

GLOBALIOJO OPTIMIZAVIMO UŽDAVINIŲ SPRENDIMAS NAUDOJANT GENETINIUS ALGORITMUS

Ervin Miloš, Dmitrij Šešok

Vilniaus Gedimino technikos universitetas

El. p.: ervin.milos@gmail.com, dmitrij.sesok@vgtu.lt

Įvadas

Žmonijai vystantis ir didėjant įmonių poreikiams, optimizavimo problemos vis sudėtingėja. Todėl optimizavimo tematika yra labai aktuali ir tokia išliks ateityje. Vienas iš svarbių optimizavimo uždavinių yra polių padėčių optimizavimas gręžtiniuose pamatuose. Priimant šį sprendimą reikia atlikti daug skaičiavimų, todėl taikomi stochastiniai optimizavimo algoritmai, kurie, nors ir negarantuoja globalaus sprendinio, bet pateikia racionalų sprendinį [6]. Genetiniai algoritmai gali būti taikomi iš esmės bet kokioms optimizuotinoms struktūroms. Tokie algoritmai yra pagrįsti evoliucinių procesų mus supančiame objektyviame pasaulyje imitavimu [7]. Šiame darbe analizuojamas polių padėčių optimizavimas rostverkinio tipo pamatuose; realizuojama programa su C++ kalba, pritaikyta spręsti globaliojo optimizavimo uždavinius, panaudojant genetinį algoritmą; atliekamas tyrimas apie genetinio algoritmo metodų panaudojimą, siekiant surasti optimalią strategiją, kurią pritaikius uždavinio rezultatas būtų kuo geresnis ir būtų pasiektas per apribotą iteracijų skaičių. Optimizavimo uždaviniu parinktas statybos uždavinys, turintis surasti optimalias atramų padėtis poliniuose pamatuose.

Tyrimo tikslas – pritaikyti genetinį algoritmą, siekiant polių padėčių optimizavimo gręžtiniuose pamatuose.

Tyrimo metodai:

1. Teoriniai: įvairios literatūros nagrinėjama tema analizė.
2. Skaitiniai sprendimo metodai.

Globaliojo optimizavimo uždavinių sprendimo teoriniai aspektai

Matematikoje ir informatikoje optimizavimas siejamas su geriausio pagal tam tikrą kriterijų elemento parinkimu iš galimos kandidatų aibės. Matematiškai optimizavimo uždavinys apibrėžiamas kaip funkcijos $f(\mathbf{x})$ minimalios reikšmės paieška:

$$f_{\min} = \min_{x \in D} f(x) \quad (1)$$

Čia funkcija $f : D \rightarrow R^d$ yra vadinama tikslo funkcija, $x = (x_1, x_2, x_3, \dots, x_d)$ – tikslo funkcijos kintamųjų vektoriumi, d – kintamųjų skaičiumi, o $D \subset R^d$ – parametrų reikšmių leistiną sritimi [5].

Genetinis algoritmas (GA) yra tam tikra evoliucinių algoritmų klasė, naudojanti gamtoje egzistuojančius gyvybės evoliucinius mechanizmus: paveldėjimą, mutaciją, natūraliąją atranką ir rekombinaciją [3]. Norint panaudoti genetinį algoritmą, kai ieškoma užduoties sprendimo, reikalingi du pagrindiniai elementai: tinkama pradinė duomenų struktūra ir atrankos kriterijus. Sprendžiant polių padėčių optimizavimo uždavinį, duomenų struktūrą atitinka sijyno polių padėtis, o atrankos kriterijus – polių padėčių įvertinimas, gaunamas naudojant gręžtinių pamatų skaičiavimo programą, parašytą FORTRAN kalba. Norint matematiškai įvertinti polių padėčių pozicijas gręžtiniuose pamatuose, naudojama tikslo funkcija, parodanti rezultato patikimumą. Kuo mažesnė tikslo funkcijos reikšmė, tuo geresnis yra sprendimas.

Gręžtinių pamatų skaičiavimo programos pritaikymas

Programa yra taikoma optimizavimo algoritams ir sukurta pagal „juodosios dėžės“ principą. Pagrindinis programos tikslas – sumažinti maksimalią polių reakcijos jėgą.

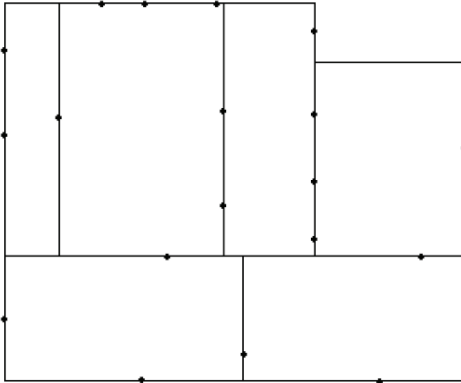
Optimizavimo uždavinys formuluojamas taip:

$$\min_{x \in D} P(x), \quad (2)$$

Programa įvertina polių padėtis naudodama tikslo funkciją, kuriai užtenka pateikti tik du parametrus:

- polių padėčių masyvą;
- bendrą sijyno geometrinės schemos ilgį.

Sijyno geometrine schema nusakoma konkrečiu atramų tipu, nustatytu įvairių skerspjūvių skaičiumi ir skirtingu sijų medžiagų planu (žr. 1 pav.) [6].



1 pav. Sijyno geometrinė schema

Kadangi ši programa naudojama pagal „juodosios dėžės“ principą, nereikia gilintis į pačią uždavinio sprendimo problematiką ir taikomas matematinės formules – užtenka pasinaudoti šios programos OBJECTIVE_FUNCTION funkcija ir pateikti atitinkamus parametrus.

Genetinis algoritmas gerai tinka norint panaudoti gręžtinių pamatų skaičiavimo programą, kadangi priklauso euristinio tipo algoritmams: sprendimas gaunamas bandymų ir klaidų keliu. Tai greitas ir mažiau sąnaudų reikalaujantis sprendimo būdas.

Vykdamas genetinį algoritmą, iš pradžių generuojama pradinė populiacija, kurią sudaro individai. Populiacijos vienas individas atitinka polių padėčių masyvą. Pagal atrankos kriterijų įvertinamas populiacijos individų tinkamumas. Atrinkami geriausi individai, iš kurių palikuonių bus sudaryta naujoji populiacija. Atliekama rekombinacija ir mutacija, siekiant suteikti populiacijai įvairovės. Po kiekvieno ciklo iteracijos kiekvieno individo polių padėtys pakartotinai įvertinamos ir algoritmas kartojamas tol, kol neviršija nustatyto iteracijų skaičiaus limitu.

Genetinis algoritmas realizuojamas C++ programavimo kalba, todėl gręžtinių pamatų skaičiavimo programa yra integruota į C++ aplinką. Integruotos skaičiavimo programos C++ aplinkoje vykdymo greitis buvo palygintas su tos pačios programos našumu FORTRAN aplinkoje. Testuojant kiekvienoje aplinkoje OBJECTIVE_FUNCTION funkcija buvo iškviesta 500 kartų. Rezultatai parodė, kad FORTRAN aplinkoje tikslo funkcijos apskaičiavimo laikas po 500 bandymų buvo lygus 5,765 sekundės, o C++ aplinkoje – 5,772 sekundės. Galima daryti išvadą, kad, integravus gręžtinių pamatų skaičiavimo programą į C++ valdančiąją programą, bendras programos našumas sumažėjo tik 0,008 sekundės.

Genetinio algoritmo pritaikymas ir išskirtinumas

Optimizavimo uždavinys sprendžiamas originaliu genetiniu algoritmu. Jo originalumas išsiskiria tuo, kad prie klasikinių genetinio algoritmo procesų – populiacijos generavimo, selekcijos, rekombinacijos ir mutacijos – pridedamas papildomas žingsnis – dinaminis mutacijos keitiklis.

Kai vykdamas genetinį algoritmą iteracijų skaičius tampa labai didelis, visų individų tikslo funkcijos reikšmės vienu momentu pasidaro vienodos: tai reiškia, kad populiacija nustoja tobulėti ir įstringa lokalo minimumo taške – toks reiškinys vadinamas ankstyvąja konvergencija. Konvergencija yra tada, kai 95 proc. individų populiacijos yra vienodi – turi tokią pačią tikslo funkcijos reikšmę.

Tam, kad būtų išvengta lokalo minimumo, taikomas dinaminis mutacijos keitiklis, kuris suveikia tada, kai aptinkama populiacijos konvergencija. Dėl keitiklio visi parametrai, susieti su mutacijos procesu, pavyzdžiui, mutacijos tikimybė, individo mutacijos tikimybė ir vieno geno mutacijos tikimybė, yra dinamiški. Tai suteikia populiacijai įvairovės, pakeičia individų genų reikšmes atsitiktiniais skaičiais. Tokiu atveju individų tikslo funkcijos reikšmės gali pablogėti, tačiau, pašalinus konvergenciją, individai dar labiau pradeda gerėti.

Skaitiniams eksperimentams realizuoti paruošti genetinio algoritmo metodai pateikti 1 lentelėje:

1 lentelė. *Genetinio algoritmo realizuoti metodai*

Populiacijos generavimas	Selekcija	Rekombinacija	Mutacija	Mutacijos keitiklis
Klasikinis generatorius	Turnyro selekcija	Vieno taško rekombinacija	Vieno taško mutacija	Vieno taško keitiklis
Modifikuotas generatorius	Rangų selekcija	Dviejų taškų rekombinacija	Uniforminė mutacija	Uniforminis keitiklis
		Uniforminė rekombinacija	Modifikuota mutacija	Modifikuotas keitiklis

Skaitiniai eksperimentai ir optimalios genetinio algoritmo strategijos paieška

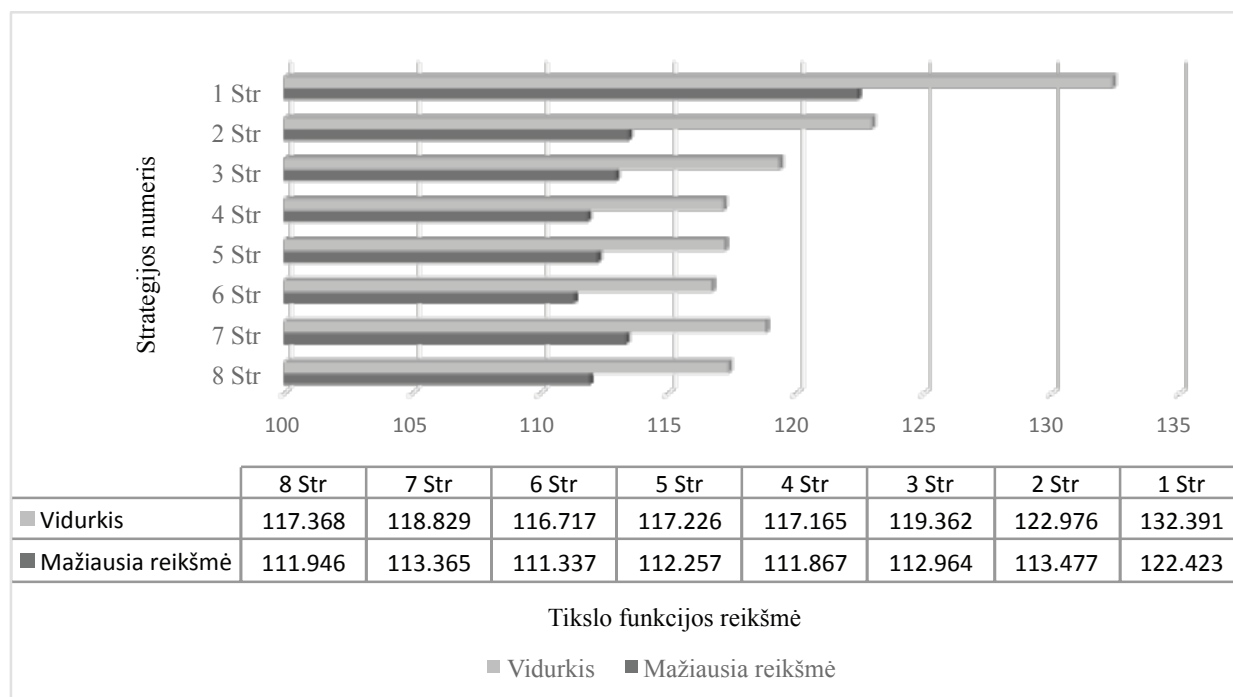
Skaitiniams eksperimentams yra nustatomos tam tikros sąlygos, t. y. pasirinkti genetinio algoritmo realizuoti metodai ir aprašyti pagrindiniai nustatymai, kuriems esant programa daro tiesioginę įtaką tikslo funkcijai. Šių eksperimentų tikslas – surasti optimalią optimizavimo programos strategiją, kurios polių padėčių gręžtiniuose pamatuose uždavinio sprendimas bus arčiausiai optimaliojo.

Išbandomos aštuonios strategijos, naudojant įvairias genetinio algoritmo metodų kombinacijas. Toks skaičius yra parinktas tam, kad galima būtų iš-

bandyti visus realizuotus genetinio algoritmo metodus, pateiktus 1 lentelėje. Kiekviena strategija kartojama 28 kartus, kadangi genetinio algoritmo rezultatas yra nenuspėjamas, t. y. vienu bandymo atveju rezultatas gali būti geras, kitu atveju blogas. Sudaroma sąlyga, kad sprendimas turi būti pateikiamas įvykdžius ne daugiau kaip 5 000 iteracijų.

Siekiamas rezultatas – tikslo funkcijos reikšmė, lygi 104,12.

Buvo atlikta bandymų keičiant selekcijos, rekombinacijos ir mutacijos metodus, kaitaliojant populiacijos ir iteracijų dydžius. Visų atliktų strategijų eksperimentų bendri rezultatai pavaizduoti 2 paveiksle.



2 pav. Visų atliktų strategijų eksperimentų rezultatai

Visų atliktų strategijų eksperimentų rezultatų diagrama parodė, kad geriausi rezultatai, sprendžiant polių padėčių optimizavimo gręžtiniuose pamatuose uždavinį, buvo gauti pritaikius **šeštąją strategiją**.

Bendri algoritmo nustatymai:

- populiacijos dydis – 30;
- iteracijų skaičius – 150.

Šios strategijos taikomi metodai:

- modifikuotas generatorius, suteikiantis pradinei populiacijai žymiai geresnių individų, panaudojus tokius parametrus:
 - pradinis atsitiktinis generavimo skaičius – 500; tai reiškia, kad bus sugeneruota tiek individų, o geriausi 30 pateks į pradinę populiaciją;
 - populiacija rūšiuojama didėjančia tvarka;
 - taikomas diskretusis išdėstymas;
- rangų selekcija – metodas, išvystytas iš ruletės išrinkimo metodo. Taikant ruletės rato meto-

da, kiekvienas individas yra renkamas tiesiogiai pagal ruletės tikslo funkcijos reikšmę – kuo didesnis vertinimas, tuo didesnė tikimybė. Ruletės metodas yra skirtas labiau uždaviniui padidinti nei sumažinti, be to, gali atsirasti problemų, jei kelių individų tikslo funkcijos reikšmė sudarys didelę dalį pačios ruletės. Taigi, yra labai mažai tikimybės, kad silpniausi individai išgyvens. Šią problemą sprendžia būtent rangų selekcija. Jos metu kiekvienam individui nustatomas vertinimo rangas nuo 1 iki populiacijos dydžio, pradedant nuo blogiausio individo, turinčio didžiausią tikslo funkcijos reikšmę, iki geriausio. Taikant šį metodą, kiekvienas individas turi panašią galimybių patekti į kitą populiaciją, todėl yra tik maža tikimybė, kad individai vienu metu taps vienodi;

- konkurencinis elitai – atrinkti individai dalyvauja visame genetinio algoritmo procese. Jie kryžminami su kitais išrinktais individais, gali būti

vykdoma jų genų mutacija. Tokio tipo strategija leidžia suteikti populiacijai įvairovės, nes, palyginus su nekonkurenciniu metodu, išsaugotų individų reikšmės niekada nepasikeis, nebent atsiras geresnis individas su mažesne tikslo funkcijos reikšme, o šiuo atveju geriausias individas tik iš dalies praranda savo buvusias reikšmes, kas padidina tikimybę pagerinti šio individo vertinimą;

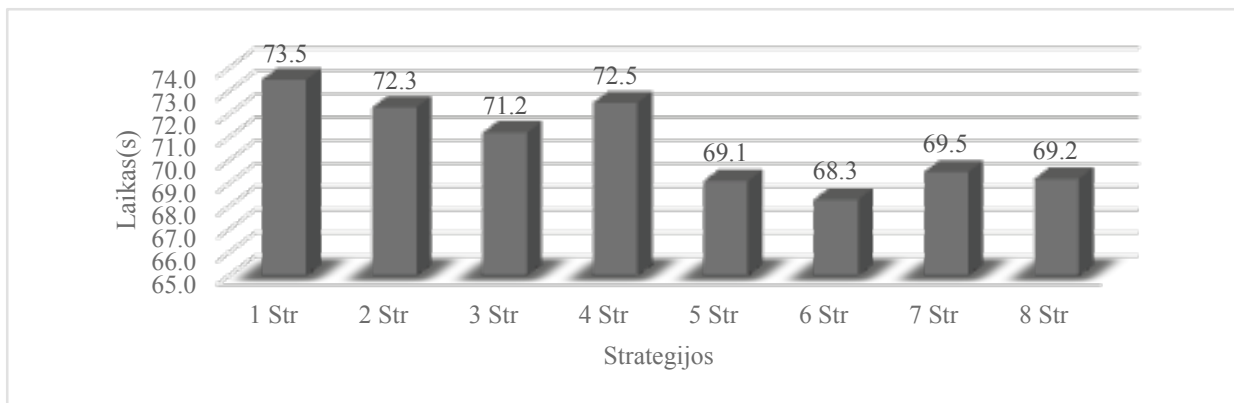
- uniforminė rekombinacija turi daug susikirtimo taškų. Tikslas – sukryžminti abu „tėvus“, perpus nustatant susikirtimo koeficientą 50 proc. (žr. 3 lentelę). Šio tipo algoritmas veikia tokiu būdu: pradedamas ciklas nuo pirmojo „tėvo“ elemento iki paskutinio ir sugeneruojamas atsitiktinis sveikasis skaičius nuo 0 iki 1. Jeigu jo reikšmė yra mažesnė nei 0,5, tada elementas patenka į antrąjį vaiką, kitu atveju – į pirmąjį. Taigi, su tikimybe 0,5 vaikai gauna maždaug 50 proc. iš pirmojo ir 50 proc. iš antrojo tėvo;
- modifikuota mutacija – sprendžiant šį uždavinį buvo naudojami tokie mutacijos parametrai:
 - geno atstumas – 1, kas reiškia, kad, atliekant geno mutaciją, jo nauja reikšmė negali viršyti arba būti mažesnė už prieš ją esančio geno ir po jos esančio geno reikšmę;
 - mutacijos tikimybė – 0,2, o tai reiškia, kad yra tikimybė individui patekti į mutacijos procesą;
 - vieno geno mutacijos tikimybė – 0,1, kas reiškia, kad yra tikimybė vienam genui patekti į mutacijos procesą;

- modifikuotas mutacijos keitiklis, didinantis mutacijos tikimybę ir vieno geno mutacijos tikimybę, jeigu pasirinktasis populiacijos individo indeksas skiriasi nuo pirmojo mažiau nei 0,01. Garantuojama, kad į šį procesą populiacija patenka visuomet išrūšiuota didėjančia tvarka.

Geriausia šio eksperimento ir apskritai visų atliktų tyrimų pasiekta tikslo funkcijos reikšmė – **111,387**.

Šių strategijų rezultatai atskleidė, kad geriausias sprendimas, nagrinėjant polių padėčių optimizavimo gręžtiniuose pamatuose uždavinį, yra atsisakyti klasikinių mutacijos metodų, pavyzdžiui, vieno taško mutacijos, uniforminės mutacijos, ir taikyti modifikuotą mutaciją kartu su ją atitinkančiu mutacijos keitikliu. Naudojant klasikinius mutacijos metodus nenustatoma kriterijų ir reikšmių, kurias po mutacijos turėtų gauti individo genai, todėl toks būdas gali pabloginti individo įvertinimą. Taikant modifikuotą mutaciją, genų reikšmės keičiasi priklausomai nuo pateikto atstumo: tokiu atveju populiacijos individų įvertinimas nuolat gerėja. Modifikuojant mutacijos keitikliu užtikrinama, kad populiacija išvengs konvergencijos.

Buvo matuojama kiekvienos strategijos vidutinė vienos strategijos vykdymo trukmė. Tai buvo daroma tam, kad galima būtų patikrinti bendrą algoritmo našumą, pritaikius visas prieš tai išvardytas strategijas. Vidutinio strategijos vykdymo greičio statistika pateikta 3 paveiksle.



3 pav. Vidutinio strategijos vykdymo greičio statistika

Kiekvieno eksperimento strategijos algoritmo greitis buvo panašus. Bendras visų pateiktų greičių vidurkis – **70,7** sekundės.

Atlikto tyrimo rezultatai atskleidė, kiek maždaug laiko užima kiekvieno genetinio algoritmo žingsnio vienai strategijos iteracijai įvykdymas: selekcijos trukmė – 0,12 sekundės, rekombinacijos trukmė – 0,013 sekundės, mutacijos trukmė – 0,0005 sekundės, mutacijos keitiklio trukmė – 0,001 sekun-

dės, vidutinis tikslo funkcijos apskaičiavimas vienam individui – 0,012 sekundės.

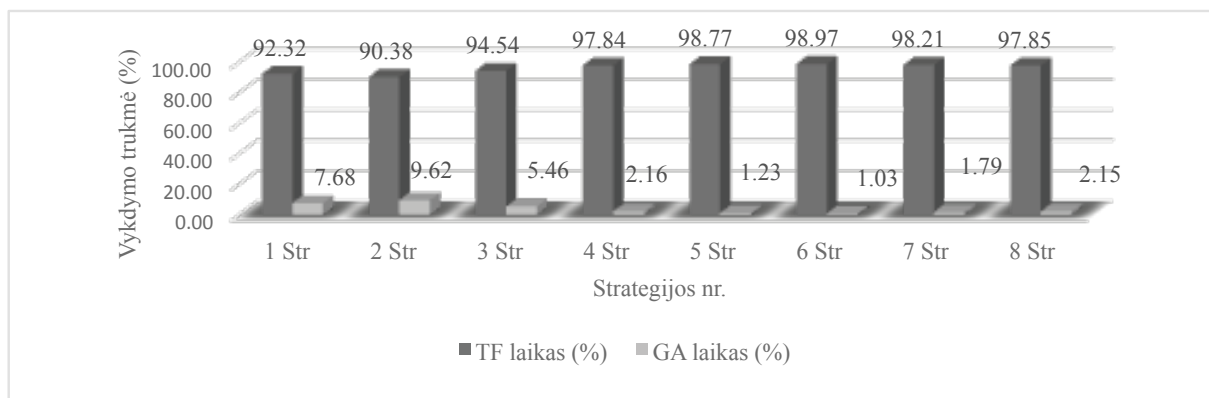
Buvo matuojamas kiekvienos strategijos, vykdančios kiekvienos iteracijos programą, genetinio algoritmo ir tikslo funkcijos apskaičiavimo visai populiacijai vidutinis laikas. Vienos iteracijos genetinio algoritmo (GA) vykdymo ir tikslo funkcijos (TF) apskaičiavimo laikas pateiktas 2 lentelėje.

2 lentelė. Vienos iteracijos genetinio algoritmo vykdymo ir tikslo funkcijos apskaičiavimo laikas

Strategijos nr.	GA apskaičiavimo laikas (s)	TF apskaičiavimo laikas (s)
1	0,116	1,394
2	0,135	1,269
3	0,073	1,265
4	0,025	1,132
5	0,014	1,123
6	0,009	0,862
7	0,018	0,989
8	0,022	1,002

Vidutinis visos populiacijos tikslo funkcijos apskaičiavimo laikas – 1,130 sekundės, o vidutinis visos populiacijos genetinio algoritmo apskaičiavimo

laikas – 0,052 sekundės. Grafiškas šių dviejų procesų procentinis palyginimas pavaizduotas 4 paveiksle.



4 pav. Genetinio algoritmo ir tikslo funkcijos apskaičiavimo laiko palyginimas procentais

Remiantis diagramoje pateiktais rezultatais galima teigti, kad pagrindiniai optimizavimo programos skaičiavimo resursai naudojami tikslo funkcijai apskaičiuoti. Genetinio algoritmo strategijos pakeitimas praktiškai nedaro įtakos skaičiavimo laikui. Vidutiniškai vienos iteracijos genetinio algoritmo vykdymas užima **3,89 proc.**, o tikslo funkcijos apskaičiavimas – **96,11 proc.**

Tyrimo rezultatų palyginimai

Polių padėčių gręžtiniuose pamatuose uždavinio sprendimo tyrimo rezultatai, naudojant genetinį algoritmą, palyginti su mokslininkų J. Mockaus, R. Belevičiaus, D. Šešoko, J. Kauno, D. Mačiūno 2012 metais atlikto tyrimo rezultatais, kurie, sprendami tą patį uždavinį, naudojo Bajeso metodą, ir su mokslininkų R. Belevičiaus, S. Ivanikovo, D. Šešoko, S. Valentinavičius ir S. Žilinsko 2011 metais atlikto tyrimo rezultatais, kurie lygino įvairius globaliojo optimizavimo metodus, sprendami polių padėčių gręžtiniuose pamatuose optimizavimo uždavinį.

Genetinio algoritmo ir Bajeso metodo palyginimas

Mokslininkai J. Mockus ir kt., sprenddami polių padėčių gręžtiniuose pamatuose uždavinį Bajeso optimizavimo metodu [8], rėmėsi tokiomis taisyklėmis: tikslo funkcija turi būti pasirenkama ne daugiau kaip 5 000 kartų ir kiekvienas eksperimentas turi būti atliekamas 28 kartus. Mokslininkų pasiektas geriausias rezultatas – **113,47**. Lyginant rezultatus galima daryti išvadą, kad, sprendžiant polių padėčių gręžtiniuose pamatuose optimizavimo uždavinį naudojantis genetiniu algoritmu, gaunami 1,9 proc. geresni rezultatai.

Genetinio algoritmo palyginimas su kitais globaliojo optimizavimo metodais

Mokslininkai R. Belevičius ir kt., sprenddami polių padėčių rostverkinio tipo pamatuose optimizavimo uždavinį, lygino šiuos globaliojo optimizavimo metodus:

- atsitiktinės paieškos (angl. *Random Search*, toliau – RS);
- modifikuotos atsitiktinės paieškos (angl. *Modified random search*, toliau – MRS);

- atkaitinimo modeliavimo (angl. *Simulated annealing*, toliau – SA);
- genetinį algoritimą (angl. *Genetic algorithm*, toliau – GA);
- „Simplex“ (toliau – SM);
- „Variable metric“ (toliau – VM);
- NEWUOA algoritimą.

Mokslininkai R. Belevičius ir kt., sprenddami polių padėčių gręžtiniuose pamatuose uždavinį išvardytais globaliojo optimizavimo metodais, rėmėsi tokiais taisyklėmis: tikslo funkcija turi būti pasirenkama ne daugiau kaip 5 000 kartų ir kiekvienas eksperimentas turi būti atliekamas 28 kartus [1]. Mokslininkų R. Belevičiaus ir kt. atlikto tyrimo rezultatai pateikti 3 lentelėje.

3 lentelė. *Geriausi uždavinio sprendimai po 28 eksperimentų*

Nr.	R_{ideal}	MRS	SA	GA	SM	VM	NEWUOA	BACoor	Realizuotas GA
2	104,12	119,16	106,36	106,52	125,83	123,18	107,18	113,47	111,33

Iš 3 lentelėje pateiktų rezultatų matyti, kad realizuotas algoritmas pagal tikslo funkcijos reikšmės optimalumą pranoko:

- modifikuotos atsitiktinės paieškos metodą – 6,6 proc.;
- „Simplex“ metodą – 11,6 proc.;
- „Variable metric“ metodą – 10,1 proc.

Tačiau realizuotas genetinis algoritmas silpnėnis už:

- NEWUOA metodą – 3,8 proc.;
- SA metodą – 4,6 proc.;
- genetinio algoritmo metodą – 4,5 proc.

Išvados

Pritaikius genetinį algoritimą polių padėčių optimizavimo uždaviniui, galima daryti išvadą, jog tai yra perspektyvus algoritmas, sprendžiant polių padėčių optimizavimo gręžtiniuose pamatuose uždavinį, kadangi jo realizavimas yra paprastas, praktiškai nedaro įtakos skaičiavimo resursams. Vidutiniškai vienos iteracijos genetinio algoritmo vykdymas užima 3,89 proc., o tikslo funkcijos apskaičiavimas – 96,11 proc. Geriausias pasiūlymas, sprendžiant uždavinį genetiniu algoritmu, – taikyti modifikuotus mutacijos ir mutacijos keitiklių metodus. Dėl pasiūlyto genetinio algoritmo pavyko gauti 1,9 proc. geresnį rezultatą nei taikant Bajeso metodą, tačiau iki geriausio literatūroje aprašomo rezultato, kai sprendžiant globaliojo optimizavimo uždavinį buvo taikomas atkaitinimo modeliavimo metodas, trūko 4,6 proc.

Literatūra

1. Belevičius R., Ivanikovas S., Šešok D., Valentinavičius S., Žilinskas J., 2011, *Optimal placement of piles in*

real gillages: experimental comparison of optimization algorithms. Prieiga per internetą: <https://www.researchgate.net/publication/229039634_Optimal_placement_of_piles_in_real_grillages_Experimental_comparison_of_optimization_algorithms/> [žiūrėta 2017-04-22].

2. Belevičius R., Mačiūnas D., Šešok D., 2011, *Momentų ir reakcijų minimizavimas rostverkiniuose pamatų sijynuose genetiniu algoritmu*. Prieiga per internetą: <http://dSPACE.vgtu.lt/bitstream/1/818/1/EST_Vol3_No2_56-63_Belevicius.pdf/> [žiūrėta 2017-02-28].
3. *Genetinis algoritmas*, 2016. Prieiga per internetą: <https://lt.wikipedia.org/wiki/Genetinis_algoritmas> [žiūrėta 2016-12-03].
4. *Gręžtinių pamatų įrengimas*, 2017. Prieiga per internetą: <https://www.ekspertai.lt/greztiniai_pamatai/straipsniai/greztiniu_pamatu_irengimas> [žiūrėta 2017-04-01].
5. Lančinskas A., 2013, *Atsitiktinės paieškos globaliojo optimizavimo algoritmu lygiagretinimas*. Prieiga per internetą: <http://www.mii.lt/files/mii_dis_2013_lancinskas.pdf> [žiūrėta 2016-12-03].
6. Mačiūnas D., Belevičius R., Kaunas J., 2011, *Daugiakriteris sijynų optimizavimas genetiniais algoritmais*. Prieiga per internetą: <<http://www.mla.vgtu.lt/index.php/mla/article/viewFile/mla.2011.110/pdf/>> [žiūrėta 2017-03-26].
7. Misevičius A., Blonskis J., Blažinskas A., Bukšnaitis V., 2009, *Genetiniai algoritmai komivojažieriaus uždaviniui: negatyvieji ir pozityvieji aspektai*. Prieiga per internetą: <<http://www.zurnalai.vu.lt/informacijos-mokslai/article/viewFile/3242/2359/>> [žiūrėta 2017-01-10].
8. Mockus J., Belevičius R., Šešok D., Kaunas J., Mačiūnas D., 2012, *On Bayesian Approach to Grillage Optimization*. Prieiga per internetą: <<http://itc.ktu.lt/index.php/ITC/article/view/1670/2033/>> [žiūrėta 2017-04-22].

Summary**GENETIC ALGORITHMS FOR SOLVING GLOBAL OPTIMIZATION PROBLEMS***E. Miloš, D. Šešok*

The authors examine the theoretical aspects of solving global optimization problems and analyse how global optimisation can be used in bored pile foundations. The position of bored pile foundations was determined with FORTRAN. When the C++ code was optimized the overall performance of the program decreased only by 0,008s. Optimization problems were solved with genetic algorithms, the time taken to execute optimization and genetic algorithms was compared. It was found that genetic algorithms have no impact on computing resources. Eight strategies for using various combinations of genetic algorithms were tested in order to identify the most effective one, the findings were compared with the findings of other scientists. The result when a global optimization problem was solved with the proposed genetic algorithm was by 1.9% better than that using the Bayesian method but by 4.6% worse than using a simulated annealing method described in literature as the best one.

Keywords: genetic algorithm (GA), global optimization, bored pile.

Santrauka**GLOBALIOJO OPTIMIZAVIMO UŽDAVINIŲ SPRENDIMAS NAUDOJANT
GENETINIUS ALGORITMUS***E. Miloš, D. Šešok*

Straipsnyje nagrinėjami globaliojo optimizavimo uždavinių sprendimo teoriniai aspektai. Analizuojamas polių padėčių optimizavimo gręžtiniuose pamatuose uždavinys. Polių padėčių įvertinimas apskaičiuojamas naudojant integruotą gręžtinių pamatų skaičiavimo programą, realizuojamą FORTRAN programavimo kalba. Po integravimo į C++ aplinką bendras programos našumas sumažėjo tik 0,008 sekundės. Panaudojus genetinį algoritmą buvo realizuota optimizavimo programa, palygintas optimizavimo programos tikslo funkcijos ir genetinio algoritmo vykdymo laikas. Nustatyta, kad genetinis algoritmas praktiškai nedaro įtakos skaičiavimo resursams. Išbandytos aštuonios strategijos, pritaikius įvairias genetinio algoritmo metodų kombinacijas atliekant optimalios strategijos paiešką. Tyrimo rezultatai palyginti su kitų mokslininkų atliktais tyrimo rezultatais. Dėl pasiūlyto genetinio algoritmo pavyko gauti 1,9 proc. geresnį rezultatą nei taikant Bajeso metodą, tačiau iki geriausio literatūroje aprašyto rezultato, kai sprendžiant globaliojo optimizavimo uždavinį buvo taikomas atkaitinimo modeliavimo metodas, trūko 4,6 proc.

Prasminiai žodžiai: genetinis algoritmas, globalusis optimizavimas, rostverkas.

Įteikta 2017-06-02
Priimta 2017-06-23