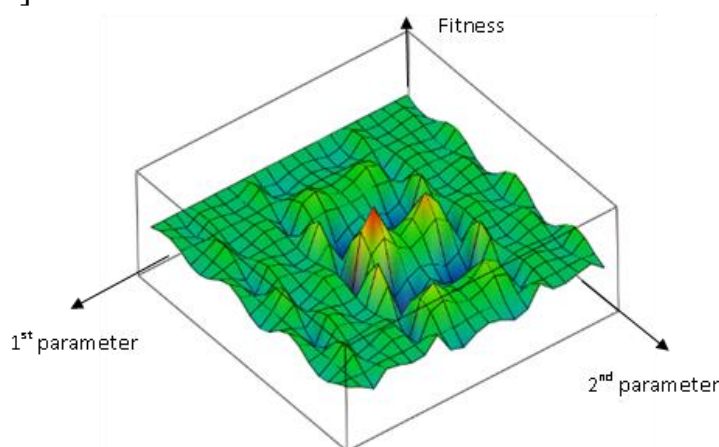
Pirmo reizi piemērotības ainavas jēdziens ir pieminēts [52] kā rīks bioloģiskas evolūcijas uzvedības interpretācijai. Evolūcija ir izskatīta kā apstaiga uz ainavas virsmas, kuras augstums nosaka piemērotības pakāpi indivīdiem, kas atrodas noteiktajā ainavas punktā. Evolūcijai ir tendence virzīties uz augstākām virsotnēm, kur indivīdi ir visvairāk piemēroti. Tāda piemērotības ainavas definīcija rāda, ka dažreiz evolūcijai ir jāpārvieta indivīdus no vienas virsotnes uz citu caur ieleju ar zemu piemērotību, un šādas struktūras ietekmē evolūcijas procesu.

Šāda piemērotības ainavas koncepcija tika pārnesta optimizācijas problēmu sfērā, kur tika piedāvāts, ka problēmas piemērotības ainavas struktūras ietekmē to, kādā veidā šī pārmeklēšanas telpa tiek pārmeklēta ar heiristisko algoritmu. Tika piedāvāts, ka problēmas piemērotības analīze ļaus iegūt vairāk informācijas par optimizācijas algoritmu specifiskām problēmu īpašībām, un tas ļaus virzīt optimizācijas procesu [19, 39, 50].

Viens no piemērotības ainavas analīzes mērķiem ir optimizācijas problēmu sarežģītības noteikšana. Piemērotības ainavas īpašības, kas ietekmē problēmas sarežģītību, ir tās neģludums, modalitāte un epistasis. Šīs īpašības var pielietot meklēšanas algoritma izvēlei [21]. Vēl viens viedoklis ir, ka piemērotības ainavas analīzi var pielietot, lai iegūtu izpratni par algoritma veikspējas atšķirību dažādiem vienas problēmas eksemplāriem un atbilstoši konfigurētu optimizācijas algoritmu atsevišķiem problēmas eksemplāriem [37]. Ar piemērotības ainavas analīzi ir iespējams iegūt problēmas sarežģītības mērus un tas, savukārt, ļauj rekomendēt optimizācijas algoritma konfigurāciju, un, tajā pat laikā, iegūt dziļāku izpratni par algoritma uzvedību attiecīgajai problēmas klasei. Kaut gan piemērotības ainavas analīze var būt vairāk resursu patērējoša, nekā turpmāka optimizācija, bet tomēr tā piedāvā vairāk informācijas par optimizācijas problēmas iekšējām īpašībām. Tādā veidā piemērotības ainavas analīze ir jaudīgs instruments optimizāciju problēmu klašu padziļinātai izpratnei, bet - vājš instruments atsevišķas problēmas pastiprinātai optimizācijai [37].

Piemērotības ainavas analīzi var izmantot, kā rīku līdzīgu optimizācijas problēmu sadalīšanai vairākās klasēs un apakšklasēs ar identiskām piemērotības ainavas raksturīpašībām, kas atbilstoši nosaka identiskas grūtības optimizācijas algoritmam. Problēmu apakšklašu atbilstošu algoritmu un to konfigurāciju meklēšana piedāvā zināšanas par problēmas risināšanas scenārijiem [37].

Interesanti būtu attēlot optimizācijas problēmas piemērotības ainavu grafiski, kā, piemēram, trīsdimensiju attēlu, bet, protams, tas ir iespējams tikai problēmām ar diviem nepārtrauktiem parametriem (sk. 1. att.). Piemērotības ainava jebkurā gadījumā var būt aprakstīta [26] kā orientēts grafs, kuram visas virsotnes atbilst risinājumiem un ir iezīmētas ar piemērotības vērtību, bet loki attēlo kaimiņattiecības starp attiecīgiem risinājumiem [46].



1. att. Trīsdimensiju piemērotības ainavas interpretācija

Ir jāņem vērā, ka literatūrā par heuristisku optimizāciju, algoritma uzvedība pārmeklēšanas telpā bieži ir saistīta ar piemērotības ainavas ģeogrāfisko metaforu, tomēr, šī koncepcija bieži vien nav pamatota ar formalizāciju un var būt maldinoša [37].

Lai izvairītos no interpretācijas problēmām, literatūrā ir piedāvātas dažas formalizētas piemērotības ainavas definīcijas. Visām definīcijām kopīgs ir tas, ka tās definē piemērotības ainavu kā piemērotības funkcijas un pārmeklēšanas telpas topoloģiskās struktūras kombināciju.

Sekojošās formālajās definīcijās ir pielietots tā saucamo multikopu [25] (angl.: *multiset*) jēdziens. Ir arī definēta tāda multikopa $M(S)$ kā bezgalīga kopa no visiem multikopas S elementiem. Turklāt $M_q(S)$ ir visu multikopu kopa, kas ir iegūta no S un kuriem kopas apjoms ir vienāds ar q [20].

Problēmas *objektu telpa* ir definēta ar O . Meklēšanas algoritmam ir nepieciešams izvēlēties atbilstošu objektu no visiem O objektiem, kas var būt jebkura veida struktūras: korteži, permutāciju kopas, u.tml. *Reprezentāciju telpa* R nosaka tādu *reprezentāciju* kopu, kas attēlo objektus no O telpas. Attēlojums starp O un R tiek dēvēts par reprezentāciju.

Reprezentācija starp kopām \mathbf{O} un \mathbf{R} ir definēta ar relāciju Γ . Notācija $o\Gamma r$ definē elementiem $o \in \mathbf{O}$ un $r \in \mathbf{R}$ to, kā o ir reprezentēts ar r . Pretēja relācija Γ^{-1} (reprezentē) starp kopām \mathbf{R} un \mathbf{O} ir definēta kā $\Gamma^{-1} = \{(r, o) \mid (o, r) \in \Gamma\}$, kur notācija $r\Gamma^{-1}o$ apzīmē, ka r reprezentē o . Ja $\Gamma(o) \neq \emptyset$, tad objekts o ir reprezentēts. Ja $\Gamma^{-1}(r) = \emptyset$ tad reprezentācija nav atļauta.

Problēmai, kurai var piemērot piemērotības ainavas metaforu, ir jāsatur funkciju $g : \mathbf{O} \rightarrow \mathbf{G}$, kas nosaka risinājumu kvalitāti. Šeit ar \mathbf{G} ir apzīmēta tāda kopa, kurai ir daļēja kārtība $>_{\mathbf{G}}$ un kurai ir noteikts, ka, ja $o_1, o_2 \in \mathbf{O}$ un $g(o_1) >_{\mathbf{G}} g(o_2)$, tad o_1 ir labāks problēmas risinājums, nekā o_2 .

Laika momentā t algoritms operē ar \mathbf{R} elementu galīgo skaitu, kas veido ar \mathbf{C} apzīmētu multikopu. Multikopas \mathbf{C} modificēšanai algoritms izmanto *operatorus*. Vispārīgais operators būs apzīmēts ar ϕ .

Operators ir uzdots kā funkcija: $\phi: \mathbf{M}(\mathbf{R}) \times \mathbf{M}(\mathbf{R}) \rightarrow [0..1]$. Risinājumiem $v, w \in \mathbf{M}(\mathbf{R})$ vērtība $\phi(v, w) = p$ nosaka varbūtību p , ka, pielietojot stohastisko procedūru, kas definēta ϕ operatoram, risinājums v būs modificēts uz w .

Tad meklēšanas algoritms vispārīgā veidā var būt attēlots kā operatoru secība [20], kas modificē izvēlēto risinājumu multikopas:

$$C_0 \xrightarrow{\phi_1} C_1 \xrightarrow{\phi_2} C_2 \xrightarrow{\phi_3} \dots \quad (1)$$

Lai definētu ainavu, ϕ operatoram ir jādefinē ϕ -apkaime. Risinājuma $v \in \mathbf{M}(\mathbf{R})$ ϕ -apkaime ir kopa $N_{\phi}(v)$, kas sastāv no $\mathbf{M}(\mathbf{R})$ elementiem, kuri ir sasniedzami no v ar vienu ϕ operatora iterāciju. Citiem vārdiem, kaimiņu risinājumu kopa ir definēta šādi [20]:

$$N_{\phi}(v) = \{w \in \mathbf{M}(\mathbf{R}) \mid \phi(v, w) > 0\}. \quad (2)$$

Ja risinājums $w \in N_{\phi}(v)$, tad w ir v ϕ -kaimiņš. Pielietot ϕ apkaimes notācijā ir būtiski, jo ϕ -kaimiņi vienam operatoram var nebūt kaimiņi citiem operatoriem.

Kopas $P \subseteq \mathbf{M}(\mathbf{R})$ apkaime $N_{\phi}(P)$ ir tāda kopas $\mathbf{M}(\mathbf{R}) - P$ elementu kopa, kas ir P elementu ϕ -kaimiņi:

$$N_{\phi}(P) = \{w \in \mathbf{M}(\mathbf{R}) - P \mid \phi(v, w) > 0 \text{ un } v \in P\}. \quad (3)$$

Ir nepieciešams izvēlēties noteiktu mērķa funkciju $f: \mathbf{M}(\mathbf{R}) \rightarrow \mathbf{F}$, kas ir attēlota uz kādas kopas \mathbf{F} ar definētu daļēju kārtību $>_{\mathbf{F}}$. Tā noteiks, kādi risinājumi ir labāki pašreizējā risinājumu multikopā, un noteiks, kurā virzienā algoritmam ir jāmeklē. Tādēļ, ja $v, w \in \mathbf{M}(\mathbf{R})$ un $f(v) >_{\mathbf{F}} f(w)$, tad meklēšanas turpinājumam multikopa v ir labāka nekā w . Visbiežāk funkcija f ir saistīta ar risināmās problēmas mērķa funkciju g , tādā veidā, ka $f = g$, $\mathbf{F} = \mathbf{G}$ un $>_{\mathbf{F}} = >_{\mathbf{G}}$. Bet tāda ekvivalence nav obligāta. Tā, piemēram, lietojot ģenētiskos algoritmus, bieži vien funkcija f iekļauj sevī problēmas mērķa funkciju g un papildus funkciju, kas nosoda neatļautus risinājumus.

Precīzāka piemērotības ainavas definīcija ir sniegta [20], kur ainava ir definēta ar 5 vērtību kortežu:

$$\mathbf{L} = (\mathbf{R}, \phi, f, \mathbf{F}, >_{\mathbf{F}}), \quad (4)$$

kuras komponentes ir šādas: reprezentāciju telpa \mathbf{R} , operators ϕ , kas veido ainavu, piemērotības funkcija $f : \mathbf{M}(\mathbf{R}) \rightarrow \mathbf{F}$, kas attēlo reprezentācijas multikopu uz kopu \mathbf{F} , kas ir dēvēta par piemērotības telpu, un katrai ir noteikta daļēja kārtība $>_{\mathbf{F}}$.

Ainava šeit ir metafora, kas ir izmantota, lai iegūtu vairāk informācijas par meklēšanas algoritmu. Ainavas 5-elementu kortežs var būt attēlots orientēta iezīmēta grafa $G_L = (V, E)$ veidā, kur virsotnes ir $V \subseteq \mathbf{M}(\mathbf{R})$ un loki ir $E \subseteq V \times V$, kur loks ir definēts, ja $(v, w) \in E \Leftrightarrow \phi(v, w) > 0$. Tādā attēlojumā virsotne $v \in V$ ir iezīmēta ar $f(v)$ un loks (v, w) ar $\phi(v, w)$. Kamēr ainavas formālā definīcija ir kortežs no 5 elementiem, ainava var būt interpretēta kā grafs, kas ir iegūts no tāda korteža [20].

Tādā ainavas definīcijā lokālas smailes termins nevar būt definēts ārpus operatora apkaimes definīcijas. Risinājums ir smaile vienam operatoram, bet var nebūt smaile citam. Tādējādi ir sarežģīti izmantot koncepciju par mutācijas operatora lokāliem optimumiem, analizējot krustošanas operatora uzvedību.

Bez tam, pielietojot minētu modeli piemērotības ainavas aprakstam, nevar runāt par to, ka ĢA meklējot risinājumu veido iešanu ainavā. Visticamāk ģenētiskais algoritms secīgi veic mazus soļus mutācijas, krustošanas ainavās un tad ar visu populāciju atbilstošus soļus selekcijas ainavā, kas būtiski atšķir ĢA no kalnā kāpšanas algoritma, kas patiešām veic tādus soļus mutācijas algoritma ainavā [18].

Literatūrā [37] pastāv arī vienkāršākas ainavas definīcijas. Tādā definīcijā būtiskākās ainavas sastāvdaļas ir piemērotības funkcija $f : \mathbf{R} \rightarrow \mathbb{R}$ kas piešķir piemērotības vērtību indivīdam, un sakarības starp dažādiem risinājuma kandidātiem. Definīcijā ir pielietots attālums starp risinājumiem $d : \mathbf{R} \times \mathbf{R} \rightarrow \mathbb{R}$, kas veido metriku, atšķirībā no kaimiņu risinājuma funkcijas $N : \mathbf{R} \rightarrow 2^{\mathbf{R}}$. Piemērotības ainava \mathbf{L} ir definēta kā divu funkciju f un d kopa, kas nosaka piemērotības vērtību un attālumu starp \mathbf{R} telpas risinājumiem:

$$\mathbf{L} = (\mathbf{R}, f, d). \quad (5)$$

Apskatītā definīcijā operators un piemērotības telpas ir izlaistas, kamēr piemērotības funkcija ir attēlota uz reālu skaitļu kopu. Apkaimes definīcija (ε -apkaimē) risinājumam x tādā notācijā ir noteikta kā:

$$N_{\varepsilon}^{+}(x) = \{n \mid n \in \mathbf{R}, n \neq x, d(x, n) \leq \varepsilon\}. \quad (6)$$

Prasot attāluma mēru kā papildinājumu piemērotības vērtībai, piemērotības ainava kļūst atkarīga ne tikai no problēmas, bet arī no reprezentācijas metodes izvēles un no tā, kādā veidā risinājumi ir savienoti savā starpā pie dažādiem operatoriem [37].

Atkarībā no interpretācijas piemērotības ainava var saturēt elementus un struktūras, kas bieži ir nosauktas līdzīgi naturālo ainavu struktūrām. Tādā veidā ir definētas virsotnes vai smailes, kurām piemērotība ir augstāka, nekā kaimiņu risinājumiem. Līdzīgi varētu noteikt virsotņu grēdas, ielejas un citus līdzīgus objektus.

Sekojošas formalizētas definīcijas ir sniegtas gadījumam, kad piemērotības ainava ir noteikta ar kortežu (4) un vairākas definīcijas prasa pilnu kārtību $>_F$ piemērotības telpā. Pamata definīcijas ir ϕ -kaimiņa un ϕ -apkaimes definīcijas, kas ir minētas iepriekš.

Virgotne ir ϕ -smaile (vai ϕ -maksimums) tikai tādā gadījumā, ja tās piemērotība ir labāka, nekā jebkuram ϕ -kaimiņam [20], un ir definēta ar:

$$v \in V \mid \forall w \in N_\phi(v), f(v) >_F f(w). \quad (7)$$

Vienkāršotā interpretācijā (5) un ar ε -apkaimes definīciju lokālais optimums ir dots kā [37]:

$$\text{local optimum}(x) :\Leftrightarrow (\exists \varepsilon > 0) (\forall n \in N_\varepsilon^+(x)) f(x) > f(n). \quad (8)$$

Virgotne ir *globālais-maksimums* tad, ja tā piemērotība nav sliktāka par jebkura cita iespējamā risinājuma piemērotību visā ainavā. Tāda grafa virgotne v ir definēta kā:

$$v \in V \mid \forall w \in V, f(v) >_F f(w). \quad (9)$$

Savukārt, *ϕ -lokālais-maksimums* vai *ϕ -lokālais-optimums* ir tāda ϕ -smaile, kura nav globālais maksimums.

Ainavas reģions, kurā visi risinājumi ir ar vienādu piemērotību, tiek saukts par *ϕ -plato*, un tā ir sekojoša virgotņu kopa:

$$M \subseteq V, |M| > 1 : \forall v_0, v_n \in M, \exists v_1, \dots, v_{n-1} \text{ priekš} \\ f(v_i) = f(v_{i+1}) \text{ un } v_{i+1} \in N_\phi(v_i) \quad \forall 0 \leq i < n. \quad (10)$$

Bieži ainavas analīzes rezultātu interpretācijā ir lietots pievilksšanas baseina termins. Kādai virgotnei tā ir virgotņu kopa, no kurām dotā virgotne var būt sasniegta iteratīvi pielietojot operatoru. Formāli *ϕ -pievilksšanas-baseins* virgotnei v_n ir sekojoša virgotņu kopa:

$$B_\phi(v_n) = \{v_0 \in V \mid \exists v_1, \dots, v_{n-1} \text{ kur } v_{i+1} \in N_\phi(v_i) \quad \forall 0 \leq i < n \}. \quad (11)$$

Operators ϕ ir fiksētas kardinalitātes operators $\phi_{k \rightarrow l}$, ja tas ir definēts izvēlētām konstantēm k un l kā $M_k(\mathbf{R}) \times M_l(\mathbf{R}) \rightarrow [0..1]$.

Operatori, kas ir noteikti kā $\phi_{k \rightarrow k}$ tiek dēvēti par apstaigājošiem operatoriem. Apstaigājoša operatora izeja var būt izmantota kā šī operatora ieejas vērtība nākamajai iterācijai. Tādā veidā ir iespējams veikt apstaigas pa piemērotības ainavu. Tā, piemēram, mutācijas operatori, galvenokārt, ir apstaigājošie. Turklāt, operators ϕ tiek dēvēts par simetrisku, ja $\phi(v, w) = \phi(w, v)$ jebkurām divām virgotnēm v un $w \in V$. Tādā veidā varbūtība pāriet no viena risinājuma uz blakus risinājumu un ir vienāda ar pretējas pārejas varbūtību. Evolūcijas operatori galvenokārt ir simetriski. Par dabisku piemērotības ainavu tiek dēvēta ainava, kas ir formēta ar fiksētas kardinalitātes simetrisku operatoru, kas pieder klasei $\phi_{l \rightarrow l}$.

Vairāki faktori, kas ietekmē optimizācijas problēmas sarežģītību optimizācijas algoritmam un kurus ņem vērā optimizācijas algoritma izvēlē un konfigurēšanā, ir definēti literatūrā. Galvenie no tiem ir modalitāte, negludums, neitralitāte un epistāze [39]. *Epistāze* nosaka mijiedarbību starp gēniem hromosomā, kad indivīda piemērotība ir atkarīga no gēnu

kombinācijas mijiedarbības [4, 26, 39]. *Modalitāte* nosaka optimumu skaitu un blīvumu pārmeklēšanas telpā [39]. *Negludums* raksturo visu ainavas struktūru ietekmi uz meklēšanas sarežģītību. Piemērotības ainava ir negluda, ja tai ir liels smaiļu skaits, un ir maza korelācijas starp blakus risinājumiem [33, 48]. *Neitralitāte* raksturo tādu struktūru daudzumu ainavā, kurām visi risinājumi struktūrā ir ar vienādu (vai ļoti tuvu) piemērotību [40, 43]. Tā novērtē tādu struktūru (plato) skaitu un izmēru, jo algoritms var būt iestrēdzis neitrālos reģionos.

Faktori, kas ietekmē piemērotības ainavas struktūru, un kurus ir jāanalizē izvēloties, konstruējot vai pielāgojot algoritmu, ir šādi: problēmas sarežģītība, risinājumu reprezentācija, meklēšanas operatori un piemērotības funkcija. Problēmas sarežģītība ir vienīgais faktors, kuru nevar izmainīt, nemainot optimizācijas problēmas nostādni.

Risinājuma attēlošanai piemērotības ainavā ir iespējams pielietot vairākas reprezentācijas metodes, bet reprezentācija ietekmē reprezentācijas telpu un piemērotības ainavas topoloģisko struktūru. Lietojot dažādas kodēšanas metodes iegūst dažādas piemērotības ainavas īpašības un raksturlielumus [26]. Meklēšanas operators būtiski ietekmē piemērotības ainavas topoloģisko struktūru, un katrs operators definē savu piemērotības ainavu. Piemērotības funkcijas izvēle nemaina ainavas topoloģiju, bet ietekmē pārmeklēšanas algoritma uzvedību.

Piemērotības ainavas analīzes metožu apskats

Literatūrā [6, 8, 16, 18, 21, 43, 44, 46, 47, 48] ir apskatītas vairākas piemērotības ainavas analīzes tehnikas, kas ir bāzētas uz ainavas strukturālo raksturlielumu novērtēšanu. Šīs tehnikas ļauj analizēt piemērotības ainavu, analizējot tikai nelielu ainavas daļu, un var būt sadalītas divās grupās: statistiskā un informācijas analīze. Statistiskās analīzes tehnikas izmanto savāktos statistiskos datus, lai novērtētu korelāciju starp ainavas blakus risinājumiem. Informācijas tehnika un mēri ir bāzēti uz informācijas teoriju.

Darbā aprakstītās metodes ir bāzētas uz vienkāršiem soļiem. Šādu soļu virkne uz ainavas izveido trajektoriju, ko tālāk analizē. Lai iegūtu šādu trajektoriju, darbā ir pielietotas sekojošas trīs apstaigas metodes: gadījuma apstaiga, adaptīvā apstaiga, neitrālā apstaiga. Lielākā ainavu analīzes metožu daļa ir bāzēta uz gadījuma apstaigu, kurā risinājums ir modificēts iteratīvi ar attiecīgu operatora varbūtību. Adaptīvajā apstaigā, apstaigas virziens ir izvēlēts tā, lai uzlabotu (vai pasliktinātu) nākamā risinājuma piemērotību [22]. Viens no adaptīvo apstaigu veidiem – apstaigā uz leju-uz augšu šis virziens tiek mainīts vairākas reizes, kad apstaiga sasniedz lokālo minimumu vai maksimumu [48]. Neitrālā apstaiga tieši pretēji prasa, lai trajektorijas nākamā risinājuma piemērotība būtu pēc iespējas līdzīgāka ar iepriekšējo, un ir paredzēta ainavas plakano reģionu analīzei [40].

Statistiskā analīzes tehnikā ir analizēta autokorelācijas funkcija, kas ir iegūta gadījuma apstaigā [50]. Ainavas negludums tiek mērīts ar korelāciju. Pie lielas korelācijas starp divām risinājumu kopām, kas ir atdalītas ar noteiktu attālumu, piemērotības ainava tiek uzskatīta par gludu. Un pretēji, pie zemas korelācijas ainava ir negluda.

Statistiskajā analīzē, gadījuma apstaigā ir iegūta piemērotības vērtību laika rinda $f_{t=1}^N$. Tādas laika rindas autokorelācijas funkcija ir definēta, kā:

$$\rho(\Gamma) \approx \frac{E(f_t f_{t+s}) - E(f_t)E(f_{t+s})}{V(f_t)}, \quad (12)$$

kur atbilstoši $E(f_t)$ un $V(f_t)$ ir laika rindas $\{f_t\}_{t=1}^N$ matemātiskā cerība un dispersija, Γ ir attālums starp risinājumiem, un N ir secības garums [46]. Gludām ainavām autokorelācijas vērtība ir tuvu 1, un tā samazinās līdz nullei negludām ainavām [39].

Vēl viens mērs ir korelācijas attālums, kas nosaka attālumu starp risinājumiem, kad tie kļūst savstarpēji nekorelēti un [50] tiek izskaitļoti ar:

$$\tau = \frac{1}{\ln(\rho(1))}, \quad (13)$$

kur $\rho(1)$ ir autokorelācijas funkcija starp kaimiņu risinājumiem. Jo lielāks ir korelācijas attālums, jo negludāka ir ainava. Bet formula (13) ir pielietojama tikai pie pieņēmuma, ka laika rindas gadījuma apstaiga ir stacionārs gadījuma process.

Literatūras avotā [18] aprakstītais korelācijas attālums paredz, ka korelācijas attālums ir nozīmīgs, tikai, ja tas pārsniedz divas standarta kļūdas robežas $(-2/\sqrt{N}; +2/\sqrt{N})$, kur N ir laika rindas garums, un korelācijas attālums τ ir par vienu mazāks, nekā attālums, kad autokorelācija $\rho(\tau+1)$ būs kļūdas robežā.

Informācijas analīzes tehnika analizē piemērotības ainavu kā dažādu objektu ansamblis, un ir analizēta šo objektu forma, izmērs un sadalījums. Šādi objekti sastāv no ainavas risinājuma, un to kaimiņ-risinājumiem [48]. Šajā tehnikā informācijas saturs ir mērs, kas raksturo, cik informācijas ir nepieciešams, lai rekonstruētu sistēmu (entropija). Literatūrā [48] ir piedāvāti četri ainavas informācijas mēri:

1. informācijas saturs $H(\epsilon)$;
2. daļējs informācijas saturs $M(\epsilon)$
3. informācijas stabilitāte ϵ^* ;
4. blīvuma-baseina informācija $h(\epsilon)$.

Liela informācijas satura vērtība norāda uz augstu entropiju sistēmā, un attiecīgi to, ka ainavā ir lielāka struktūru daudzveidība un tā ir negluda. Visi informācijas mēri tiek aprēķināti ar jutīguma parametru ϵ , kas nosaka mazāku starpību starp kaimiņ-risinājumu piemērotības vērtībām. Struktūras ar mazāku starpību ir pieņemtas kā plakanas [48]. Daļējs informācijas saturs raksturo trajektorijas modalitāti [46]. Informācijas stabilitāte nosaka lielāku

kaimiņ-risinājumu piemērotības vērtību starpību. Blīvuma-baseina informācija raksturo ainavas plakano un negludo struktūru daudzveidību.

Lai aprēķinātu informācijas mērus, pirmkārt, ar ainavas apstaigu no ainavas L ir iegūta piemērotības vērtību laika rinda $\{f_{i=0}^N\}$, kur N ir apstaigas garums. Tad, piemērotības vērtību virkne tiek pārveidota ansambļu virknē $S(\varepsilon)$, kas sastāv no elementiem $s_i \in \{\bar{1}, 0, 1\}$. Katrs elements s_i jebkurai ε ir aprēķināts ar:

$$s_i(\varepsilon) = \begin{cases} \bar{1}, & \text{if } f_i - f_{i-1} < -\varepsilon \\ 0, & \text{if } |f_i - f_{i-1}| < \varepsilon \\ 1, & \text{if } f_i - f_{i-1} > \varepsilon \end{cases} \quad (14)$$

Parametrs ε ir jebkurš reāls skaitlis starp 0 un attālumu starp vismazāko un vislielāko piemērotības vērtību analizētajā laika rindā.

Informācijas varbūtība $H(\varepsilon)$ tiek aprēķināta ar [48]:

$$H(\varepsilon) = - \sum_{p \neq q} P_{[pq]} \log_6 P_{[pq]}, \quad (15)$$

kur $P_{[pq]}$ ir elementu s ansambļu pq sastapšanas varbūtība virknē starp visiem ansambļiem. Daļējas informācijas saturs tiek noteikts ar [33, 48]:

$$M(\varepsilon) = \frac{\Phi_s(1, 0, 0)}{n}, \quad (16)$$

kur n ir virknes $S(\varepsilon)$ garums, bet funkcija Φ ir aprēķināta rekursīvi ar:

$$\Phi_s(i, j, k) = \begin{cases} k & , \text{ja } i > n \\ \Phi_s(i+1, i, k+1), & \text{ja } j = 0 \text{ un } s_i \neq 0 \\ \Phi_s(i+1, i, k+1), & \text{ja } j > 0, s_i \neq 0 \text{ un } s_i \neq s_j \\ \Phi_s(i+1, j, k) & \text{citādi} \end{cases} \quad (17)$$

Blīvuma-baseinu informācija tiek aprēķināta ar [48]:

$$h(\varepsilon) = - \sum_{p \in \{\bar{1}, 0, 1\}} P_{[pp]} \log_3 P_{[pp]}, \quad (18)$$

kur varbūtība $P_{[pp]}$ attēlo apakšbloku pp biežumu virknē $S(\varepsilon)$.

Informācijas stabilitāte ε^* ir zemākā ε vērtība, pie kuras visa apstaigas trajektorija tiks uzskatīta kā plakana.

Tās problēmas, kurām piemērotības ainavas informācijas mēru vērtības ir tuvākas 0, tiek uzskatītas kā vieglas, bet sarežģītām problēmām šīs vērtības tiecas uz 1 [48].

Gan statistiskā, gan informācijas analīze ir izmantojamas tikai pie pieņēmuma, ka ainava ir statistiski izotropa. Tas nozīmē, ka gadījuma apstaiga uz analizējamās ainavas veido stacionāru gadījuma procesu [48].

Problēmas formalizēta nostādne

Lai paplašinātu piemērotības ainavas koncepciju uz imitācijas modelēšanā sakņotu optimizāciju un lai iegūtu analīzes praktisku pielietojamību tādā optimizācijā, darbā ir piedāvāta imitācijas modelēšanā sakņotas piemērotības analīzes koncepcija.

Imitācijas modelēšanas piemērotības ainava formalizēti ir definēta kā L' , kas ir esošās [20] definīcijas (4) paplašinājums, aizvietojot piemērotības funkciju f ar imitācijas modeli. Formalizētā definīcija ir šāda:

$$L' = (\mathbf{R}, \phi, S, \mathbf{F}', <_{\mathbf{F}'}), \quad (19)$$

kur \mathbf{R} ir reprezentāciju telpa, ϕ ir meklēšanas operators, S ir imitācijas modelis, \mathbf{F}' ir iespējamo imitācijas modeļa izeju kopa, kurai piemīt ar $<_{\mathbf{F}'}$ definēta kārtība. Ar pieņēmumu, ka imitācijas modeļa izejas vērtība ir reāls skaitlis, augstāk piedāvāto definīciju var pārveidot šādā vienkāršākā definīcijā:

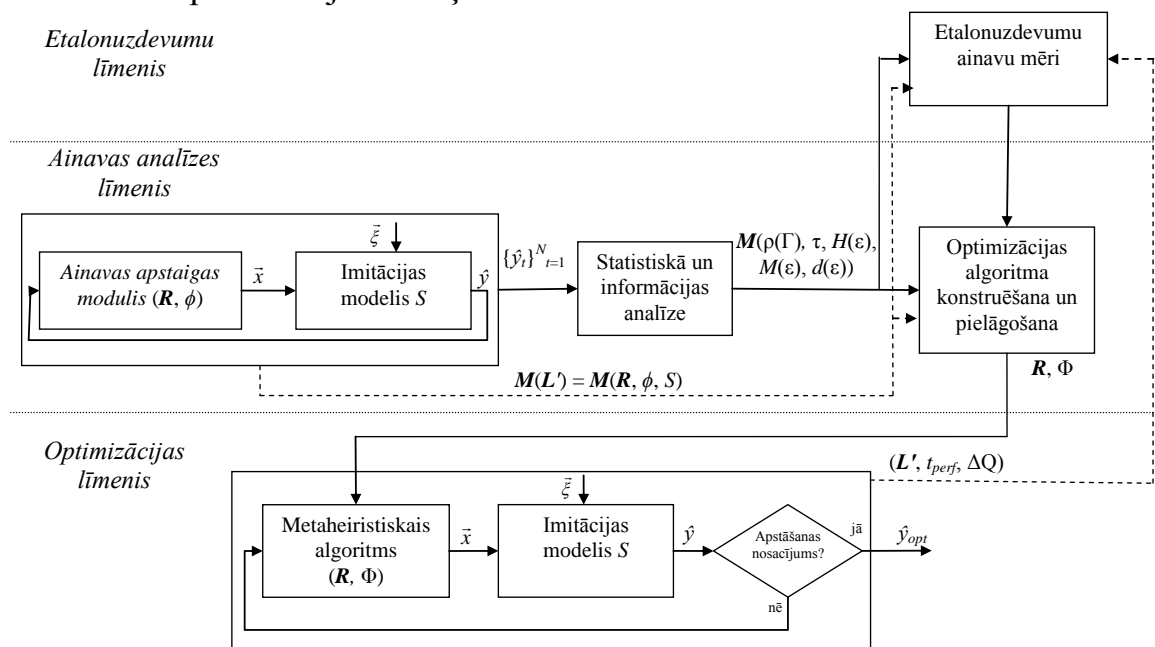
$$L' = (\mathbf{R}, \phi, S), \quad (20)$$

kur \mathbf{R} ir reprezentāciju telpa, ϕ ir meklēšanas operators, kas nosaka ainavu un S ir imitācijas modelis ar vienu izejas vērtību:

$$S = f: \bar{x} \times \xi \mapsto \hat{y}, \quad (21)$$

kur $\bar{x} \in \mathbf{R}$ ir imitācijas modeļa ieejas vērtību (lēmumu mainīgo) vektors, kas attēlo risinājumu reprezentācijas telpā, f – mērķa funkcija, ξ ir imitācijas modeļa stohastiskā komponente un $\hat{y} \in \mathbb{R}$ imitācijas modeļa izejas vērtības matemātiskā cerība.

Lai pielietotu piemērotības ainavas analīzi imitācijas modelēšanā sakņotā optimizācijā, darbā ir piedāvāta sekojoša trīs līmeņu formalizēta shēma (sk. 2. att.). Shēma satur etalonuzdevumu, piemērotības ainavas analīzes un optimizācijas līmeņus.



2. att. Imitācijas modelēšanā sakņota optimizācija ar piemērotības ainavas analīzi

Etalonuzdevumu līmenī ir apkopota informācija par dažādu ainavu struktūrām, mēriem un optimizācijas algoritma veiktspēju šīm ainavām. *Ainavu analīzes līmenis* ir galvenais procedūras līmenis. To veido pati

piemērotības ainavas analīzes procedūra. Pirmkārt, pielietojot dažādas apstaigas stratēģijas, ir veikta ainavas apstaiga. Apstaigā ir iegūta piemērotības vērtību laika rinda un ar ainavas statistiskās un informācijas analīzes pieejām ir aprēķināti ainavas mēri. Iegūtā datu kopa kopā ar informāciju par pielietoto apstaigas stratēģiju ir izmantota, lai izvēlētos un pielāgotu atbilstošu optimizācijas algoritmu. Papildus tam, savāktie dati ir pievienoti arī etalonuzdevumu datu kopai, kuru izmantos nākotnē. *Optimizācijas līmenī* izvēlētais metaheiristiskais algoritms ir izmantots pētāmās sistēmas optimizācijai, pielietojot imitācijas modelēšanā sakņotu metaheiristisku optimizāciju. Informācija par izmantotā algoritma veikspēju ir savākta un pievienota pie etalonuzdevumu datu kopas.

Ainavas apstaigas moduli LW var interpretēt kā:

$$\vec{x}_{t+1} = LW(\vec{x}_t, \phi, \hat{y}), \quad (22)$$

kur t ir paveikto ainavas apstaigas iterāciju skaits no N gaidāmajiem, \vec{x}_{t+1} ir imitācijas modeļa ieejas vērtību vektors nākamajai iterācijai, \vec{x}_t ir imitācijas modeļa ieejas vērtību vektors iepriekšējā iterācijā, ϕ ir operators un \hat{y} ir imitācijas modeļa izeja.

Ainavas apstaigā ir izmantots operators ϕ , ņemot vērā informāciju par reprezentācijas telpu. Adaptīvas un neitrālas apstaigas virziena noteikšanai ir izmantota imitācijas modeļa izeja \hat{y} . Ainavas apstaigas moduļa izeja ir tāds imitācijas modeļa ieejas vērtību vektors $\vec{x} = (x_1, x_2, \dots, x_k)$, kur $\vec{x} \in \mathbf{R}$, un \mathbf{R} ir reprezentāciju telpa, savukārt, k ir imitācijas modeļa ieejas vērtību skaits. Imitācijas modeļa vērtībām ir jābūt reāliem skaitļiem $x_i \in \mathbb{R} \quad \forall 0 \leq i < k$.

Imitācijas modelis S ir izmantots optimizējamās sistēmas darbības kvalitātes novērtēšanai. Tas atgriež vairāku replikāciju izeju gaidāmo vērtību un tiek definēts šādā veidā:

$$S = f: \vec{x} \times \vec{\xi} \mapsto \hat{y}, \quad (23)$$

kur f ir mērķa funkcija, kuras optimālā vērtība tiek meklēta optimizācijā;

$\vec{x} = (x_1, x_2, \dots, x_k)$ – modeļa k ieejas vērtību vektors;

$\vec{\xi} = (\xi_1, \xi_2, \dots, \xi_d)$ – ārējās vides d vērtību traucējumu vektors;

\hat{y} – imitācijas modeļa izejas matemātiskā cerība.

Darbā tiek pieņemts, ka *imitācijas modelis* S ir vairākas imitācijas modeļa replikācijas, kas dod vidējo izejas vērtību $\hat{y} = E[y]$, kur $y \in \mathbb{R}$ ir reāla imitācijas modeļa izeja un $E[\]$ ir matemātiskā cerība. Promocijas darbā ir pieņemts, ka imitācijas modeļa iekšējās konstantes un stāvokļi nav analizēti, bet imitācijas modelis tiek pieņemts kā „melnā kaste”.

Ainavas apstaigas apvienotais process, kas ietver sevī LW un S , nodrošina vairāku laika rindu $\{\hat{y}_t\}_{t=1}^N$ iegūšanu, atbilstoši apstaigātajām trajektorijām. Šīs laika rindas ir ekvivalentas laika $\{f_t\}_{t=1}^N$ rindām no ainavas analīzes pieeju apraksta, un ir N modeļa izeju (piemērotības vērtību) vektors katrai ainavas trajektorijai.

Statistiskā un informācijas analīze ir modulis, kas izpilda piemērotības vērtību secības $\{f_t\}_{t=1}^N$ analīzi. Modulī ir izpildīta ainavu statistiskā un informācijas analīzes mēru skaitļošana. Ir iegūta $\rho(\Gamma)$, τ , $H(\varepsilon)$, $M(\varepsilon)$, $h(\varepsilon)$ vērtību kopa dažādiem attālumiem Γ starp risinājumiem un dažādām jutīguma vērtībām ε . Analīzes modulis *S&IA* var būt interpretēts sekojošā veidā:

$$S\&IA : \mathbb{R}^N \times \mathbb{N} \times \mathbb{R}_+ \rightarrow \mathbb{R}^5 \\ (\{f_t\}_{t=1}^N) \times \Gamma \times \varepsilon \mapsto (\rho, \tau, H, M, h). \quad (24)$$

Optimizācijas algoritma konstruēšanas un pielāgošanas modulis ļauj izvēlēties atbilstošu optimizācijas algoritmu un pielāgot tā parametrus sarežģītās sistēmas, kas ir attēlota ar S , optimizācijai. Algoritma, to komponentu un parametru izvēle ir bāzēta uz imitācijā sakņotas piemērotības ainavas analīzi un etalonuzdevumu ainavu datiem. Moduļa ieejas dati ir vērtību kopa, kas apraksta analizējamo piemērotības ainavu. Izejas vērtības ir izvēlētais metaheiristiskais algoritms un tā konfigurācija. Konfigurācija ietver sevī risinājumu reprezentāciju, kas ir definēta ar reprezentāciju pārmeklēšanas telpu R un metaheiristiskā algoritma operatoru kopu Φ , piemēram, mutācijas un krustošanas operatorus, ja ir izvēlēts ģenētiskais algoritms.

Metaheiristiskā algoritma un tā konfigurācijas izvēle ir bāzēta uz likumiem un rekomendācijām, kas ir lietotas zināmām piemērotības ainavas mēru vērtībām. Alternatīvais algoritma izvēles un konfigurācijas veids ir balstīts uz etalonuzdevumu datiem, kuros labākie optimizācijas algoritmi tiek asociēti ar atbilstošiem optimizācijas scenārijiem.

Optimizācijas līmenī imitācijas modelis ir definēts ar formulu (23). **Metaheiristiskais algoritms** šeit lieto reprezentācijas metodi (kas ir definēta ar R), un piedāvātos operatorus Φ , kas ir determinēti ar konstruēšanas un pielāgošanas moduli. Metaheiristiskais algoritms *MA* šeit var būt formalizēts sekojošā veidā:

$$MA : (R, \Phi, \hat{y}, M(\langle \bar{x}, \hat{y} \rangle)_t) \mapsto \bar{x}_{t+1}, \quad (25)$$

kur t – ir imitācijas modeļa izpildīto novērtējumu skaits;
 \bar{x}_{t+1} – ir piedāvāto imitācijas modeļa lēmumu mainīgu vērtību vektors;

$M(\langle \bar{x}, \hat{y} \rangle)_t$ – ir iepriekš izanalizēto risinājumu kandidātu atmiņa, kas ir iegūta pēc t novērtējumiem;

\hat{y} – imitācijas modeļa izejas vērtības matemātiskā cerība.

Algoritma apstāšanās nosacījums nosaka, vai ir iegūts pieņemamais risinājums un attiecīgi optimizācijas cikls var būt pārtraukts. Kad optimizācijas cikls apstājas, kā optimālais risinājums \hat{y}_{opt} ir izdots labākais atrastais risinājums $\langle \bar{x}, \hat{y} \rangle$.

Optimizācijas modeļa veikspēja var būt pievienota pie etalonuzdevumu mērījumu datu kopas. Informācija par optimizācijas veikspēju ir attēlota trīs

vērtību korteža formā. Šīs vērtības ir risināmas optimizācijas problēmas: piemērotības ainava (20), optimizācijas algoritma patērētā laika mērījums t_{perf} , kas attēlo laiku patērētu labākā risinājuma ieguvei, un ΔQ – algoritma veiktspējas mērs, kas ir determinēts ar labākā atrastā risinājuma kvalitātes uzlabošanas starpību.

Šeit piemērotības ainavas analīze ir paredzēta imitācijas modeļa piemērotības ainavas analīzei un priekšstata iegūšanai, par to, kādā veidā ir jāpielāgo un jānokonfigurē optimizācijas modulis.

2. Etalonuzdevumos bāzēta piemērotības ainavas analīze

Sadaļā ir piedāvāta praktisku piemērotības ainavu analīzes tehniku realizācija, lai noteiktu, kā iegūt mērus etalonuzdevumu ainavās ar zināmu struktūru. Sadaļā arī ir analizēts, kādā veidā troksnis piemērotības funkcijā ietekmē izvēlētos piemērotības ainavas mērus. Lai noteiktu piemērotības ainavas struktūru ietekmi uz ainavas mēru vērtību un optimizācijas algoritma atbilstošu uzvedību, tika veikta etalonuzdevumu ainavu detalizēta izpēte.

Etalonuzdevumu piemērotības ainavas

Statistiku un informācijas mēru analīzei ir pielietotas sekojošas četras etalonuzdevumu funkcijas, kas ir plaši aprakstītas literatūrā: Sfēras funkcija [9]; Rastrigina funkcija [53]; Rosenbrock funkcija [9] un Ackley funkcija [1]. Šīs funkcijas var būt pārmeklētas vienādā pārmeklēšanas telpā un ir viegli interpretējamās pie diviem parametriem.

Sfēras funkcija ir nepārtraukta, izliekta, kvadrātiska un vienmodāla funkcija. Parametru vektoram $X = \langle x_1, \dots, x_n \rangle$ ar garumu n , tā ir definēta sekojoši:

$$f_{Sphere}(\langle x_1, \dots, x_n \rangle) = \sum_{i=1}^n x_i^2. \quad (26)$$

Rastrigina funkcijai (27) ir negluda ainava, un tā ir definēta kā:

$$f_{Rastrigin}(\langle x_1, \dots, x_n \rangle) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i)). \quad (27)$$

Rosenbrock funkcija (28) ir nepārtraukta, neizliekta, ceturtās pakāpes un vienmodāla funkcija:

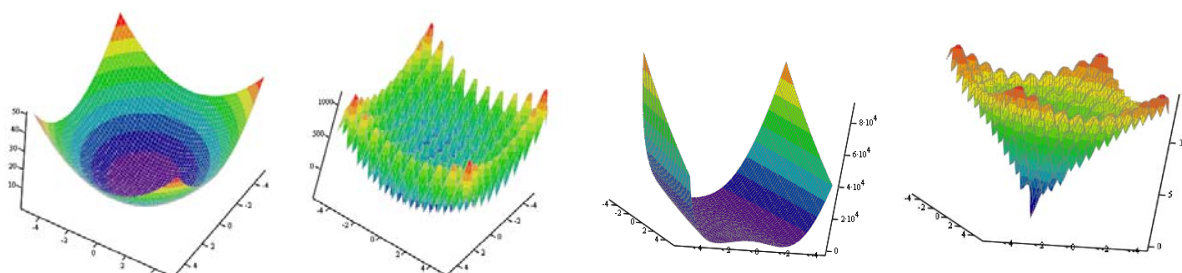
$$f_{Rosenbrock}(\langle x_1, \dots, x_n \rangle) = \sum_{i=1}^{n-1} \left((1 - x_i^2) + 100(x_{i+1} - x_i^2)^2 \right) \quad (28)$$

Ackley funkcijā (29) līdzīgi kā funkcijai (27) ir vairāki lokālie optimumi, tā ir multimodāla, negluda, un tā ir definēta ar:

$$f(\langle x_1, \dots, x_n \rangle) = 20 + e - 20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right). \quad (29)$$

Sfēras, Rastrigina un Ackley funkcijām globālais optimums ir punkts ar koordinātēm $x_i = 0$, $i = 1 \dots n$. Rosenbrock funkcijas globālais optimums

atrodas punktā ar koordinātēm $x_i = 1, i = 1 \dots n$, un atrodas paraboliskas formas plakanaajā ielejā. Visu četru funkciju grafiskie attēlojumi 2-dimensiju problēmām ir parādīti 3. attēlā.



Sfēras funkcija

Rastrigina funkcija

Rosenbrock funkcija

Ackley funkcija

3. att. Etalonuzdevumu funkciju vizualizācija

Promocijas darbā funkciju analīzei ir izvēlēti sekojoši pārmeklēšanas telpas parametri: funkcijas parametru skaits $n = 2$, un pārmeklēšanas robežas nodefinētas kā $-5 \leq x_i \leq 5, i = 1, 2$. Ir pielietoti divi reprezentācijas veidi: reālu skaitļu kodēšana un binārā kodēšana. Pirmajā gadījumā vērtības x_1 un x_2 tiek kodētas kā reālu skaitļu vektors ar skaitļu precizitāti 0.01, piemēram, -3.49 un 0.84 . Bināri kodēto hromosomu garums bija izvēlēts 20, un pirmie 10 biti kodēja x_1 , bet pārējie kodēja x_2 .

Mutācijas operators ir atkarīgs no reprezentācijas veida. Reālo skaitļu reprezentācijā mutācijas operators izmaina katru hromosomas vērtību ar $+0.01$ vai -0.01 un varbūtību $1/3$. Ja visas vērtības x_i netika izmainītas, tad mutācijas operators tiek pielietots atkārtoti, līdz tiks sasniegta vērtību izmaiņa. Bināri reprezentētiem risinājumiem tika pielietots bita apgriešanas operators.

Šādi sadaļā tika analizētas 8 piemērotības ainavas četrām funkcijām - katrai ar divām dažādām reprezentācijām (un attiecīgo mutācijas operatoru).

Etalonuzdevumu eksperimentālā analīze

Darbā tika izpildīta trīs etalonuzdevumu ainavu ātra analīze, lai attēlotu piemērotības ainavas analīzes procedūras darbību. Detalizētākai ainavas analīzei tika izstrādāts un lietots lietojumprogrammas prototips statistisko un informācijas piemērotības ainavas mēru izskaitļošanai. Lai novērtētu piemērotības ainavu strukturālos mērus, tika veikti vairāki eksperimenti, kuros ir analizētas vairākas ģenerētās ainavas trajektorijas ar sākuma punktiem, kas ir vienmērīgi sadalīti reprezentācijas telpā.

Autokorelācijas funkcijas vērtības ir gandrīz vienādas starp dažādām etalonuzdevumu funkcijām. Palielinoties trajektorijas garumam, tie tiecas uz vērtību 1. Kaut gan korelācijas mēri ir atkarīgi no analizētās trajektorijas garuma, informācijas mēriem tāds efekts nav redzams.

Otrajā eksperimentu sērijā bija salīdzināti statistiskie ainavas mēri ainavām, kas uzdotas ar atšķirīgu reprezentāciju. Analīze rāda, ka ainavām ar

reālu skaitļu kodēšanu korelācijas vērtības starp blakus risinājumiem ir augstākas, un meklēšanas procesam ir jābūt ātrākam. Tomēr vienā reprezentācijā korelācijas vērtības starp dažādām funkcijām ir gandrīz vienādas.

Trešajā eksperimentu sērijā tika salīdzināti informācijas mēri starp visām apskatītajām ainavām, lietojot dažādas koeficienta ε vērtības. Analīzes rezultāti rāda lielu informācijas satura vērtību, pie mazām ε vērtībām. Augstāka vērtība Sfēras funkcijai ar reālu skaitļu kodēšanu ir sasniegta kad ε ir tuvs 0.03. Pie $\varepsilon = 0$, kad analizējot tiek ņemtas vērā vismazākās izmaiņas piemērotībā, informācijas mēri dažādām ainavām kļūst gandrīz vienādi un nesniedz jaunu informāciju par piemērotības ainavām. Pie tādas jutīguma vērtības informācijas saturs analizētajām ainavām ir tuvs 0.388, daļējas informācijas saturs 0.5 un blīvuma baseina informācija tuva 0.63. Tajā pat laikā, mazākas informācijas satura vērtības Rosenbrock funkcijai, salīdzinot ar Sfēras funkciju, nozīmē ainavas lielāku plakanumu.

Ņemot vērā to, ka piemērotības funkcija dažādām etalonuzdevumu funkcijām mainās pēc izmēra atšķirīgos intervālos, gandrīz nav iespējams salīdzināt šos mērus pie $\varepsilon > 0$. Lai to pārvarētu, katrai ainavai tika izvēlēta tāda ε_1 vērtība, kura apzīmē vienādu intervālu proporcionāli starpībai starp mazāko un lielāko piemērotības vērtību apskatītajā trajektorijā. Sfēras, Rastrigina un Ackley funkcijām tā tika izvēlēta $\varepsilon_1 = 0.04$, bet Rosenbrock funkcijai: $\varepsilon_1 = 50$.

Piemērotības funkcijas troksnis un ainavu analīze

Lai noteiktu to, kā ainavas apstaigas vietas izvēle ietekmēs ainavas analīzes rezultātus, Sfēras funkcijai tika analizēti trīs trajektorijas uzsākšanas punkti: blakus globālajam optimumam, blakus lokāliem maksimumiem, un pa vidu starp tiem. Tādā veidā tika izvēlēti sekojoši atbilstoši ainavas apstaigas uzsākšanas punkti: $\langle 0.05; -0.01 \rangle$, $\langle 4.49; 4.82 \rangle$ un $\langle -3.71; 1.23 \rangle$. Sfēras funkcijas analīzes rezultāti nerada ievērojamu ainavas analīzes reģiona izvēles ietekmi uz autokorelācijas funkciju, bet piemērotības vērtību starpība blakus risinājumiem ievērojami ietekmē informācijas mērus. Tā var secināt, ka ε parametra vērtībai ir jābūt izvēlētai uzmanīgi, ņemot vērā iespējamās izmaiņas analizējamajā reģionā.

Lai noteiktu gadījuma trokšņa ietekmi uz ainavas analīzes rezultātiem, tika izpildīti papildus eksperimenti. Šajos eksperimentos piemērotības funkcija ar troksni f^* tika definēta sekojošā veidā:

$$f^* = f + \xi, \quad (30)$$

kur f ir etalonuzdevuma piemērotības funkcija un ξ ir gadījuma troksnis, kas ir definēts kā statistiskā kļūda ar normālu sadalījumu, vidējo vienādu ar 0 un standartnovirzi σ^2 .

Veikto eksperimentu rezultāti rāda, ka gan informācijas gan statistiskā analīze ir ļoti jutīgas pie trokšņa. Palielinoties trokšņa standartnovirzei,

autokorelācijas vērtības kļūst mazākas pie maziem attālumiem starp risinājumiem un lielākas – pie lieliem attālumiem. Līdzīgā veidā troksnis palielina informācijas satura vērtības, palielinot struktūru entropiju ainavā.

Etalonuzdevumu funkciju optimizācijas eksperimenti

Lai noteiktu korelāciju starp piemērotības ainavas analīzes rezultātiem un reālas problēmas sarežģītību evolūcijas algoritmiem, tika izpildīta etalonuzdevumu optimizācijas eksperimentu sērija. Optimizācijā tika lietots ģenētiskais algoritms ar viena punkta krustošanu un augstāk apskatītie mutācijas operatori. Eksperimentos tika noteikta *kopējā panākumu varbūtība* (KPV) [23], kas nosaka, kāda ir varbūtība pie noteiktas paaudzes iegūt globāli optimālu risinājumu. Optimizācijas eksperimentu rezultāti rāda, ka Sfēras, Rastrigina un Rosenbrock funkcijām, kas ir kodētas binārajā sistēmā, globālais optimums tiek atrasts pēc 20. paaudzes 50-60% algoritma izpildēs, un KPV nekļūst lielāks pie lielāka paaudžu skaita. Rosenbrock funkcijai globālais optimums tiek atrasts tikai 10-20% gadījumos.

Galvenokārt, izņemot Rosenbrock funkciju, ĢA atrod risinājumus problēmām ar piemērotības ainavām reālo skaitļu reprezentācijā, nekā binārajā reprezentācijā, kas arī bija iepriekš noteikts statistiskajā analīzē. Tā kā autokorelācija starp kaimiņu risinājumu piemērotības vērtībām ir ļoti augsta, ģenētiskajam algoritmam ir vieglāk pārvietoties pie risinājuma ar lielāku piemērotību. Tomēr, neskatoties uz gandrīz vienādajiem ainavas analīzes mēriem, Rosenbrock funkcija bija sarežģītāka optimizācijai ar izvēlēto ĢA. Lai noteiktu šī fakta iemeslus, tika analizēta dažādu paaudžu populācijas dinamika, kur tika noskaidrots, ka ģenētiskajā algoritmā ar reālu skaitļu kodēšanu populācija tiecas uz funkcijas ainavas ielejas vidu, nevis uz globālo optimumu, kas atrodas ārpus šīs vietas.

Piemērotības ainavu klasteru analīzē

Literatūrā [38] piemērotības ainavas analīze tika lietota, lai noteiktu, kādā veidā vienas klases problēmas ir līdzīgas, un kādā veidā tās savstarpēji atšķiras. To noteikšanai visas problēmas piemērotības analīzes mēru vērtības, iegūtas lietojot dažādus operatorus, ir apvienotas vienā problēmas raksturīpašību vektorā. Izmantojot raksturīpašību vektoru, ir iespējams veikt problēmu klasteru analīzi un noteikt kādi problēmu gadījumi ir līdzīgi, bet kādi ir atšķirīgi.

Etalonuzdevumu ainavu detalizēta analīze

Lai iegūtu vēl vairāk informācijas par etalonuzdevumu ainavas struktūrām, tika veikta ainavu grafu analīze. Sfēras funkcijai tika konstruētas četru ainavas grafu apkaimes vienam nejauši izvēlētam risinājumam: diviem augstāk aprakstītiem mutācijas operatoriem un diviem viena punkta krustošanas operatoriem (binārai un reālu skaitļu reprezentācijai). Mutācijas operatoriem tika noteikti kaimiņu risinājumi, un tika noteikta šo risinājumu

piemērotība. Krustošanas operatoram tika izvēlēta tāda ainavas grafu reprezentācija, kur grafa virsotne sastāv no risinājuma pāra, un tika noteikti citi risinājuma pāri, kurus ir iespējams sasniegt no apskatāmā risinājuma ar izvēlēto operatoru.

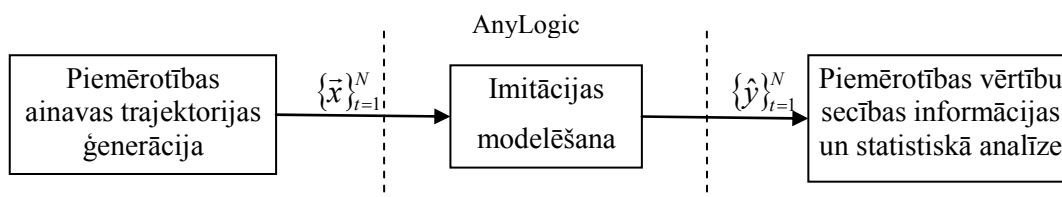
Šīs analīzes rezultātā tika secināts, ka reālu skaitļu mutācijas operators nodrošina salīdzinoši lēnu, bet tomēr virzību uz globālo optimumu. Izvēlētais krustošanas operators ir pilnīgi neefektīvs, jo no jebkuriem diviem indivīdiem var iegūt tikai vienu iespējamo indivīdu pāri, un izvēlēta ĢA reālo skaitļu reprezentācija optimizējot uzdevumu izmantoja galvenokārt tikai mutāciju. Binārajā reprezentācijā mutācijas operatora ainavā indivīdam ir lielāks kaimiņu risinājumu skaits, un šis operators ļauj tuvoties globālajam optimumam ar mazāku operatora iterāciju skaitu, taču arī ar mazāku varbūtību, nekā reālo skaitļu operatoram. Krustošanas operators binārajā kodēšanā ir efektīvāks, nekā izvēlētais reālo skaitļu operators. Kaut gan krustošanas operatora piemērotības ainavas grafs sastāv no vairākām nesavienotām komponentēm, tomēr šīs komponentes satur vairākus risinājumus, un, pat tikai pielietojot krustošanas operatoru bez mutācijas, var tuvoties funkcijas optimumam.

3. Imitācijas modelēšanā sakņota piemērotības ainavas analīze un optimizācija

Piemērotības ainavas analīzes instrumenta prototipa izstrāde

Potenciālo risinājumu piemērotības novērtēšana ir veikta, izmantojot imitācijas modelēšanu. Prototipā realizētā imitācijas modelēšanā sakņotā piemērotības ainavas analīzes procedūra sastāv no trim soļiem (sk. 4. att.):

1. Trajektorijas ģenerēšana piemērotības ainavā.
2. Trajektorijas risinājumu piemērotības novērtēšana.
3. Piemērotības vērtību virknes statistiskā un informācijas analīze.



4. att. Imitācijas modelēšanā sakņotas piemērotības ainavas analīzes procedūra

Darbā ir izstrādātas divas Java lietojumprogrammas trajektorijas ģenerēšanai piemērotības ainavā (gadījuma ainavas apstaigai) un iegūtas piemērotības vērtību virknes analīzei. Piemērotības novērtēšanai ir izmantots imitācijas modelis, kas izveidots AnyLogic 6 modelēšanas programmatūrā. Sākotnēji lietojumprogramma izpilda gadījuma ainavas apstaigu analizētās

problēmas piemērotības ainavā. Apstaigas rezultātā ir iegūta virkne $\{\bar{x}_t\}_N^{t=1}$ no trajektorijas potenciālo risinājumu parametru vērtībām.

Otrajā solī imitācijas modelis, lietojot AnyLogic parametru variācijas eksperimentu tipu tiek palaists ar dažādām lēmumu mainīgo vērtībām, kas ir noteiktas ar risinājumu parametriem saņemtajā risinājumu virknē. Rezultātā, imitācijas modelis izveido piemērotības vērtību masīvu $\{\hat{y}_t\}_N^{t=1}$.

Pēdējā solī iegūtā piemērotības vērtību virkne ir analizēta lietojumprogrammā, un ir aprēķināti analizētās piemērotības ainavas statistiskas un informācijas mēri. Statistiskajā analizē ir aprēķināti korelācijas attālums un autokorelācijas funkcija pie dažādiem attālumiem starp risinājumiem. Informācijas analīzes algoritms, savukārt, sastāv no sekojošiem soļiem:

1. Informācijas stabilitātes ε^* noteikšana.
2. Informācijas satura $H(\varepsilon)$, daļēja informācijas satura $M(\varepsilon)$ un blīvuma-baseinu informācijas $h(\varepsilon)$ iteratīva aprēķināšana pie dažādām ε vērtībām.
3. Aprēķināto datu izvade.

Lai noteiktu ε^* vērtību, iespējamo ε parametra vērtību intervāls ir iteratīvi sadalīts uz pusēm. Katrā iterācijā ir izvēlēts tas apakšintervāls, kas satur ε^* vērtību (tas ir $H(\varepsilon)$, kas pie šīs vērtības kļūst vienāds ar 0). Informācijas mēri $H(\varepsilon)$, $M(\varepsilon)$ un $h(\varepsilon)$ ir aprēķināti iteratīvi intervālā $[0, \varepsilon^*)$ ar soli vienādu ar $0.05 \cdot \varepsilon^*$.

Analīzē ir savākti statistiskie un informācijas mēri par vairākām ainavas gadījuma apstaigām. Līdz ar to, ka gadījuma apstaigas ir uzsāktas dažādās ainavas vietās, ir iegūti ainavas mēri par lielāko ainavas daļu.

Gadījuma izpēte

Darbā analizētā gadījuma izpēte, kas ir izmantota prototipa validācijai, ir bāzēta transportlīdzekļu kustības grafika optimizācijas problēmā ar laika logiem. Uzņēmuma transportlīdzekļiem ir jāpiegādā dažāda tipa preces no viena sadales centra uz vairākiem veikaliem (klientiem). Transportlīdzekļu iespējamie maršruti ir zināmi un tie ir iekļauti reisos. Katram reisam ir noteikti: veikalu secība (maršruts), vidējie laika intervāli transportlīdzekļa pārvietošanai starp maršruta punktiem, iekraušanas un izkraušanas vidējie laiki un preču tipi, kas ir piegādāti atbilstošā reisā. Veikali pieņem preces tikai noteiktos laika logos. Veikaliem ir noteikts vidējais preču dienas pieprasījums. Transportlīdzekļu ietilpība ir ierobežota. Problēmas būtība ir reisu sadalīšana starp transportlīdzekļiem tādā veidā, lai tiktu samazināti transportlīdzekļu dīkstāves laiki. Transportlīdzekļa dīkstāves laiks ir definēts kā visu laika periodu, kad transportlīdzeklis gaida nākamā reisa uzsākšanu sadales centrā, ilgumu summa.

Transportlīdzekļu kustības grafika optimizācijas problēma ir bieži apskatīta literatūrā [11, 35], bet praksē šīs problēmas nostādne bieži tiek

modificēta ar vairākiem papildus ierobežojumiem. Literatūrā problēmas risināšanai ir pielietotas veselo skaitļu programmēšana un heuristiskas metodes. Taču šīs metodes bieži tiek specializētas atsevišķām problēmas apakšklasēm un nav pielietojamas praksē sastopamo problēmu risināšanai [11]. Līdz ar to, ka praksē pārvietošanas laiks starp maršruta punktiem ir stohastiska procesa gadījuma vērtība, problēmas risināšanai darbā ir izvēlēta imitācijas modelēšanā sakņota optimizācija.

Problēmas risināšanas lēmuma mainīgie ir divas kopas, kas nosaka katram reisam divas vērtības:

v_i – transportlīdzeklis, kas izpildīs i -to reisu;

t_i – i -tā reisa uzsākšanas laiks.

Apskatītās problēmas ierobežojumi ir sadalīti divās grupās:

1. Transportlīdzekļu ietilpības ierobežojumi.

2. Preču piegādes laika logi veikaliem.

Problēmai ir noteikta mērķa funkcija f , kas minimizē transportlīdzekļu dīkstāves laiku. Problēmas empīriskā analīze rāda, ka vairums no potenciālajiem risinājumiem neatbilst definētajiem ierobežojumiem. Optimizācijas efektivitātes uzlabošanai funkcija (31) tika modificēta ar vairākiem suda parametriem:

$$f^* = \sum_{i=1}^N T_{idle}^i + k_1 T_c + k_2 T_m + k_3 T_o + k_4 N_{ol} + k_5 N_{ot} \rightarrow \min, \quad (31)$$

kur f ir modificētā mērķa funkcija;

T_{idle}^i – i -tā transportlīdzekļa kopējais dīkstāves laiks;

N – transportlīdzekļu skaits;

T_c – viena transportlīdzekļa pārklājošo reisu sakrītošo intervālu summārais laiks;

T_m – summārais laiks, kad piegādes reisā nesakrīt ar laika logiem;

T_o – summārais laiks, kad transportlīdzeklis tika izmantots ilgāk par 24 stundām;

N_{ol} – transportlīdzekļu, kas tika izmantoti ilgāk par 24 stundām, skaits;

N_{ot} – pārslogoto transportlīdzekļu skaits.

Mērķa funkcijā (31) visas ierobežojumu neizpildītās vērtības tiek reizinātas ar koeficientiem $k_i \gg 1$, $i=1, \dots, 5$, kas ļauj ievērojami pasliktināt neatļauto risinājumu piemērotības vērtību.

Imitācijas modeļa izstrāde

Lai novērtētu potenciālā kustības grafika risinājuma piemērotību, tika izstrādāts diskrētu notikumu imitācijas modelis AnyLogic modelēšanas vidē. Modelī katrs transportlīdzeklis ir modelēts kā atsevišķs aktīvs objekts. Imitācijas modeļi ir noteikti problēmas ierobežojumu neizpildītie gadījumi un tiek novērtēts to apjoms. Problēmas imitācijas modeļa izstrādei tika izvēlēts AnyLogic modelēšanas rīks, kas ir bāzēts objektorientētā koncepcijā un atbalsta vairāku imitācijas modelēšanas metodoloģiju kombinēšanu.

Imitācijas modeļa galvenais uzdevums ir novērtēt transportlīdzekļu kustības grafika efektivitāti, novērtējot kopējo dīkstāves laiku, ņemot vērā stohastiskos nosacījumus. Ar lēmuma mainīgajiem katram reisam ir noteikts tā izpildošs transportlīdzeklis un reisa uzsākšanas laiks. Imitācijas modelis papildus ģenerē Ganta diagrammu, kas attēlo transportlīdzekļu kustības grafiku, lai to varētu novērtēt plānotājs.

Imitācijas modelī ir definēti divi aktīvi objekti (apakšmodeļi): galvenais objekts un transportlīdzekļa objekts. Galvenais objekts ietver sevī: lēmumu mainīgos, imitācijas modeļa ieejas datu inicializāciju funkciju, reisu un veikalu datu mainīgo kolekcijas, kopējā dīkstāves un kopējā lietošanas laika mainīgos. Transportlīdzekļa aktīvais objekts imitē transportlīdzekļa procesus, izmantojot stāvokļu diagrammu. Tā definē iespējamus transportlīdzekļa stāvokļus un pārejas no viena stāvokļa uz nākamo. Transportlīdzekļa objekts satur uzdevumu sarakstu (kā mainīgo kolekciju), kā arī mainīgus transportlīdzekļa dīkstāves laika aprēķināšanai un problēmas ierobežojumu pārkāpšanas skaita uzkrāšanai. Transportlīdzekļu piešķiršana reisiem imitācijas modelī ir pārveidota uz uzdevumu piešķiršanu transportlīdzeklī, kur uzdevums ir reisa un tā uzsākšanas laika apvienojums. Modeļa inicializācijas funkcija pārveido modeļa lēmumu mainīgo vērtības uz katra transportlīdzekļa ieplānoto uzdevumu sarakstu.

Imitācijas modeļa procesu attēlošanai katram transportlīdzeklī ar dažādu krāsu uz laika grafika ir parādīti stāvokļi, kuros tas ir bijis. Modeļa animācijā visu transportlīdzekļu laika grafiki ir apvienoti vienā Ganta diagrammā, un tiem ir pievienots sadales centra iekraušanas vārtu izmantošanas grafiks. Informācija par transportlīdzekļu dīkstāvi un izpildītajiem uzdevumiem ir parādīta blakus atbilstošā transportlīdzekļa izmantošanas laika grafikam.

Modeļa validācijai un tā atbilstības noteikšanai darbā bija modelēts uzņēmumā esošais transportlīdzekļu kustības grafiks, lietojot izstrādāto imitācijas modeli. Iegūtais kustības grafiks un tā parametri atbilst uzņēmuma Ganta diagrammai, kas ir iegūta praksē.

Lai noteiktu kādā veidā imitācijas modeļa gadījuma faktori ietekmē modeļa izejas vērtības, tika veikta stohastisko parametru ietekmes analīze. Imitācijas modelī transportlīdzekļu pārvietošanas laiki starp maršruta punktiem tika definēti kā gadījuma vērtības ar normālu sadalījumu. Normālā sadalījuma vidējai vērtībai tika piešķirts transportlīdzekļa pārvietošanas vidējais laiks, bet standartnovirze tika definēta proporcionāli vidējam braukšanas laikam un dažādos eksperimentos tika ņemta ar dažādu proporcionalitātes koeficientu. Eksperimentāli tika noteikts, ka transportlīdzekļu pārvietošanas laika stohastiskums jūtami ietekmē transportlīdzekļu dīkstāves laiku, kas palielinās pie pārvietošanas laika izkliedes palielināšanas. Tajā laikā tiešās izmaksas, kas ir aprēķinātas ar imitācijas modeli un kas ir atkarīgas no transportlīdzekļa laika ceļā, ir

vienādas ticamības intervālā neatkarīgi no transportlīdzekļu pārvietošanas intervāla izkliedes amplitūdas.

Optimizācijas scenāriji

Lai atrisinātu transportlīdzekļu kustības grafika optimizācijas problēmu, tika pielietoti trīs optimizācijas scenāriji:

1. Optimizācija OptQuest komerciālā optimizācijas rīkā.
2. Problēmas imitācijas modelēšanā sakņota piemērotības ainavas analīze un optimizācija ar izstrādāto prototipu.
3. Problēmas piemērotības ainavas analīze un optimizācija HeuristicLab struktūrietvarā.

Tā kā transportlīdzekļu kustības grafika imitācijas modelis ir izstrādāts AnyLogic imitācijas modelēšanas vidē, vispirms problēmas optimizācijai tika pielietots izvēlētajā programmatūrā iekļautais komerciālais optimizācijas rīks OptQuest. Optimizācijas rīkā pielietotās metodes nav atklātas, bet ir informācija, ka to pamatmetodes ir izklīdināta meklēšana un tabu meklēšana. Ar OptQuest rīku neizdevās iegūt transportlīdzekļu kustības grafika optimizācijas problēmas kvalitatīvus risinājumus, kas atbilstu visu ierobežojumu izpildei, pat samazinot lēmumu mainīgo skaitu. Taču ar izvēlēto rīku izdevās uzlabot esošo risinājumu, kas sniedz informāciju par problēmas turpmākas optimizācijas iespējām.

Problēmas izpēte ar izstrādāto instrumentu

Otrajā scenārijā transportlīdzekļu kustības grafika optimizācijas problēma vispirms tika analizēta ar izstrādāto imitācijas modelēšanā sakņoto piemērotības ainavas analīzes rīku un tad tika optimizēta ar pielāgoto ģenētisko algoritmu.

Transportlīdzekļa kustības grafika optimizācijas problēmas risinājumi tika kodēti ar veselu skaitļu vektora hromosomu, kuras garums ir divreiz lielāks nekā reisu skaits. Nepāra gēni attēlo reisa uzsākšanas laiku, bet pāra gēni transportlīdzekļa numuru, kas izpildīs esošo reisu. Izmantotais mutācijas operators maina vienu nejauši izvēlētu reisu. Reisa izvēles varbūtība visiem reisiem ir vienāda. Izvēlētajam reisam tiek piešķirts jauns transportlīdzeklis, bet uzsākšanas laiks ir nobīdīts uz konstanti.

Tika izanalizētas piemērotības ainavas gan problēmai ar stohastiskiem, gan problēmai ar determinētiem transportlīdzekļu pārvietošanas laikiem. Katrā ainavas analīzes eksperimentu sērijā tika veikti 5 eksperimenti ar 100 risinājumu garu trajektoriju. Gadījuma apstaigā iegūtie mēri ir parādīti 1. tabulā. Informācijas ainavu parametra ε vērtība ir noteikta kā 0.1 no kopējā intervāla starp mazāku un lielāku piemērotību analizētajā trajektorijā.

No informācijas mēru vērtībām var redzēt, ka problēmai ar stohastiskiem parametriem ir lielākā struktūru entropija ainavā un tās ainavai ir jābūt ar lielāku modalitāti, nekā problēmas ar determinētiem datiem

ainavai. Kopumā informācijas saturs $H(0.1)$ ir relatīvi liels, daļējais informācijas saturs $M(0.1)$ ir zems, kas liecina par mazu problēmas ainavas modalitāti. Blīvuma baseina informācijas vērtība $h(0.1)$ rāda, ka smaiļu blīvums ir liels. Korelācijas attālums ir ļoti mazs, un tas nozīmē, ka problēmas piemērotības ainava var būt sarežģīta pārmeklēšanai.

1. tabula

Transportlīdzekļu kustības plānošanas problēmas ainavas informācijas un statistiskie mēri

Pārvietošanas laiks	$H(0.1)$	$M(0.1)$	$h(0.1)$	ε^*	$\rho(1)$	$\rho(10)$	τ
Stohastisks	0.66	0.20	0.49	0.40	0.84	0.21	7.24
Determinēts	0.62	0.17	0.37	0.35	0.89	0.32	8.75

Otrajā eksperimentu sērijā tika noteikta problēmas piemērotības ainavas informācijas mēru atkarība no informācijas analīzes jutīguma parametra ε vērtības. Informācijas saturs ir vislielākais, taču ne pie mazām ε vērtībām, kad ir ņemti vērā vismazākie ainavas slīpumi, bet gan pie lielākām vērtības, kad ainava ir padarīta gludāka. Daļēja informācijas satura vērtība ar ε vērtības palielināšanu tikai samazinājās.

No piemērotības ainavas analīzes rezultātiem var secināt, ka pētītajai problēmai nav jābūt īpaši sarežģītai priekš evolūcijas algoritmiem. Salīdzinot ar iepriekš veikto etalonuzdevumu analīzi, transportlīdzekļu kustības grafika problēmas statistiskie mēri ir līdzīgi Ackley funkcijas mēriem reālu skaitļu kodēšanā. Salīdzinošā analīze rāda, ka problēmas ainava ir gludāka nekā etalonuzdevumu piemērotības ainavas ar bināro reprezentāciju. Šādi analizētās problēmas var risināt ar ĢA ne sliktāk, kā etalonuzdevumu problēmas.

Imitācijas modelēšanā sakņotajai transportlīdzekļu kustības grafika problēmas optimizācijai tika izmantots klasisks ģenētiskais algoritms. Tas ir realizēts kā Java lietojumprogramma un mijiedarbojas ar AnyLogic imitācijas modeli, izmantojot AnyLogic parametru variācijas eksperimentu. Līdz ar to, ka imitācijas modelēšanas eksperimenti ir laiktīlpi, izpildītie imitācijas modeļa novērtējumi tika saglabāti atmiņā, kas ļāva ekonomēt laiku piemērotības novērtēšanai pie vienām un tām pašām lēmumu mainīgo vērtībām.

Veiktajos eksperimentos populācija tika veidota no 200 indivīdiem, bet mutācijas un krustošanas operatori bija pielāgoti pie izvēlētā kodēšanas veida. Eksperimentos tika izmantots viena punkta krustošanas operators veselu skaitļu vektora hromosomai un augstāk aprakstītais mutācijas operators. Atbilstošas krustošanas un mutācijas likmes bija 70% un 1%, bet apstāšanās nosacījums tika definēts kā 150 paaudzes, vai arī gadījumos, kad vairs nebūs redzama labākā atrastā risinājuma uzlabošanās lielā paaudžu skaitā īpaši precīzos eksperimentos.

Pirmā optimizācijas eksperimentu sērijā ir izmantots imitācijas modelis ar determinētiem datiem, un optimizācijas rezultāti rāda, ka ar ĢA ir iespējams atrast risinājumu, kas apmierinās definētos nosacījumus. Pieņemamie rezultāti tika iegūti pie populācijas, ko veido 1000 indivīdi. Savukārt, tālāka populācijas izmēra palielināšana noveda pie meklēšanas laika palielināšanās. tika secināts, ka piemērotības vērtību atmiņa nepalīdz pirmajās paaudzēs, kur indivīdi ir ļoti atšķirīgi, bet būtiski paātrina algoritmu pēdējās paaudzēs, kur risinājumu uzlabošana galvenokārt notiek ar mutācijas operatoru.

Otrajā optimizācijas eksperimentu sērijā tika risināta problēma, izmantojot imitācijas modeli ar stohastiskiem datiem. Iegūtie labākie risinājumi ir līdzīgi modelim ar determinētiem datiem, bet optimizācijas algoritmam ir nepieciešami uzlabojumi, jo daži atrastie lokālie optimālie risinājumi neapmierina problēmas ierobežojumus. Taču otrā scenārija rezultāti ir labāki, nekā pirmajā scenārijā, un pielāgotais ĢA ļauj iegūt labākus rezultātus nekā aklā OptQuest rīka pielietošana.

Problēmas izpēte HeuristicLab struktūrietvarā

Trešajā scenārijā ir lietots atvērtā koda optimizācijas struktūrietvars HeuristicLab problēmas piemērotības ainavas analīzei un optimizācijai. Lai veiktu ātrāku un vispusīgāku analīzi, tika izstrādāts jauns spraudnis HeuristicLab struktūrietvaram [49], kas pilnībā atbilst transportlīdzekļu kustības grafika problēmas imitācijas modeļa [32] loģikai un funkcionalitātei.

Lai uzlabotu optimizācijas rezultātu kvalitāti, darbā tika piedāvāta transportlīdzekļu kustības grafika optimizācijas problēmas risinājumu permutācijas kodēšana. Permutācijas kodēšana ir bāzēta transportlīdzekļu maršrutēšanas problēmā ar tā saucamo Alba kodēšanu [3]. Hromosoma sastāv no $m + n$ gēniem, kur n ir transportlīdzekļu skaits, bet m ir reisu skaits. Gēni ar vērtību no 1 līdz m kodē reisa numuru, atlikušie gēni kodē transportlīdzekļa numurus. Hromosomā tie atdala reisu secību, kas būs jāizpilda atbilstošam transportlīdzeklī.

Modelis permutācijas reprezentācijai ir papildināts ar šādu loģiku: ja transportlīdzekļa pirmajam reisam nav laika logu ierobežojumu, tad tas tiek izpildīts uzreiz pusnaktī, citā gadījumā, tas tika uzsākts tādā laikā, lai atbilstu pirmā klienta laika logiem. Nākamais reiss tika uzsākts vai nu uzreiz, vai arī ar nobīdi, lai transportlīdzeklis atbrauktu līdz brīdim, kad sākas laika logs. Tādā veidā netika kodēti reisu uzsākšanas laiki un tas ļauj izvairīties no reisu pārklāšanās. Problēmas risināšanai tika lietoti gatavi permutācijas kodēšanas mutācijas un krustošanas operatori.

Lai salīdzinātu piemērotības ainavas rezultātus, tika veiktas vairākas eksperimentu sērijas. Problēma, kas aprakstīta avotā [32], ir dēvēta par VSP_37, un tās paplašinātā versija ar vairākiem reisiem – par VSP_133. Šīs

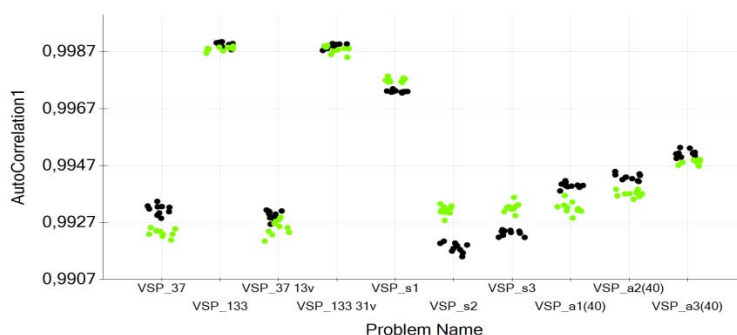
divas problēmas ir analizētas pie dažāda pieejamo transportlīdzekļu skaita. Tika izveidoti trīs problēmu gadījumi VSP_s* ar neparastiem datiem (ar īsiem reisiem un gariem laika logiem) un trīs mākslīgi izveidoti problēmas gadījumi VSP_a*. Tādā veidā ir izanalizēti 10 problēmas gadījumi.

Piemērotības ainavas analīzes eksperimenti ir sadalīti 3 lielās grupās:

1. Veselu skaitļu vektoru kodēšanas dažādu operatoru salīdzināšana.
2. Stohastisku imitācijas modeļa vērtību ietekmes piemērotības ainavas struktūrā noteikšana.
3. Veselu skaitļu un permutācijas kodēšanas salīdzināšana.

Veselu skaitļu reprezentācijai tiek analizētas divu operatoru piemērotības ainavas. Abi operatori nejauši izvēlas gēnu gadījuma pozīcijā un tie abi ar gadījuma skaitli izmaina transportlīdzekli, kas ir piešķirts esošajam reisam. Vienas pozīcijas izmaiņas manipulators (*VSPManipulator*) izmaina arī reisa uzsākšanas laiku uz jaunu ar vienmērīgi sadalītu gadījuma skaitli. Vienas pozīcijas nobīdes manipulators (*VSPShiftManipulator*) nobīda reisa uzsākšanas laiku uz vienmērīgi sadalītu gadījuma skaitli no intervāla [-400; 400]. Ainavas analīzes eksperimenti tika izpildīti ar 20000 soļu apstaigām. Tāpat tika veiktas 10 analīzes replikācijas katram mutācijas operatoram, ainavas apstaigas tipam un problēmas gadījumam.

Gadījuma apstaigā autokorelācijas vērtības $\rho(1)$ ir nedaudz mazākas *VSPManipulator* operatoram, nekā otrajam operatoram, jo šis operators ļauj veikt lielākas risinājuma izmaiņas. Adaptīvajā uz-leju-uz-augšu apstaigā ir pretēja situācija: aizvietošanas mutācijai ir lielāka korelācija nekā nobīdes mutācijai, bet vērtības mākslīgajām problēmām atšķiras (sk. 5. att.). Turklāt autokorelācija ir lielāka problēmām ar lielāku reisu skaitu. Autokorelācija neitrālās apstaigās ir līdzīga autokorelācijai gadījuma apstaigās.



5. att. Autokorelācijas $\rho(1)$ vērtība dažādiem problēmu gadījumiem (melns – aizvietošanas mutācija; zaļš – nobīdes mutācija)

Gadījuma apstaigas informācijas analīzē netika iegūta papildus informācija. Uz-leju-uz-augšu apstaigas informācijas saturs dažādiem operatoriem ir atšķirīgs un tas būtiski atšķiras dažādiem problēmas gadījumiem. Informācijas saturs ir augstāks problēmām ar mazāku autokorelāciju. Ainavu neitrālajā apstaigā ir izdalāma viena problēma

(VSP_s2), kurai būtiski atšķiras informācijas saturs, kas rāda, ka šai problēmai ir augstāka neitralitāte, kas netika novērots statistiskajā analīzē ļoti augstas autokorelācijas dēļ.

Ainavu mēru vērtības ir vienādas problēmu ar vienādiem ieejas datiem, bet ar dažādu pieejamo transportlīdzekļu skaitu, analīzē. No tā var secināt, ka transportlīdzekļu kustības grafika problēmas transportlīdzekļu skaits maz ietekmē piemērotības ainavas struktūras, bet galvenie ietekmējošie faktori ir reisu skaits un to daudzveidība.

Imitācijas modeļa stohastisko mainīgo ietekmes novērtēšanai ir pievienota papildus loģika imitācijas modelim izstrādātajā spraudnī. Visi pārvietošanas laiku garumi tiek gadījuma veidā palielināti vai samazināti ar trīsstūra sadalījuma gadījuma vērtību ± 20 minūtes intervālā. Katra modeļa analīzei ir veikti 10 eksperimenti ar dažādiem 20000 soļu apstaigas tipiem un izmantojot *MultiVSPManipulator* mutācijas operatoru, kas gadījuma veidā lieto vai nu *VSPManipulator*, vai arī *VSPShiftManipulator* operatoru.

Autokorelācijas vērtība $\rho(1)$ ir mazāka stohastisku problēmu piemērotības ainavām, izņemot reālu problēmu gadījuma apstaigā iegūtos datus, kur starpība starp autokorelācijas vērtībām bija niecīga. Tajā pat laikā šī starpība ir lielāka problēmām ar mākslīgiem datiem, kur reisu secības var būt ļoti kompaktas. Pie visām apstaigas stratēģijām informācijas satura vērtībām $H(0)$ ir ievērojama atšķirība, kas ir pamatots ar stohastisku problēmu lielāku entropiju. Viena veida trokšņa pievienošanai ir dažāda ietekme uz atšķirīgiem problēmu gadījumiem, un tā var būt novērtēta ar $H(0)$ mēru gadījuma apstaigās. Līdzīgi rezultāti ir iegūti daļējam informācijas saturam $M(0)$ – tā vērtības ir augstākas problēmām ar stohastiskiem datiem, un pretēji blīvuma baseina informācija ir lielāka problēmām ar determinētiem pārvietošanas intervāliem, tā kā šādām problēmām ir lielākie pievilksanas baseini un mazāka modalitāte.

Lai noteiktu, kādā veidā imitācijas modeļa replikāciju skaits ietekmē piemērotības ainavas analīzes rezultātus, ir veikti papildus eksperimenti: veikta eksperimentu sērija VSP_37 problēmai ar determinētiem intervāliem un eksperimentu sērijas ar šīs problēmas stohastisko versiju - 1, 5 un 10 imitācijas modeļa replikācijām. Katrā eksperimentu sērijā ir izpildītas 20 gadījuma apstaigas ar 10000 soļiem. Analīzē netika novērota ievērojama atšķirība starp autokorelācijas un korelācijas garuma vērtībām pie dažāda replikāciju skaita. Informācijas satura vērtības ir augstākas modelim ar stohastiskiem datiem, un ir vēl augstākas pie maza replikāciju skaita. Daļēja informācijas satura un blīvuma-baseina informācijas mēri rāda līdzīgu uzvedību. Informācijas mēri ir ļoti jutīgi pret troksni piemērotības funkcijā un pie lielāka imitācijas modeļa replikāciju skaita ietekme kļūst mazāka.

Salīdzinot modeļus ar dažādu reisu skaitu, tika iegūti šādi rezultāti. Visos apstaigas tipos autokorelācija $\rho(1)$ problēmai ar determinētiem datiem ir augstāka, nekā stohastiskai problēmai, un starpība starp tām ir lielāka

problēmām ar lielāku reisu skaitu. Stohastiskam modelim informācijas saturs $H(0)$ ir daudz lielāks problēmām ar lielāku reisu skaitu. Bet ir jāatzīmē, ka gadījuma apstaigā lielu nestohastisku problēmu informācijas saturs ir mazāks problēmas citas struktūras dēļ.

Lai salīdzinātu dažādu transportlīdzekļu kustības grafika optimizācijas problēmas reprezentācijas veidus, katram reprezentācijas veidam tika veikti piemērotības analīzes 10 eksperimenti ar trijām apstaigas stratēģijām un 20000 soļiem. Visām problēmām uz-leju-uz-augšu gadījuma apstaigas autokorelācijas vērtības ir mazākas permutācijas kodēšanai, kas nozīmē, ka minētajā reprezentācijā piemērotības ainavas ir negludākas. Izņēmums ir neitrālās apstaigas, kur neparastām problēmām permutācijas reprezentācijas ainavas autokorelācija ir augstāka, kas nozīmē, ka šādu problēmu ainavām ir lielāka neitralitāte permutācijas kodēšanā.

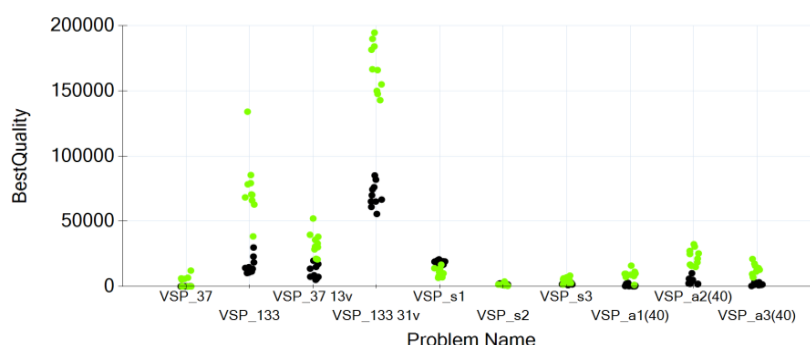
Gadījuma apstaigā iegūtās informācijas satura vērtības ir neatkarīgas no reprezentācijas veida, izņemot VSP_s1 un VSP_s2 problēmas permutācijas reprezentācijā, kur tie ir ļoti zemi. Uz-leju-uz-augšu apstaigās informācijas satura vērtības ir augstākas visām problēmām permutācijas reprezentācijā. Neitrālās apstaigas informācijas saturs pretēji ir mazāks visām problēmām permutācijas kodēšanā.

Salīdzinošajos optimizācijas eksperimentos tiek pielietota evolūcijas stratēģija (ES), imitēta atkvēlināšana (SA) un ģenētiskais algoritms (ĢA). Veselo skaitļu kodēšanā ES un SA algoritmi ir ātri un ļoti efektīvi, bet ES spēj atrast labākus risinājumus. ĢA ar pēcnācēju selekciju [2] spēj atrast vēl labākus risinājumus, bet ar lielāku modeļa novērtēšanas skaitu. No eksperimentiem ir secināts, ka permutācijas kodēšana ir efektīvāka transportlīdzekļu kustības grafika problēmas atrisināšanai, un to lietojot optimizācijas algoritmi strādā ātrāk. Kaut gan permutācijas kodēšanai ir sarežģītāka un negludāka pārmeklēšanas telpa, tomēr, līdz ar to, ka pārmeklēšanas telpa paliek mazāka, globālā optimuma meklēšana kļūst efektīvāka. ES ir efektīvāks nekā ĢA arī permutācijas kodēšanā.

Optimizācijas eksperimenti veikti vienādiem problēmas gadījumiem rāda, ka ir svarīgas sakarības starp piemērotības ainavas analīzes vērtībām un optimizācijas veikspēju. Vienā optimizācijas eksperimentu sērijā tika lietots ģenētiskais algoritms ar populācijas izmēru 100 indivīdi un 500 paaudzēm, savukārt, otrajā sērijā tika lietota evolūcijas stratēģija (20+100) ar krustošanu un 1000 paaudzēm.

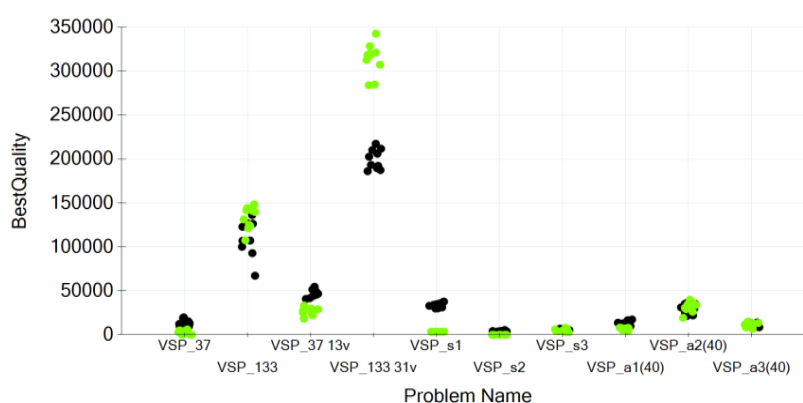
Salīdzinot dažādu mutācijas operatoru optimizācijas rezultātus, var redzēt, ka statistiskā analīze var prognozēt mutācijas operatora veikspēju. Šajā gadījumā ĢA nobīdes operators ir labāks VSP_s1 un VSP_s2 problēmām, kurām ir novērota lielāka nobīdes mutācijas operatora autokorelācija (sk. 5. un 6. att.). Līdzīga sakarība tika atrasta arī evolūcijas stratēģijas algoritmam.

Evolūcijas stratēģijas optimizācijas eksperimentos lielāka starpība starp stohastiskiem un nestohastiskiem problēmu gadījumiem ir problēmām ar lielāku informācijas saturu, bet ģenētiskais algoritms tādu sakarību nerāda.



6. att. ES atrastā labāku risinājumu piemērotība (melnie punkti – aizvietošanas mutācija; zaļie – nobīdes mutācija)

Salīdzinot dažādas reprezentācijas, interesantākie rezultāti ir iegūti lietojot ĢA. Tādējādi vairumam problēmu, kur reisu skaits ir mazs, algoritms, izmantojot permutācijas kodēšanu, rāda labāku veikspēju, bet lielām problēmām (piem., VSP_133) algoritma veikspēja pie permutācijas kodēšanas ir mazāka (sk. 7. att.). Iespējamais izskaidrojums ir tas, ka permutācijas kodēšanai lielāka efektivitāte ir mazākas pārmeklēšanas telpas dēļ. Kā tas bija noteikts statistiskajā analīzē, permutācijas reprezentācijas ainavām ir jābūt sarežģītākām meklēšanā. Līdz ar to, ka VSP_133 problēmas pārmeklēšanas telpa ir ļoti liela, piemērotības ainavas faktors sāk dominēt. Tajā pat laikā optimizējot problēmu ar ES, permutācijas reprezentācija bija efektīvāka gandrīz visiem problēmas gadījumiem.



7. att. ĢA labāko atrasto risinājumu piemērotība (zaļie punkti – permutācijas reprezentācija, melnie – veselu skaitļu)

Nodaļas beigās ir definētas rekomendācijas. Pirmā nosaka, ka transportlīdzekļu kustības plāna optimizācijas problēmas atrisināšanai vislabāk ir lietot evolūcijas stratēģiju. Gadījumā, ja optimizācija ir veikta ar

ģenētisko algoritmu, ir jāizvēlas permutācijas kodēšanu, bet tikai, ja problēmai ir mazāk nekā 100 reisi. Veselo skaitļu kodēšanā atbilstošu mutācijas operatoru izvēlas, balstoties uz piemērotības ainavas analīzes rezultātiem. Ir jāizvēlas operatoru, kuram uz-leju-uz-augšu apstaigā iegūtā autokorelācijas vērtība ir lielāka.

4. Metožu izmantošana preču piegādes plānošanā

Nodaļā ir risināti divi apvienoti integrētas preču piegādes plānošanas metodoloģijas [29, 30] optimizācijas uzdevumi. Uzdevumi tika risināti operatīvā līmenī. Kā ieejas datus uzdevumi lieto veikalu sarakstu un piegādājamo preču plānotos apjomus, kas ir iegūti metodoloģijas iepriekšējos soļos.

Šī līmeņa pilns uzdevums ir noteikt transportlīdzekļiem, kas piegādā preces no sadales centra (SC), labākus maršrutus un grafikplānu. Transportlīdzekļu sadalījuma optimalitātes sasniegšanai ir jāsamazina izmantoto transportlīdzekļu skaits un jāsamazina piegāžu kopējais attālums, minimizējot transportlīdzekļu dīkstāves laiku. Pie tam plānam ar maršrutiem un grafikplānu ir jāatbilst ierobežojumiem, tādiem kā transportlīdzekļu ietilpība, laika logi, un noliktavas iespējas

Salikta transportlīdzekļu maršrutēšanas un grafikplāna ar laika logiem uzdevuma atrisināšanai tika izmantotas divas līdzīgas, bet ne identiskas problēmas nostādnes, metaheiristiskie algoritmi un HeuristicLab optimizācijas rīks.

Divas ar pielāgotiem rīkiem secīgi risināmas optimizācijas problēmas tika izmantotas kompleksa transportlīdzekļu maršrutu un grafikplāna optimizācijas uzdevuma atrisināšanai. Pirmais uzdevums tika risināts kā klasiska transportlīdzekļu maršrutēšanas problēma ar laika logiem (TMPLL), kuras mērķis ir transportlīdzekļu maršrutu optimizācija. Otrajam uzdevumam tika definēta problēmas nostādne, kuras uzdevums ir iepriekšdefinētu maršrutu grafikplāna optimizācija.

Transportlīdzekļu maršrutēšanas problēmas nostādne

Klasiska transportlīdzekļu maršrutēšanas problēma (TMP, angl.: *vehicle routing problem*) ir vairāku komivojažieru problēmas (*multi travelling salesman problem*) veids, kurā katrai pilsētai ir definēts pieprasījums, bet komivojažieri ir interpretēti kā transportlīdzekļi ar vienādu ietilpību. Pieprasījumu summa maršruta ietvaros nevar pārsniegt transportlīdzekļa ietilpību un ir nepieciešams minimizēt maršrutu garumu summu. Ierobežotas ietilpības TMP (ITMP, angl.: *capacitated VRP*) nav pilnīgi ģeogrāfiska, jo ietilpība var būt kā ierobežojošs faktors. Gadījumā, ja katram klientam ir uzdots laiksprauga, kuras periodā klientu ir jāapmeklē, tad tiek iegūta TMP ar laika logiem (TMPLL, angl.: *VRPTW*). Kopā ar ietilpības ierobežojumu transportlīdzeklim ir jāapmeklē arī klients noteiktā laikposmā, kas ir uzdots ar uzsākšanas un pabeigšanas laiku. Parasti ir atļauta situācija, kad

transportlīdzeklis var atbraukt pirms uzsākšanas laika, bet ir aizliegts atbraukt pēc pabeigšanas laika. Visām šīm problēmām kopīgs ir tas, ka tās ir „sarežģītas” risināšanai. Bieži vien klientu skaits kombinācijā ar reālu lietišķās problēmas datu sarežģītību neļauj risināt problēmu ar eksaktām metodēm. Tādos gadījumos tiek rekomendēts izmantot aptuvenus algoritmus vai heuristikas [2].

Klasiska TMPLL matemātiskais formulējums ir balstīts uz modeli, kas ir definēts avotā [7]. TMPLL aprakstā ir uzdots ar viendabīgu transportlīdzekļu V kopu, klientu C kopu un orientētu grafu G . Grafs sastāv no $|C|+2$ virsotnēm, kur klienti ir definēti kā $1,2,\dots,n$ un depo ir definēts ar virsotnēm 0 un $n+1$. Virsotņu kopa tiek dēvēta par N . Loku kopa A atbilst savienojumiem starp klientiem un starp depo un klientiem, kur neviens loks nebeidzas virsotnē 0 un neviens loks nesākās no virsotnes $n+1$. Ar katru loku (i, j) , kur $i \neq j$ ir asociētas izmaksas c_{ij} un laiks t_{ij} , kas var ietvert apkalpošanas laiku pie klienta i . Katram transportlīdzeklim ir noteikta ietilpība q_j un katram klientam i ir noteikts pieprasījums d_i . Pie tam, katram klientam i ir noteikts laika logs $[a_i, b_i]$. Transportlīdzeklis var atbraukt pirms a_i , tomēr transportlīdzeklim ir jāierodas pie klienta līdz b_i . Vispārīgajā aprakstā depo arī ir noteikts savs laika logs, jeb plānošanas horizonts: transportlīdzekļiem nav atļauts atstāt depo pirms a_0 un tiem ir jāatgriežas ne vēlāk kā b_{n+1} . Uzdevumā ir izvirzīta prasība, ka q, a_i, b_i, d_i , un c_{ij} ir veseli nenegatīvie skaitļi, tajā laikā t_{ij} ir jābūt pozitīvam veselam skaitlim. Šis modelis satur divas lēmumu mainīgo kopas, attiecīgi x un s . Katram lokam (i, j) , kur $i \neq j, i \neq n+1, j \neq 0$ un katram transportlīdzeklim k ir definēts x_{ijk} : ja transportlīdzeklis k nebrauc no virsotnes i uz virsotni j , tad $x_{ijk} = 0$, un pretēji $x_{ijk} = 1$ ja transportlīdzeklis k brauc no virsotnes i uz virsotni j . Lēmumu mainīgais s_{ik} tiek definēts katrai virsotnei i un katram transportlīdzeklim k , nosakot laiku, kad transportlīdzeklis k uzsāk klienta i apkalpošanu. Ir pieņemts, ka $a_0 = 0$ un tādējādi visiem k ir $s_{0k} = 0$.

Uzdevuma mērķis ir konstruēt tādu maršrutu kopu ar minimālām izmaksām, lai pie tām: katrs klients tiktu apkalpots tieši vienu reizi; katrs maršruts tiktu uzsākts virsotnē 0 un beigtos virsotnē $n+1$; ir izpildīti laika logu un ietilpības ierobežojumi. TMPLL mērķa funkcijas formulējums ir šāds:

$$\sum_{k \in V} \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ijk} \rightarrow \min. \quad (32)$$

Problēmas nostādnē [7] ir definēti arī vairāki ierobežojumi: tie prasa, lai katrs klients tiktu apmeklēts tieši vienu reizi un, lai neviens transportlīdzeklis nav pārkrauts; tie nodrošina, ka katrs transportlīdzeklis secīgi apmeklēs maršruta punktus; nodrošina, ka transportlīdzeklim k , kas brauc no i uz j , nav atļauts atbraukt pie j agrāk, nekā $s_{ik} + t_{ij}$ un nodrošina, ka tiek ņemti vērā laika logi. Šajā uzdevuma nostādnē neizmantotais transportlīdzeklis tika modelēts kā tukšu maršrutu braucošs [2].

Transportlīdzekļu maršrutēšanas eksperimenti

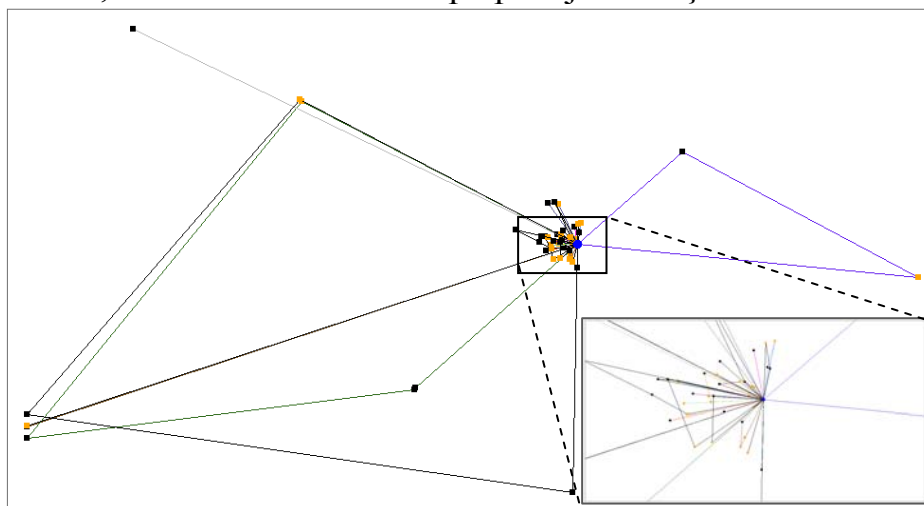
Lai veiktu transportlīdzekļu maršrutēšanas eksperimentus, tika sagatavoti ieejas dati, kas ir bāzēti uz apskatītā uzņēmuma vēsturiskajiem datiem. Eksperimenti tika sagatavoti, balstoties uz produktu piegādes plānu vienai izvēlētai dienai un preču grupām, kuras tiek piegādātas kopā vienā transportlīdzeklī. Maršrutēšanas uzdevuma datos tika definēts kopējais veikalu saraksts, kuros ir jāpiegādā preces noteiktajā dienā. Katrai preču grupai ir veiktas atsevišķas eksperimentu sērijas: “Dry” tipa preces; “Cooler” un “F&V” tipa preces kopā. Vienai un tai pašai dienai var būt definēti atšķirīgi veikali dažādiem preču grupām.

Visi eksperimenti ir izpildīti, izmantojot Salu un pēcnācēju selekcijas ģenētisko algoritmu (*Island Offspring Selection Genetic Algorithm, IOSGA*) [2], kas ir speciāls ģenētiskā algoritma veids, kas apvieno salu ĢA un ĢA ar pēcnācēju selekciju. Algoritma izvēle tiek pamatota ar to lielo optimizācijas spiedienu un pielāgojamību. Maršrutēšanas eksperimenti ir izpildīti HeuristicLab optimizācijas struktūrietvarā [49], izmantojot esošos IOSGA un TMP (*VRP*) struktūrietvara spraudņus. Visos eksperimentos ar IOSGA ir definēti šādi algoritma parametri: proporcionālā selekcija un HeuristicLab rīka *MultiVRPMutator* mutācijas operators ar 5% likmi. Krustošanas operators tika noteikts eksperimentāli. Populācijas izmērs ir 200 indivīdi, ieskaitot vienu elites indivīdu. Populācija tika sadalīta 5 salās un labāko risinājumu migrācija ir veikta katrā 20-tajā paaudzē. Apstāšanās nosacījums ir definēts kā maksimālais pēcnācēju selekcijas spiediens, kas ir lielāks par skaitli 200.

HeuristicLab TMP spraudnim ir sagatavoti un definēti šādi problēmas parametri: transportlīdzekļu ietilpība roll-konteineros (ruļļi); katra veikala un sadales centra koordinātes; veikalu dienas pieprasījums ruļļos; laika logu uzsākšanas un pabeigšanas laiki minūtēs no pusnakts; katra veikala apkalpošanas laiks minūtēs; pieejamo transportlīdzekļu skaits. Apskatāmajā problēmā koordinātes tika definētas tādā veidā, lai Eiklīda distance starp diviem punktiem būtu aptuveni vienāda ar braukšanas laiku starp dotiem punktiem (minūtēs).

Lai problēmai noteiktu piemērotāko krustošanas operatoru, tika veikta eksperimentu sērija. Tika salīdzināti GVR krustošana [36], loku rekombinācija (ERX) [34] un maksimālās saglabāšanas (MPX) [51] krustošanas operatori. Kā rezultātā tika noteikts, ka ERX krustošana rāda labākus rezultātus, optimizējot kopējo distanci, savukārt, GVR krustošana labāk kontrolē pārslogotības ierobežojumus. Kā labākais tika izvēlēts GVR krustošanas operators, tā kā tas ļauj uzturēt nepārslogotus maršrutus, kaut gan neievēro pieejamo transportlīdzekļu skaita ierobežojumu. Lai samazinātu nepieciešamo transportlīdzekļu skaitu, zemāk ir piedāvāta un atrisināta transportlīdzekļu maršrutu grafikplāna optimizācijas problēma.

Viena algoritma palaišanas rezultāti „Dry” (sauso) preču tipu piegādei izvēlētajai dienai ir parādīti 8. attēlā. Ar zilo riņķi ir apzīmēts sadales centrs, melnie punkti apzīmē veikalu vietas un dzelteni punkti tos veikalus, kuri maršrutā tika apkalpoti pirmie. Visu transportlīdzekļu ietilpība ir uzdota vienāda – 30 konteineri. Var redzēt, ka algoritms piedāvā visus klientus, kas atrodas ārpus centrālās lielās pilsētas, apkalpot ar gariem maršrutiem, kas ietver no 3 līdz 4 veikaliem. Tomēr daži veikali ir jāapkalpo ar atsevišķu transportlīdzekli, konkrētā veikala lielā pieprasījuma dēļ.



8. att. Sauso preču piegādes maršruti vienai dienai

Piedāvātos maršrutus centrālajai lielpilsētai var apskatīt 9. attēlā labajā pusē. Lielpilsētas piegādes maršruti galvenokārt ir paredzēti 1, 2 vai 3 veikalu apmeklēšanai. Veikalu skaits, kurus ir jāapkalpo individuāli, ir lielāks, kas ir saistīts ar lielāku lielpilsētas veikalu pieprasījumu. Visi eksperimenti beidzās pēc 30-50 paaudzēm, kad selekcijas spiediens pārsniedza skaitli 200. Eksperimentos tika iegūts, ka visu veikalu apkalpošanai ir nepieciešami 34 transportlīdzekļi. Taču, līdz ar to, ka vairāki maršruti ir ļoti īsi, tos varētu apvienot vienam transportlīdzeklim vairākkārt atgriežoties sadales centrā, kas nav paredzēts klasiskajā TMP nostādnē.

Līdzīgā veidā tika veikti eksperimenti „POD Cooler” un „F&V” preču tipu piegādes maršrutu optimizācijai. Tika pieņemts, ka abu preču grupas var tiks pārvestas kopā vienā transportlīdzeklī. Pirmajā algoritma palaišanā transportlīdzekļu ietilpība tika definēta kā 30 konteineri. Izejas dati demonstrē, ka vairākos gadījumos uzdotā transportlīdzekļu ietilpība nav pietiekama visu preču transportēšanai un uzņēmumam būs jāsaņem liels skaits mazas ietilpības transportlīdzekļu. Turklāt transportlīdzekļu izlietošanas vērtība izejas datos tādā gadījumā ir vienāda ar 46 vienībām, kas ir daudz lielāka, nekā pieejamo transportlīdzekļu daudzums. Otrā eksperimentu sērija “FV+POD” precēm tika veikta ar transportlīdzekļiem, kuru ietilpība ir vienāda ar 60 konteineriem. No eksperimentu rezultātiem var secināt, ka lielākas transportlīdzekļu ietilpības maršruti ir lielākais apkalpojamo

veikalu skaits. Turklāt ir formētas vairāku veikalu struktūras, kuras var būt definētas kā reģioni, kas ir aprakstīti reģionu plānošanas uzdevumā, kas ietilpst integrētajā preču piegādes metodoloģijā [30]. Otrajā gadījumā ir izmantoti tikai 24 transportlīdzekļi. Salīdzinot maršrutos centrālajā daļā, kur ir liels veikalu blīvums, var secināt, ka, veikalu skaits maršrutos ir ierobežots transportlīdzekļu ietilpības dēļ, nevis īso laiku logu dēļ.

Eksperimentu rezultāti rāda, ka HeuristicLab ietvara TMP spraudņa izmantošana kopā ar salu pēcnācēju selekcijas ģenētiskā algoritma lietošanu var efektīvi un pietiekami ātri risināt transportlīdzekļa maršrutu optimizācijas uzdevumu projektā apskatītajam loģistikas uzņēmumam: eksperimenti dienas plānam aizņem dažas minūtes.

Maršrutu grafikplāna optimizācijas problēmas nostādne

TMPLL nostādne tika pieņemta, ka viens transportlīdzeklis var izpildīt tikai vienu maršrutu plānošanas horizonta ietvaros. Gadījuma izpētē visi maršruti tika saīsināti transportlīdzekļu mazas ietilpības dēļ, kas rada neefektīvus maršrutēšanas problēmas risinājumus. Lai atrisinātu šo jautājumu, darbā tika piedāvāta maršrutu grafikplāna optimizācijas problēma. Tas var būt formulēts līdzīgi transportlīdzekļu kustības plānošanas problēmai ar laika logiem un to var atrisināt ar metodēm un rīkiem, kas ir izstrādāti darba 3. nodaļā. Formulētajā problēmā maršruti atbilst kustības grafika problēmas reisiem. Transportlīdzekļiem dienas laikā ir atļauts izpildīt vairākus maršrutus. Līdz ar to, ka TMPLL galējais risinājums atbilst transportlīdzekļu ietilpības un laika logu ierobežojumiem, to var optimizēt, kombinējot maršrutus un palielinot transportlīdzekļu izmantošanu. Maršrutu grafikplāna optimizācijas izmantošana ļauj samazināt nepieciešamo transportlīdzekļu skaitu.

Problēmas nostādne ir aprakstīta 3. nodaļā un [28] ir modificēta. Maršrutēšana tika veikta katrai preču grupai atsevišķi. Visiem klientiem tika definēti laika logi un apkalpošanas laiks. Ieejas dati, kas tika izmantoti maršrutēšanā tika lietoti arī kustības plānošanā. Veikalu virkne reisos jaunajā nostādne ir definēta kā maršruts, savukārt, pārvietošanas laiki ir definēti kā transportēšanas laiki. Transportlīdzekļa ietilpība problēmā netika kontrolēta.

Maršrutu grafikplāna optimizācijas problēmas formālā nostādne iekļauj klientu (veikalu) kopu N un maršrutu sarakstu R , kas tika iegūta no TMPLL risinājuma. Katrs maršruts definē apkalpojamo klientu secību. Nostādne satur transportlīdzekļu pārvietošanas laikus t_{ij} starp maršruta veikaliem i un j un transportlīdzekļu kopu V ar pieejamo transportlīdzekļu skaitu $|V|$. Katram veikalam i ir definēta tā laika loga uzsākšanas a_i un pabeigšanas b_i laiki, kā arī apkalpošanas laiks z_i minūtēs.

Mērķa mainīgie ir līdzīgi maršrutēšanas modeļa mērķa mainīgajiem, un satur kopas x un s , izņemot to, ka $x_{ijk} = 1$ apzīmē, ka transportlīdzeklim k maršruts j seko pēc maršruta i . Problēmai ir doti divi nestingri

ierobežojumu tipi: 1) laika logi; 2) virsstundu ierobežojums. Problēmā laika logu ierobežojums ir noteikts gadījumu skaits N_{ad} , kad transportlīdzeklis aizbrauc no klienta pēc tā laika loga pabeigšanas. Problēmas nostādne ļauj transportlīdzeklim atbraukt agrāk nekā laika loga uzsākšanas laiks, bet laika loga atvēršanas gaidīšanas laiks tika novērtēts kā dīkstāves laiks. Problēmas virsstundu ierobežojums ir novērtēts kā tādu transportlīdzekļu skaits N_{ot} , kuriem nācās strādāt virs 24 stundām. Nostādnē tika definēti papildus ierobežojumi, kas nosaka modeļa integritāti un grafikplāna imitācijas modelēšanu.

Maršrutu grafikplāna optimizācijas problēmas mērķa funkcija f apvieno transportlīdzekļu kopējo dīkstāves laiku un ierobežojumu neievērošanas skaitu reizinātu ar soda koeficientiem:

$$f = \sum_{k \in V} l_k + p_{ad} N_{ad} + p_{ot} N_{ot} \rightarrow \min, \quad (33)$$

kur l_k ir k -tā transportlīdzekļa kopējais dīkstāves laiks; V ir pieejamo transportlīdzekļu kopa; N_{ad} un N_{ot} ir atbilstošu ierobežojumu nenovērošanas gadījumu skaits; p_{ad} un p_{ot} ir koeficienti risinājuma piemērotības pasliktināšanai, un tiem jābūt daudz lielākiem par 1.

Tā kā maršrutu grafikplāna optimizācijas problēma ir atvasināta no augstāk aprakstītās transportlīdzekļu kustības grafika problēmas, tās atrisināšanai labāk izmantot evolūcijas stratēģijas algoritmu, jo šī metaheiristika rādīja labākus rezultātus kustības grafika problēmas atrisināšanā.

Transportlīdzekļu grafikplāna optimizācijas eksperimenti

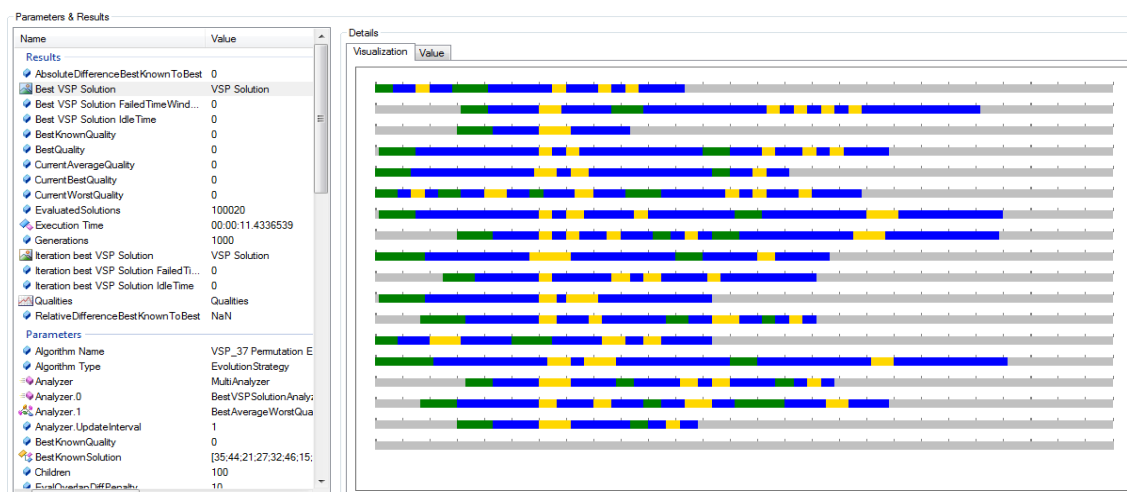
Darbā tika veikti transportlīdzekļu kustības grafika optimizācijas eksperimenti divās grupās ar gadījuma izpētes ieejas datiem. Pirmā eksperimentu grupa ir bāzēta transportlīdzekļu kustības grafika optimizācijas problēmas nostādnē, kas ir formulēta darba 3. nodaļā. Otrā grupa ir bāzēta 4. nodaļas maršrutu grafikplāna optimizācijas problēmas nostādnē.

Transportlīdzekļu kustības grafika optimizācijas eksperimenti tika veikti HeuristicLab optimizācijas struktūrietvarā, izmantojot izstrādāto kustības plānošanas problēmas spraudni. Optimizācija ir veikta lietojot evolūcijas stratēģijas (20+100) algoritmu. Apstāšanās nosacījums ir 1000 paaudzes, bet mutācijas operators ir atkarīgs no izvēlētās kodēšanas. Lai attēlotu transportlīdzekļu kustības plāna optimizāciju, tika veiktas divas eksperimentu sērijas: ar veselu skaitļu un ar permutācijas kodēšanu.

Grafikplāna risinājuma Ganta diagramma, kura ir attēlota izmantotā spraudņa galvenajā logā, ir parādīta 9. att. Attēlā ir parādīts risinājums, kas ir iegūts optimizācijā, lietojot permutācijas kodēšanu. Veselu skaitļu kodēšanas risinājumā paliek lieli dīkstāves laiki, kaut arī visi ierobežojumi ir izpildīti. Permutācijas kodēšanas risinājumiem ir labāka kvalitāte. Visi reisi ir kompakti apvienoti un dīkstāves laiki ir izslēgti. Transportlīdzekļu skaitu var samazināt tā, ka viens transportlīdzeklis paliek bez reisiem.

Lai izveidotu risināšanas rīku transportlīdzekļu maršrutu grafikplāna optimizācijas problēmai, ir izstrādāts HeuristicLab optimizācijas struktūrietvara problēmas spraudnis. Risinājuma piemērotības novērtēšanai spraudnis imitē risinājuma kandidāta grafikplānu un nosaka grafikplāna neatbilstību laika logiem, novērtē dīkstāves laikus un kopējo transportlīdzekļu darba laiku. Piemērotības funkcija ir novērtēta ar (33).

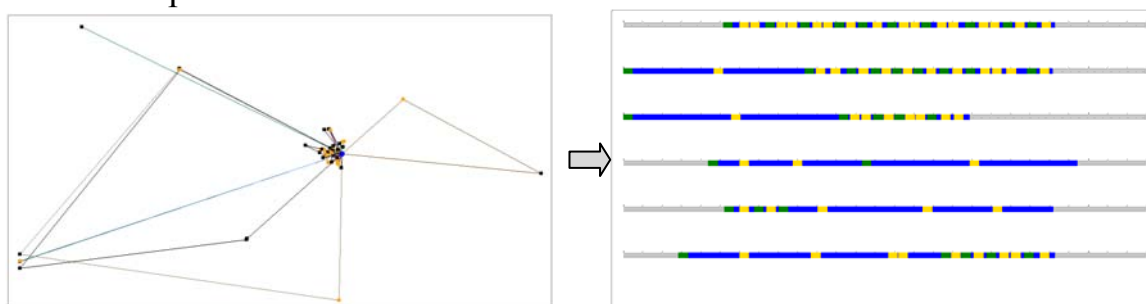
Grafikplāna optimizācijai ir lietota transportlīdzekļu kustības plāna optimizācijas problēmas permutācijas kodēšana, bet reisu numuri šeit attēlo maršrutu numurus. Spraudņa ieejas dati ir veikalu laika logu uzsākšanas un pabeigšanas laiki, apkalpošanas laiku vektori, maršrutu un pārvietošanas laiku starp maršrutu punktiem saraksti un pieejamo transportlīdzekļu skaits. Visi laiki ir doti minūtēs, laika logu uzsākšanas un pabeigšanas laiki ir doti no pusnakts.



9. att. Transportlīdzekļu kustības grafika optimizācijas problēmas risinājums

Lai noteiktu piemērotāko algoritmu maršrutu grafikplāna optimizācijas problēmas risināšanai tika veiktas vairākas optimizācijas eksperimentu sērijas. Tika izpētīti šādi algoritmi: evolūcijas stratēģija (ES), ģenētiskais algoritms, salu ģenētiskais algoritms ar 5 salām (IGA) un ģenētiskais algoritms ar pēcnācēju selekciju (OSGA) [2]. Maksimālas saglabāšanas krustošana (MPX) [34] un ievietošanas mutācija tika definēti kā ģenētiskie operatori. Ģenētiskajos algoritmos ir lietota proporcionālā selekcija. Lai noteiktu piemērotāko algoritmu, tika salīdzināts visu algoritmu risinājumu novērtēšanas daudzums, kas ir nepieciešams, lai sasniegtu labākā atrastā risinājuma līdzīgu piemērotības vērtību. Eksperimentos ir izpētītas sarežģītas problēmas ar mazu pieejamo transportlīdzekļu skaitu. Evolūcijas stratēģijas algoritms tika izvēlēts, jo tas ļauj atrast globāli optimālus risinājumus ar mazāku risinājumu novērtējumu skaitu. Tā kā maršrutu grafikplāna optimizācijas problēma ir atvasināta no transportlīdzekļu kustības grafika problēmas, optimizācijas algoritmi rādīja līdzīgu veikspēju.

Darbā ir aprakstīts parauga eksperiments, kas ir bāzēts vienas dienas piegādes plāna datos ar 53 veikaljiem. IOSGA algoritma pielietošana TMPLL uzdevumā parāda risinājumu ar 34 maršrutiem (sk. 10. att.). Lielākā maršrutu daļa ir ļoti īsi transportlīdzekļu ietilpības dēļ, bet līdz ar to, ka laika logi ir lieli, ir iespējams kombinēt šos maršrutus. Tika optimizēts maršrutēšanas eksperimentu izejas datu grafikplāns, izmantojot (20+100) evolūcijas stratēģijas. No rezultātiem var secināt, ka problēmai eksistē globāli optimāls risinājums, kuru var izpildīt ar 6 transportlīdzekļiem. Atbilstoša risinājuma Ganta diagramma ir parādīta 10. attēlā. Zaļās līnijas ir iekraušanas intervāls sadales centrā, zilās ir pārvietošanas intervāli, savukārt, dzeltenās apzīmē izkraušanas intervālus.



10. att. Maršrutēšanas un grafikplāna risinājums

Piedāvātā transportlīdzekļu maršrutu grafikplāna optimizācijas metode, kas papildina transportlīdzekļa maršrutēšanu, nodrošina efektīvus maršrutu un grafikplāna risinājumus īsā laika piegāžu plānošanā. Metodi var izmantot maršrutēšanas un grafikplāna optimizācijas uzdevumos, kuros maršruti ir īsi salīdzinot ar plānošanas horizontu.

PROMOCIJAS DARBA REZULTĀTI UN SECINĀJUMI

Promocijas darba mērķis bija metožu un algoritmu izstrāde imitācijas modelēšanā sakņotai sarežģītu sistēmu piemērotības ainavas analīzei un optimizācijai.

Promocijas darba rezultāti un secinājumi ir šādi:

1. NP-sarežģītu problēmu imitācijas modelēšanā sakņotas optimizācijas literatūras izpēte ir ļāvusi izvēlēties metaheirstiskus algoritmus kā efektīvākas metodes NP-sarežģītu problēmu risināšanā. Imitācijas modelēšanā sakņotu skaitlisku optimizācijas metožu apskats, piemērotības ainavu un to struktūru formālo definīciju apskats kopā ar informācijas un statistikas piemērotības ainavas analīzes metožu apskatu ir ļāvuši izstrādāt ar piemērotības ainavas analīzi papildinātu formālu shēmu imitācijas modelēšanā sakņotai optimizācijai.

2. Izstrādātā ar piemērotības ainavu analīzi papildinātā imitācijas modelēšanā sakņotās optimizācijas formālā shēma ir ļāvusi paplašināt piemērotības ainavas analīzes metodes uz imitācijas modelēšanā sakņotas optimizācijas uzdevumiem, un ir ļāvusi noteikt prasības imitācijas modelēšanā sakņotas piemērotības ainavas analīzes metodēm, algoritmiem un lietojumprogrammas instrumentam.
3. Etalonuzdevumu ainavu eksperimentālā piemērotības ainavas analīze ar dažādiem operatoriem un reprezentācijām ir ļāvusi atrast sakarības un atkarības starp etalonuzdevumu strukturālām īpašībām, to informācijas un statistiskiem mēriem un optimizācijas algoritma uzvedību apskatītās ainavās. Trokšņaino piemērotības funkciju eksperimentālā analīze ir ļāvusi noteikt papildprasības imitācijas modelēšanā sakņotai piemērotības ainavas analīzei.
4. Izstrādātie imitācijas modelēšanā sakņotās piemērotības ainavas analīzes procedūras algoritmi ir ļāvuši izstrādāt piemērotības ainavas analīzes rīka lietojumprogrammas prototipu. Izstrādātā rīka izmantošana ir ļāvusi veikt transportlīdzekļu kustības grafika optimizācijas problēmas ar laika logiem analīzi imitācijas modelēšanā sakņotas optimizācijas ietvaros.
5. Vispusīgā transportlīdzekļu kustības grafika optimizācijas problēmas ar laika logiem eksperimentālā piemērotības ainavas analīze optimizācijas struktūrietvarā ir ļāvusi noteikt tās piemērotības ainavas problēmai specifiskās īpašības un iekšējās raksturīpašības. Šī analīze ir ļāvusi noteikt problēmas specifisko īpašību ietekmi uz piemērotības ainavas struktūru un optimizācijas algoritma veikspēju. Tas ļāva izstrādāt rekomendācijas labākiem scenārijiem transportlīdzekļu kustības grafika optimizācijai ar laika logiem.
6. Izstrādātās metodes un algoritmi tika lietoti piegādes plānošanas operāciju līmeņa optimizācijas uzdevumu risināšanā, kas ir ļāvis uzlabot apvienotās transportlīdzekļu maršrutēšanas grafikplāna optimizācijas problēmas ar laika logiem risinājumus. Pielāgotā metaheiristiskā algoritma izmantošana definētajiem maršrutēšanas un grafikplānošanas uzdevumiem ir ļāvusi samazināt nepieciešamo transportlīdzekļu skaitu un piedāvāja efektīvāku transportlīdzekļu izmantošanu.

GALVENIE INFORMĀCIJAS AVOTI

1. Ackley D. An empirical study of bit vector function optimization. // Genetic Algorithms and Simulated Annealing. - 1987. - pp. 170-215.
2. Affenzeller M., Winkler S., Wagner S., Beham A. Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications. - Chapman and Hall/CRC, 2009. - 379p.
3. Alba E., Dorronsoro B. Solving the vehicle routing problem by using cellular genetic algorithms. // Lecture Notes in Computer Science Vol. 3004. - Springer, 2004. - pp. 11-20.
4. Beerenwinkel N., Pachter L, Sturmfels B. Epistasis and Shapes of Fitness Landscapes // Statistica Sinica Vol. 17, No. 4. - 2007. - pp. 1317-1342.
5. Biles, W.E. A Gradient Regression Search procedure for Simulation Experimentation // Proceedings of the 1974 Winter Simulation Conference. - 1974. - pp. 491-497.
6. Collard P., Verel S, Clergue M. Local Search Heuristics: Fitness Cloud versus Fitness Landscape // 16th European Conference on Artificial Intelligence. - 2004. - pp. 973-974.
7. Cordeau F., Desaulniers G., Desrosiers J., Solomon M.M., Soumis F. VRP with Time Windows // Toth P. and Vigo D. (Eds.), The vehicle routing problem. - Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2001. - pp. 157-193.
8. Czech Z.J. Statistical measures of a fitness landscape for the vehicle routing problem. // IEEE International Symposium on Parallel and Distributed Processing. - 2008. - pp. 1-8.
9. De Jong K. D. An analysis of the behavior of a class of genetic adaptive systems. PhD thesis. - Ann Arbor, USA: Department of Computer and Communication Sciences, University of Michigan, 1975. - 256p.
10. Drezo J., Petrowski A., Siarry P., Taillard E. Metaheuristics for Hard Optimization. Methods and Case Studies. - Springer-Verlag, Berlin Heidelberg, 2006. - 369 p.
11. Eiiyi D.T., Ornek A., Karakutuk S.S. A vehicle scheduling problem with fixed trips and time limitations // International Journal of Production Economics, Vol. 117, No.1. - 2008. - pp. 150-161.
12. Glover, F.; Kochenberger, G.A. Handbook of metaheuristics. International Series in Operations Research & Management Science. - Springer, 2003. - 570p.
13. Glover F. Tabu Search - Part I // INFORMS Journal on Computing, Vol. 1, No. 3. - 1989. - pp. 190-206.
14. Goldberg, D. E. Genetic Algorithms in Search, Optimization and Machine Learning. - Addison-Wesley Professional, 1989. - 432 p.
15. Gosavi A. Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning. - Kluwer Academic Publishers, 2003. - 554 p.
16. He J., Yao X., Zhang Q. To Understand Fitness Landscapes in Continuous Space by Using Drift Analysis // Congress on Evolutionary Computation, CEC2004, Vol. 2. - 2004 - pp. 1248-1253.
17. Holland J.H. Adaptation in Natural and Artificial Systems. - University of Michigan Press, 1975. - 183p.
18. Hordijk W. A Measure of Landscapes // Evolutionary computation, Vol. 4. - 1996. - pp. 335-360.

19. Horn J., Goldberg D.E. Genetic Algorithm Difficulty and the Modality of Fitness Landscapes // Foundations of Genetic Algorithms, Vol. 3. - Morgan Kaufman, 1994. - pp. 243-269.
20. Jones T. Evolutionary Algorithms, Fitness Landscapes and Search. PhD Thesis. - Albuquerque: The University of New Mexico, 1995. - 224p.
21. Jones T., Forrest S. Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithms // Proceedings of the Sixth International Conference on Genetic Algorithms. – San Francisco, CA: 1997. - pp.184-192.
22. Kauffman S. Adaptation on rugged fitness landscapes // Lectures in the Science of Complexity. - Addison-Wesley, 1989. - pp. 527-618.
23. Kinnear K. E. Fitness Landscapes and Difficulty in Genetic Programming // Proceedings of the 1994 IEEE World Conference on Computational Intelligence. - Orlando, Florida, USA: 1994. - pp. 142-147.
24. Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P. Optimization by Simulated Annealing // Science, Vol. 220, No. 4598. - 1983. - pp. 671-680.
25. Knuth D. The Art of Computer Programming. Vol. 2: Seminumerical Algorithms. - Reading, Massachusetts, USA: Addison-Wesley, 1997. - 762p.
26. Langdon W.B., Poli R. Foundations of Genetic Programming. - Berlin, Heidelberg: Springer-Vorlag, 2002. - 260p.
27. Manderick B., Weger M., Spiessens P. The Genetic Algorithm and Structure of the Fitness Landscape // Proceedings of the Fourth International Conference on Genetic Algorithms. - 1991. - pp. 143-150.
28. Merkuryeva G., Bolshakov V. Simulation-based fitness landscape analysis and optimisation for vehicle scheduling problem // EUROCAST 2011, Part I, LNCS 6927. - Springer, 2011. - pp. 280-286.
29. Merkuryeva G., Bolshakov V. Simulation Optimisation and Monitoring in Tactical and Operational Planning of Deliveries // Proceedings of the European Modeling and Simulation Symposium. - 2012. - pp 226-231.
30. Merkuryeva G., Bolshakov V., Kornevs M. An Integrated Approach to Product Delivery Planning and Scheduling // Scientific Journal of RTU. 5. series., Computer Science. Vol. 49. - Riga: RTU Publishing House, 2011. - pp. 97-103.
31. Merkuryeva G., Bolshakov V. Comparative Analysis of Statistical and Information Measures for Benchmark Fitness Landscapes // Proceedings of the UkSIM Fourth European Modelling Symposium on Computer Modelling and Simulation. - 2010. - pp. 96-101.
32. Merkuryeva G., Bolshakov V. Vehicle Schedule Simulation with AnyLogic // 12th International Conference on Computer Modelling and Simulation. - Cambridge, 2010. - pp. 169-174.
33. Merz P., Freisleben B. Fitness Landscape Analysis and Memetic Algorithms for the Quadratic Assignment Problem // IEEE Transactions on Evolutionary Computation, Vol. 4, Issue 4. - 2000. - pp. 337-352.
34. Muhlenbein H. Evolution in time and space - the parallel genetic algorithm // Foundations of Genetic Algorithms. - Morgan Kaufmann, 1991. - pp. 316-337.
35. Nagamochi H., Ohnishi T. Approximating a vehicle scheduling problem with time windows and handling times // Theoretical Computer Science, Vol. 393, Issue 1-3. - 2008. - pp. 133-146.
36. Pereira, F.B., Tavares, J., Machado, P., Costa E. GVR: A New Genetic Representation for the Vehicle Routing Problem // Proceedings of the 13th Irish International Conference on Artificial Intelligence and Cognitive Science (AICS '02). - Limerick, Ireland, 2002. - pp. 95-102.

37. Pitzer, E., Affenzeller, M. A Comprehensive Survey on Fitness Landscape Analysis // *Recent Advances in Intelligent Engineering Systems*. - Springer, 2011. - pp. 167–196.
38. Pitzer E., Vonolfen S., Beham A., Affenzeller M., Bolshakov V., Merkurjeva G. Structural Analysis of Vehicle Routing Problems using General Fitness Landscape Analysis and Problem Specific Measures // *1st Australian Conference on the Application of Systems Engineering (ACASE'12)*. - Australia, Sydney: 2012. - pp. 36-38.
39. Reeves C.R., Rowe J.E. *Genetic Algorithms - Principles and Perspectives. A Guide to GA Theory*. – Springer, 2002. - 344p.
40. Reidys C.M., Stadler, P.F. Neutrality in Fitness Landscapes // *Appl. Math. & Comput.*, Vol. 117. - 1998. - pp. 321-350.
41. Schwefel H.P. *Evolution and Optimum Seeking*. - Wiley-Interscience, 1995. - 465p.
42. Sipser M. *Introduction to the Theory of Computation*. Second Edition. - Thomson Course Technology, 2006. - 431p.
43. Smith, T., Husband, P., Layzell, P., O’Shea, M. Fitness landscapes and evolvability // *Evol. Comput.*, Vol. 10, No. 1. - 2002. - pp. 1-34.
44. Stadler P.F. Fitness Landscapes // *Appl. Math. & Comput*. Vol. 117. - 2002. - pp. 187-207.
45. Stadler P.F. Towards a Theory of Landscapes // *Complex Systems and Binary Networks*. Lecture Notes in Physics, Vol. 461. - 1995. - pp. 78-163.
46. Teo J.T.W. Pareto Multi-Objective Evolution of Legged Embodied Organisms. PhD Thesis. - Sydney: University of New South Wales, 2003. - 281p.
47. Vanneschi L., Tomassini M., Collard P., Verel S. Negative Slope Coefficient: A Measure to Characterize Genetic Programming Fitness Landscapes // *Lecture Notes in Computer Science*, Vol. 3905 - Springer, 2006. - pp. 178-189.
48. Vassilev V.K., Fogarty T.C., Miller J.F. Information Characteristics and the Structure of Landscapes. // *Evolutionary Computation*, Vol. 8, Issue 1. - Cambridge, MA, USA: MIT Press, 2000. - pp. 31-60.
49. Wagner S. Heuristic Optimization Software Systems - Modeling of Heuristic Optimization Algorithms in the HeuristicLab Software Environment. PhD Thesis. - Linz, Austria: Institute for Formal Models and Verification, Johannes Kepler University, 2009.
50. Weinberger. E. Correlated and Uncorrelated Fitness Landscapes and How to Tell the Difference // *Biological Cybernetics*, Vol. 63, Issue 5. - Springer-Verlag, 1990. - pp. 325-336.
51. Whitley D.L., Starkweather T., Fuquay D. Scheduling Problems and Traveling Salesmen: The Genetic Edge Recombination Operator // *Proceedings of the 3rd International Conference on Genetic Algorithms*, J. David Schaffer (Ed.). - San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1989. - pp. 133-140.
52. Wright S. The roles of mutation, inbreeding, crossbreeding and selection in evolution. // *Proceedings of the Sixth International Congress on Genetics*, Vol. 1. - 1932. - pp. 356-366.
53. Растрингин Л.А. Системы экстремального управления // *Теоретические основы технической кибернетики*. - М.: Наука, 1974. - 632с.