



GLOBALUSIS ROSTVERKINIŲ PAMATŲ OPTIMIZAVIMAS HIBRIDINIŲ GENETINIŲ ALGORITMU

Dmitrij Šešok¹, Rimantas Belevičius²

¹Matematikos ir informatikos institutas, Akademijos g. 4, LT-08663 Vilnius, Lietuva

^{1,2}Vilniaus Gedimino technikos universitetas, Saulėtekio al. 11, LT-10223 Vilnius, Lietuva

El. paštas: ¹dms@fm.vgtu.lt; ²rb@fm.vgtu.lt

Įteikta 2009 04 27; priimta 2009 06 01

Santrauka. Straipsnio tikslas – sukurti technologiją polių padėtims rostverkiniuose pamatuose optimizuoti. Siekiama minimizuoti polių skaičių rostverke. Tikslų funkcija imama didžiausia atraminė poliaus reakcija, kuri optimizavimo metu yra minimizuojama. Iš pradžių uždavinys sprendžiamas genetiniu algoritmu, o kai toliau genetinį algoritmą naudoti tampa neefektyvu, gautas rezultatas gerinamas specifiniu atsitiktinės lokalių paieškos algoritmu. Analizuodami eksperimentinius duomenis autoriai bando nustatyti momentą, kada reikia pereiti nuo genetinio algoritmo naudojimo prie lokalių paieškos algoritmo, kad geresnis rezultatas būtų pasiekiamas per kuo trumpesnę laiką. Bendrą algoritmą, sujungiantį genetinį algoritmą su lokalių paieškos algoritmu, autoriai vadina hibridiniu genetiniu algoritmu.

Reikšminiai žodžiai: globalusis optimizavimas, genetiniai algoritmai, poliniai pamatai.

1. Įvadas

Optimizavimas yra neatsiejama inžinerinės praktikos dalis. Šiame straipsnyje bus nagrinėjama viena specifinė statybos inžinerijos optimizavimo uždavinių klasė – polių padėčių optimizavimas rostverkinių tipo pamatuose. Sijynai, kuriuos sudaro poliai ir jungiamosios sijos, yra populiariausios ir efektyviausios pamatų schemos, ypač kai statybos vyksta ant silpnų gruntų. Optimalus yra toks sijynas, kuris atitinka du kriterijus: yra palaikomas mažiausiu galimu polių skaičiumi ir patiria minimalius lenkimo momentus polių jungiančiose sijose. Šioje formuluotėje laikoma, kad polių ir jungiamųjų sijų charakteristikos yra žinomos prieš optimizuojant. Susiduriame su dviem skirtingais optimizavimo uždaviniais: sijyno optimizavimu, siekiant minimizuoti polių skaičių, ir jungiamųjų sijų optimizavimu, siekiant sumažinti jų tūrį. Abu šie uždaviniai gali būti sujungti į vieną naudojant kompromisinę tikslo funkciją. Šios problemos (tik naudojant lokalius optimizavimo metodus) buvo spręstos anksčiau (Belevičius, Valentinavičius 2000; Belevičius *et al.* 2002; Belevičius, Valentinavičius

2001). Šiame straipsnyje nagrinėsime pirmojo optimizavimo uždavinio sprendimą stochastiniais globaliojo optimizavimo metodais. Antrosios problemos matematiniai modeliai yra panašūs.

Optimali polių išdėstymo topologija gali būti gauta keičiant atitinkamas polių padėtis po jungiamosiomis sijomis. Idealiame sijyne visi poliai turės vienodas reakcijų jėgas, artimas leistinajai poliaus apkrovai. Praktiškai tai sunkiai pasiekama, ypač kai projektuotojas, atsižvelgdamas į tam tikras aplinkybes, sijyno schemoje naudoja vadinamąsias „nejudamas atramas“, kurios negali keisti savo pozicijų ir todėl nedalyvauja optimizavimo procese. Kai kurios techninės problemos, pavyzdžiui, minimalus galimas atstumas tarp gretutinių polių (dėl poliakalės darbo specifikos) irgi gali kliudyti surasti idealią schemą.

Pradiniai sijyno optimizavimo duomenys yra šie:

- geometrinė jungiamųjų sijų schema;
- sijų skerspjūvių duomenys (plotas, inercijos momentai);
- sijų medžiagos duomenys (vienos sijos medžiaga laikoma izotropine);

- nejudamų atramų pozicijos (jeigu tokių yra);
- didžiausia leistinoji reakcijos jėga, galinti kilti bet kuriame iš polių;
- mažiausias galimas atstumas tarp gretimų polių;
- poliaus vertikalusis standis ir du lenkiamieji standžiai: išilgai sijos, po kuria yra polis, ir statmena kryptimi;
- apkrovų duomenys. Aktyviosios jėgos gali būti pateiktos sutelktųjų apkrovų ir lenkimo momentų bet kuriame sijos taške pavidalu arba išskirstytų trapecinių apkrovų bet kurioje sijos dalyje pavidalu.

Sprendimo rezultatas – reikalingas polių skaičius ir polių padėtys.

Visi pradiniai duomenys projektuotojui yra žinomi arba gaunami. Išimtį sudaro tik polių standumai. Projektuotojas gali būti linkęs imti begalinius standumus, o tai reikštų nulinius atramų poslinkius polių vietose, t. y. nulines kraštines sąlygas matematiname modelyje. Tačiau tai nėra tinkamas ir aiškiai per stiprus idealizavimas, kas ir buvo parodyta keliais skaitiniais pavyzdžiais (Belevičius, Valentinavičius 2001). Pavyzdžiui, tuo atveju, kai didelei sutelktai apkrovai perimti reikia kelių atramų, šis idealizavimas veda į aklavietę. Nelyginio atramų skaičiaus atveju centrinė atrama bus kaip tik po apkrova ir perims visą jėgą. Kai yra lyginis atramų skaičius, pastebimas reakcijų pasiskirstymas pagal „pjūklo dantų“ dėsnį, ir kuo daugiau bus polių, tuo didesnės absoliutiniu dydžiu bus reakcijos. Kai tipiniai vertikalūs polių nuosėdžiai yra žinomi, akivaizdi procedūra apytiksliai standumams gauti – nustatyti vertikalų standumą iš vienmatės spyruoklės deformavimo uždavinio.

2. Matematinė formuluotė

Matematinio požiūriu šis uždavinys yra patrauklus, nes iš anksto žinoma globalųjį sprendinį atitinkanti tikslo funkcijos vertė: tam tereikia aktyviųjų jėgų sumą padalyti iš programos nustatyto polių skaičiaus. Tad galima įvertinti gautojo sprendinio kokybę. Optimizavimo uždavinys formuluojamas taip:

$$\min_{x \in D} P(x), \quad (1)$$

čia P – didžiausia reakcijos jėga bet kuriame iš polių. Galima struktūros konfigūracija D nusakoma sijyno geometrine schema, konkrečių atramų tipu, duotu įvairių skerspjūvių skaičiumi bei skirtingų sijų medžiagų planu. x žymi projektavimo kintamuosius.

Tam, kad rastųsi galimybė skirtingoms atramoms įgyti skirtingas reakcijas (tuo atveju, kai poliai turi skir-

tingas charakteristikas), vietoje didžiausios reakcijos imamas didžiausias skirtumas tarp vertikalios reakcijos jėgos atramoje ir leistinosios reakcijos tai atramai:

$$P(x) = \max_{1 \leq i \leq N_a} |R_i - f_i R_l|, \quad (2)$$

čia: N_a – atramų skaičius. R_l žymi leistinąją reakciją, o R_i yra reakcijos jėga i -tojoje atramoje. Duotieji dauginčiai f_i leidžia naudoti skirtingas leistinąsias reakcijas skirtingose atramose.

Optimizavimo proceso metu poliai gali judėti tik išilgai jungiamųjų sijyno sijų. Šis apribojimas realizuojamas tokiu būdu: dvimatis sijynas yra „išvyniojamas“ į vienmatę tiesę, išilgai kurios leidžiama laisvai judėti atramoms. Optimizavimo programa teikia atramų pasiskirstymo schemą šioje vienmatėje erdvėje, o atgalinė transformacija vėliau grąžina atramų pozicijas į realų sijyną. Baigtinių elementų programa, skirta tiesioginiam didžiausios reakcijos jėgos sijyne uždaviniui spręsti, yra jungiama prie optimizavimo programos kaip „juodoji dėžė“. Priklausomai nuo „juodosios dėžės“ atsako, optimizavimo programa numato kitą polių pasiskirstymą.

Tokioje formuluotėje vienas projektavimo kintamasis atitinka vieną atramos koordinatę vienmatėje struktūroje. Projektavimo kintamųjų apribojimus galima užrašyti taip:

$$0 \leq x_i \leq L, \quad i = 1, 2, \dots, N_a, \quad (3)$$

čia: x_i – i -tosios atramos koordinatė vienmatėje struktūroje; L – bendras visų sijyno sijų ilgis.

Tiesioginiame uždavinyje, sprendžiamame baigtinių elementų metodu, jungiamosios sijos sijyne yra idealizuojamos kaip lenkiamųjų strypų elementai, o poliai traktuojami kaip atramos, t. y. baigtinių elementų tinklo mazgai su duotomis kraštinėmis sąlygomis. Kadangi ribojamasis visų globaliojo optimizavimo uždavinių veiksnys yra skaičiavimo laikas, čia taikomos originalios, uždavinio specifiką įvertinančios ir todėl greitos FORTRANO programos su specialiais baigtinių elementų tinklo generavimo priešprocesoriais. Kita išėitis būtų komercinių programų (ANSYS, ALGOR, ABAQUS, COSMOS ir kt.) naudojimas, bet autorių (Šešok, Belevičius 2007) ir kitų tyrėjų (Puiša 2005; Ragauskas, Skukis 2007) patirtis rodo, kad šiuo atveju daug laiko prarandama kviečiant išorinę programą iš valdančiosios programos.

Lenkiamųjų strypų elementai turi du mazgus su šešiais laisvės laipsniais kiekviename mazge. Šio ele-

mento standumo matrica (Spyrakos, Raftoyiannis 1997; Barauskas *et al.* 2004) pateikta toliau:

$$[K] = \begin{bmatrix} [K_{11}] & [K_{12}] \\ [K_{12}]^T & [K_{22}] \end{bmatrix}, \quad (4)$$

$$[K_{11}] = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12 EI_z}{L^3} & & & & \frac{6 EI_z}{L^2} \\ 0 & & \frac{12 EI_y}{L^3} & & & -\frac{6 EI_y}{L^2} \\ 0 & & & \frac{GJ}{L} & & \\ 0 & & -\frac{6 EI_y}{L^2} & & \frac{4 EI_y}{L} & \\ 0 & \frac{6 EI_z}{L^2} & & & & \frac{4 EI_z}{L} \end{bmatrix}, \quad (5)$$

$$[K_{12}] = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{12 EI_z}{L^3} & & & & \frac{6 EI_z}{L^2} \\ 0 & & -\frac{12 EI_y}{L^3} & & & -\frac{6 EI_y}{L^2} \\ 0 & & & \frac{GJ}{L} & & \\ 0 & & \frac{6 EI_y}{L^2} & & \frac{2 EI_y}{L} & \\ 0 & -\frac{6 EI_z}{L^2} & & & & \frac{2 EI_z}{L} \end{bmatrix}, \quad (6)$$

čia: E – tamprumo modulis; A – sijos skerspjūvio plotas; J , I_z , I_y – skerspjūvio inercijos momentai. Submatrica $[K_{22}]$ sutampa su $[K_{11}]$, bet nediagonaliniai nariai yra priešingų ženklų.

Pagrindinė statikos lygtis yra

$$[K]^a \{u\}^a = \{F\}^a, \quad (7)$$

čia indeksas a žymi baigtinių elementų ansamblį (formulėse toliau neberodomas), $\{u\}$ yra mazgų poslinkiai, o $\{F\}$ – aktyviosios jėgos. Polių reakcijos jėgos, radus mazgų poslinkius, apskaičiuojamos taip:

$$R_i = \sum_j [K_{ij}] u_j. \quad (8)$$

3. Optimizavimas taikant GA

Keli nagrinėjamo optimizavimo uždavinio sprendimo pavyzdžiai, taikant pirmiau aprašytą matematinę modelį ir deterministinius globalaus optimizavimo metodus, publikuoti Žilinsko, Čiegio ir kt. darbuose (Ži-

linskas 2001, 2002; Čiegis *et al.* 2004). Pagrindinis šios sprendimo strategijos trūkumas susijęs su per dideliu, inžinerinei praktikai nepriimtiniu sprendimo laiku. Pavyzdžiui, dešimties projektavimo kintamųjų uždavinys panaudojus Vilniaus Gedimino technikos universiteto klasterį su 10 dualių procesorių, sprendžiamas apie 60 min., o penkiolikos projektavimo kintamųjų uždaviniui spręsti reikia jau penkių valandų. Vėliau šių uždavinių sprendimas, taikant genetinius algoritmus, aiškiai parodė, kad tikslo funkcijos pavaldas yra labai sudėtingas, su daugeliu lokalių minimumų. Globalios paieškos algoritmas, kaip ir tikimasi (Srinivas, Patnaik 1994), didžiąją skaičiavimų laiko dalį skiria būtent poaibiams aplink šiuos taškus nagrinėti. Galima daryti prielaidą, kad stochastiniai (tikimybiniai) globalaus optimizavimo metodai, tarp jų ir genetiniai algoritmai (GA), modeliuojantys gamtos evoliucijos dėsnius (Goldberg 1989; Holland 1975), gali būti pranašesni dideliems šios klasės uždaviniams. Aišku, GA nėra idealūs algoritmai visiems optimizavimo uždaviniams ir turi savo trūkumą (Tazawa *et al.* 1996; Kennedy 1993). Praktika rodo, kad GA negali garantuoti globalaus sprendinio radimo per fiksuotą iteracijų skaičių. Be to, dažniausia GA teikia tik apytikslį uždavinio sprendinį. Tačiau šis trūkumas nėra labai svarbus praktiniams uždaviniams, kuriuose teorinis globalusis sprendinys dažniausiai nėra būtinas. Kai kurie pradiniai rezultatai (Belevičius, Šešok 2008) rodo, kad naudojant GA įmanoma rasti gana gerą sprendinį per priimtina laiką.

Sprendžiant nagrinėjamus optimizavimo uždavinius, GA paprastai būdingas labai greitas konvergavimas pradiniam sprendimo etape ir lėtas nemonotinis konvergavimas į lokalų sprendinį po tam tikro generacijų skaičiaus. GA yra stochastiniai algoritmai, todėl uždavinį skaičiuoti būtina kelis kartus. Būtina prie uždavinio priderinti genetinius algoritmo parametrus: generacijų skaičių, kryžminimo ir mutacijos tikimybes; apibrėžti genetinius operatorius. Dažnai tokie atsitiktiniai skaičiavimai su parinktais optimaliais genetiniais parametrais teikia artimus sprendinius, atitinkančius visai skirtingas polių išdėstymo topologijas. Gautasis rezultatas gali būti pagerintas taikant specialias technikas (Amirjanov 2008): atrankos mechanizmo pakeitimą, GA parametrų keitimą ir pan., o dėl to vartotojas turi neišvengiamai įsikišti į optimizavimo procesą. Kita vertus, tai gali būti naudinga inžinerinėje praktikoje, nes projektuotojas gali pasirinkti tinkamą sprendinį (t. y. konkrečią polių išdėstymo topologiją) iš aibės sprendinių su skirtingomis topologijomis, bet su panašia tikslo funkcijos reikšme.

Visos šios algoritmų savybės demonstruojamos darskart sprendžiant tipinį uždavinį, kuriam sprendiniai yra gauti ir kitais metodais (Čiegis *et al.* 2006; Belevičius, Šešok 2008). Tai 15-os polių sijynas (1, 2 pav.), anksčiau optimizuotas klasikiniu GA (Goldberg 1989). Imtos tos pat genetinių parametrų reikšmės (populiacijos dydis – 30 individų, kryžminimo ir mutacijos tikimybės – atitinkamai 99,1 % ir 0,4 %), išskyrus didesnę generacijų skaičių – 500 generacijų vietoj ankstesnių 300. 30 nepriklausomų skaitinių eksperimentų leido rasti geriausią tikslo funkcijos reikšmę 162,4. Ši reikšmė yra truputį didesnė nei geriausias ankstesnių eksperimentų (Belevičius, Šešok 2008) rezultatas (157,8). Žinomas uždavinio teorinis globalusis sprendinys yra 143,0. Bet kuriuo atveju rezultatai yra panašūs ir galima daryti prielaidą, kad, atlikus 30 nepriklausomų skaitinių eksperimentų, galima gauti sprendinį apie 160,0.

I lentelėje parodyti vidutiniai visų 30 atliktų eksperimentų rezultatai. Konvergavimo istorija (sprendinių mažėjimas procentais) parodyta kas 50 generacijų. Vienas nepriklausomas atsitiktinis eksperimentas (500 generacijų) užima apie 41 min. 40 sek. (PC Pentium (R) M 1,6 GHz 512 RAM).

1 lentelė. Apibendrinti 15 polių uždavinio sprendimo rezultatai

Table 1. Generalized results of 15-pile problem

Generacijų skaičius	Vidurkis	Pagerėjimas, %
1	289,711	–
50	202,692	30,04
100	190,562	5,98
150	185,402	2,71
200	183,344	1,11
250	181,716	0,89
300	180,150	0,86
350	179,273	0,49
400	178,631	0,36
450	177,897	0,41
500	177,586	0,18

Akivaizdu, kad rezultatai labai gerėja pirmųjų 50 generacijų metu, o paieška po 200 generacijų sprendinio beveik nebergerina. Tačiau formaliai geriausias sprendinys visą laiką aptinkamas paskutinėje generacijoje. Taip yra todėl, kad naudojamame GA taikoma elitizmo strategija, visad išsauganti kelis geriausius individus.

Viena akivaizdi strategija, kuri leistų pagerinti rezultatus, būtų lokali paieška GA sprendinio, gauto iki lėtojo konvergavimo fazės, aplinkoje. Paprastai šiuo momentu sprendinys artėja kažkurio lokaliajo mini-

mumo taško link (dažniausiai ne prie globaliojo taško), ir tolesnis sprendimas GA leidžia lėtai artėti prie to taško. Čia galėtų būti naudingi jau deterministiniai optimizavimo metodai, tačiau šiems metodams būtina skaitiškai brangi jautrumo analizės informacija, todėl ir toliau taikysime stochastinį algoritmą.

4. Lokali paieška GA sprendinio aplinkoje

Lokaliajai paieškai buvo atrinkti trys geriausi sprendiniai, gauti atlikus 30 nepriklausomų atsitiktinių eksperimentų po 100 generacijų: 170,9; 177,1 ir 177,6. Stochastinė paieška apie sprendinio tašką atliekama iteracijomis tokiu būdu: kiekvienam projektavimo kintamajam (t. y. poliaus koordinatei vienmatėje konstrukcijoje) atsitiktinai generuojamas pokytis iš intervalo [–0,05 m, 0,05 m]. Intervalas toks siauras todėl, kad kartais netgi labai mažas koordinatės pokytis gali gero kai pakeisti rezultatus, pvz., kai po pozicijos pakeitimo polis pereina į kitą siją. Jeigu po pakeitimo gaunamas geresnis rezultatas, projektavimo kintamųjų reikšmės keičiamos į naujas, priešingu atveju paliekamos senos vertės. Šiam konkrečiam uždaviniui atlikta 3000 lokalių paieškos iteracijų – kad gautus rezultatus būtų galima objektyviai palyginti su ankstesnio sprendimo rezultatais (Belevičius, Šešok 2008; čia naudotas vien tik GA). Dabar tikslo funkcija skaičiuojama 6000 kartų vietoje 9000 kartų praėjusiame eksperimente.

Kadangi aprašytasis lokalių paieškos algoritmas taip pat yra stochastinis, tam pačiam sprendiniui pagerinti algoritmas kartotas tris kartus. Sprendimo rezultatai apibendrinti 2 lentelėje.

Vienas lokalių paieškos skaitinis eksperimentas vidutiniškai trunka apie 9 min 44 sek., o visas eksperimentas (GA ir lokali paieška) – 18 min 4 sek. (PC Pentium (R) M 1,6 GHz 512 RAM).

2 lentelė. Lokalių paieškos apie GA sprendinius rezultatai (15 polių uždavinys)

Table 2. Results of local search around GA solutions (15-pile problem)

Eksperimento Nr.	GA rezultatas	Lokalių paieškos rezultatas
1	170,850	157,574
2	170,850	158,549
3	170,850	158,338
1	177,068	172,852
2	177,068	173,293
3	177,068	173,179
1	177,640	157,935
2	177,640	157,884
3	177,640	158,385

3 lentelėje pateiktos dviejų geriausių sprendinių polių pozicijos vienmatėje konstrukcijoje (tikslo funkcijos reikšmės yra atitinkamai 157,574 ir 157,884). Taigi nepaisant labai mažo sprendinių skirtumo, juos atitinka visai skirtingos sijynų topologijos.

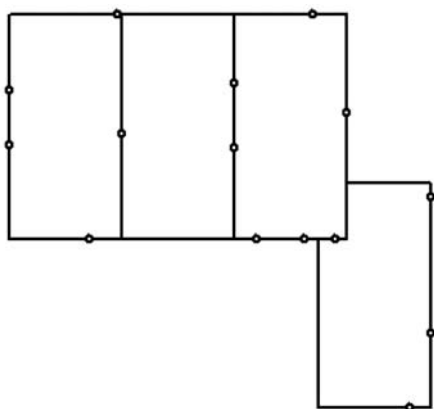
3 lentelė. Dviejų geriausių sprendinių koordinatės (15 polių uždavinys)

Table 3. Coordinates of 2 best solutions (15 pile problem)

Koordinatės Nr.	Koordinatė (1 sprendinys)	Koordinatė (2 sprendinys)
1	62,45	19,50
2	22,52	8,21
3	36,24	72,25
4	42,41	29,58
5	10,84	33,92
6	70,73	43,19
7	3,89	51,70
8	27,36	58,48
9	20,84	28,02
10	14,87	50,08
11	67,32	66,93
12	23,64	54,86
13	44,73	63,90
14	51,47	14,45
15	29,32	35,71
Tikslo funkcijos reikšmė	157,574	157,884

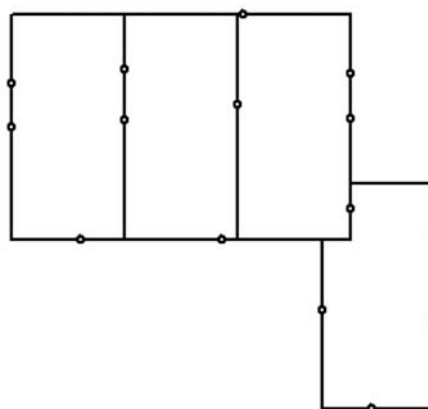
Grafiškai polių pasiskirstymas po jungiamosiomis sijomis pateiktas 1 ir 2 pav.

Pasiūlyta sprendimo strategija leidžia gauti geresnius rezultatus per trumpesnę laiką, nei taikant vien tik GA. Kitame straipsnio skyriuje aprašyti pasiūlytos strategijos taikymo rezultatai keliems kitiems tipiniams sijynų optimizavimo uždaviniams.



1 pav. Polių išdėstymo schema pirmajam sprendiniui

Fig. 1. Pile distribution scheme for 1st solution



2 pav. Polių išdėstymo schema antrajam sprendiniui

Fig. 2. Pile distribution scheme for 2nd solution

5. Skaitiniai pavyzdžiai

Pirmasis pavyzdys. Nedidelio dešimties polių sijyno optimizavimo uždavinys, kurio sprendiniai yra žinomi (Belevičius, Šešok 2008), iš naujo sprendžiamas GA su tais pačiais optimaliais šiam uždaviniui genetinėmis parametrais: populiacijos dydis – 20 individų, kryžminimo ir mutacijos tikimybės – atitinkamai 99 % ir 1 %. Tačiau, kaip ir ankstesniame pavyzdyje, generuojama tik 100 populiacijų. Atlikta 30 nepriklausomų skaitinių eksperimentų. Vidutiniškai vienas eksperimentas truko 11 min. 6 sek. Rezultatai apibendrinti 4 lentelėje.

4 lentelė. Apibendrinti 10 polių uždavinio sprendimo rezultatai

Table 4. Generalized results of 10-pile problem

Eksperimento Nr.	Tikslo funkcija	Eksperimento Nr.	Tikslo funkcija
1	202,172	16	224,768
2	221,52	17	213,947
3	241,908	18	219,082
4	207,244	19	207,764
5	220,947	20	222,286
6	225,868	21	217,222
7	223,843	22	205,688
8	220,355	23	237,228
9	235,399	24	236,859
10	226,618	25	208,65
11	218,327	26	217,122
12	235,226	27	226,473
13	239,793	28	204,985
14	236,459	29	225,087
15	251,325	30	202,951

Kitoje sprendimo fazėje imami trys geriausi GA rezultatai (202,172, 204,985 ir 202,951) ir jų aplinkoje vykdoma lokali paieška. Su kiekvienu GA sprendiniu atliekami trys nepriklausomi skaitiniai lokalaus paieškos eksperimentai, kurių rezultatai pavaizduoti 5 lentelėje.

5 lentelė. Lokalaus paieškos GA sprendinio aplinkoje rezultatai (10 polių uždavinys)

Table 5. Results of local search around GA solutions (10-pile problem)

Tikslo funkcija	GA rezultatas	Lokalaus paieškos rezultatas
1	202,951	191,294
2	202,951	191,661
3	202,951	192,053
1	202,172	189,280
2	202,172	188,431
3	202,172	189,840
1	204,985	199,503
2	204,985	199,353
3	204,985	199,940

Vienas skaitinis eksperimentas vidutiniškai truko apie 18 min. 9 sek. Bendras sprendimo laikas (GA ir lokali paieška) yra 29 min. 25 sek.

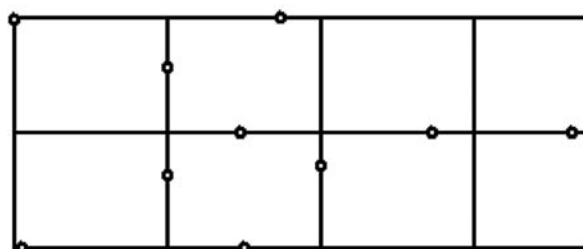
Vienmatės dviejų geriausių sprendinių koordinatės pateiktos 6 lentelėje.

Šiuo atveju abu gauti sprendiniai turi labai panašią topologiją. Grafiškai pirmąjį sprendinį atitinkantis polių išdėstymas po jungiamosiomis sijomis yra pateiktas 3 pav.

6 lentelė Dviejų geriausių sprendinių koordinatės (10 polių problema)

Table 6. Coordinates of 2 best solutions (10 pile problem)

Koordinatė Nr.	Koordinatė (1 sprendinys)	Koordinatė (2 sprendinys)
1	55,07	55,15
2	20,90	20,79
3	30,23	30,19
4	36,03	36,39
5	25,91	25,68
6	52,26	52,26
7	45,00	45,04
8	60,81	60,70
9	29,59	29,65
10	6,95	6,99
Tikslo funkcijos reikšmė	188,431	189,280



3 pav. Polių išdėstymo schema pirmajam sprendiniui

Fig. 3. Pile distribution scheme for 1st solution

Antrasis pavyzdys. Tai – praktinis sijyno optimizavimo uždavinys (Consultancy W.F.O. BV, Nyderlandai) su 40 projektavimo parametru. Šiam uždaviniui optimalūs genetiniai parametrai yra tokie: 30 individų populiacijoje, 99,75 % kryžminimo ir 0,4 % mutacijos tikimybės, 100 generacijų. Buvo atlikta 30 nepriklausomų atsitiktinių skaitinių eksperimentų. Kiekvienas eksperimentas vidutiniškai truko 5 val. 28 min. 23 sek. GA sprendimo rezultatai yra apibendrinti 7 lentelėje.

7 lentelė. Apibendrinti 40 polių uždavinio sprendimo rezultatai

Table 7. Generalized results of 40-pile problem

Eksp. Nr.	Tikslo funkcija	Eksp. Nr.	Tikslo funkcija
1	586,255	16	600,528
2	643,458	17	725,751
3	558,283	18	679,468
4	535,632	19	600,932
5	567,691	20	614,362
6	675,553	21	550,165
7	663,956	22	544,158
8	762,437	23	553,167
9	655,973	24	579,457
10	590,863	25	616,525
11	678,237	26	757,268
12	516,574	27	805,665
13	842,993	28	686,955
14	671,868	29	607,074
15	550,232	30	759,777

Trys geriausi GA rasti rezultatai (535,632; 516,574 ir 544,158) buvo atrinkti lokaliajai paieškai. Su kiekvienu GA sprendiniu atlikti trys nepriklausomi atsitiktiniai skaitiniai eksperimentai, kurių rezultatai pateikiami 8 lentelėje.

8 lentelė. Lokali paieška GA sprendinio aplinkoje rezultatai (40 polių uždavinys)

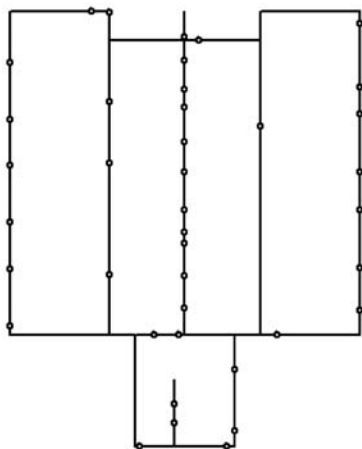
Table 8. Results of local search around GA solutions (40-pile problem).

Eksperimento Nr.	GA rezultatas	Lokali paieška rezultatas
1	535,632	445,672
2	535,632	441,863
3	535,632	446,522
1	516,574	442,922
2	516,574	442,846
3	516,574	435,411
1	544,158	432,095
2	544,158	419,661
3	544,158	414,290

Vienas skaitinis eksperimentas vidutiniškai truko apie 6 val. 25 min 50 sek. Visas sprendimas (GA ir lokali paieška) – 11 val. 54 min. 13 sek.

Vienmatės koordinatės, atitinkančios du geriausius sprendinius, pateiktos 9 lentelėje.

Šiuo atveju abu sprendinius taip pat atitinka panašios sijyno topologijos. Grafiškai polių išdėstymas po jungiamosiomis sijomis pateiktas 4 pav.



4 pav. Polių padėčių schema pirmajam sprendiniui

Fig. 4. Pile distribution scheme for 1st solution.

Taigi geriausias surastas rezultatas (414,290) yra kur kas blogesnis negu žinomas teorinis sprendinys 298,972, kuris gaunamas padalijus bendrą aktyvių apkrovų sumą iš reikiamo atramų skaičiaus sijyne. Tačiau šiandien globaliosios optimizacijos algoritmai tiesiog neleidžia gauti geresnių rezultatų per inžinerinei praktikai priimtina laiką ir taikant įprastą kompiuterinę įrangą. Pagaliau šis rezultatas yra geresnis nei deterministinių optimizavimo algoritmų rezultatai ir yra gautas per gerokai trumpesnę laiką.

9 lentelė. Dviejų geriausių sprendinių koordinatės (40 polių uždavinys)

Table 9. Coordinates of 2 best solutions (40 pile problem).

Koordinatės Nr.	Koordinatė (1 sprendinys)	Koordinatė (2 sprendinys)
1	82,37	82,36
2	68,01	68,35
3	80,92	81,04
4	9,76	9,99
5	89,28	89,31
6	65,95	65,87
7	3,66	3,90
8	22,84	22,37
9	29,75	29,99
10	43,64	43,46
11	88,46	88,27
12	73,87	73,29
13	94,47	94,00
14	91,63	91,85
15	25,44	25,70
16	113,26	113,13
17	39,90	39,57
18	90,59	90,81
19	42,65	42,12
20	36,39	35,90
21	31,62	31,31
22	27,12	26,59
23	47,55	47,40
24	18,76	18,34
25	63,40	63,68
26	85,59	85,37
27	120,95	120,93
28	102,19	102,04
29	58,79	58,91
30	61,33	61,21
31	74,73	74,31
32	82,86	82,55
33	79,51	79,29
34	76,79	76,96
35	70,55	70,53
36	86,92	86,67
37	83,87	83,47
38	4,51	4,19
39	21,62	21,46
40	99,43	99,28
Tikslo funkcijos reikšmė	414,290	419,661

6. Išvados

Stochastiniai GA gali būti sėkmingai naudojami polių padėtimis optimizuoti mažų ir vidutinių (apie 30 polių) sijynų atveju. Tačiau GA neleidžia greitai pasiekti minimumo taško iš to taško aplinkos, todėl po tam tikro sprendimo laiko (paprastai apie 100 generacijų mažiems uždaviniams) naudinga pakeisti paieškos strate-

gijas. Tai gali būti lokali deterministinė arba stochastinė paieška arba kokia nors alternatyvi GA technika. Kelių skaitinių pavyzdžių sprendimas rodo mišrios GA ir lokali stochastinės paieškos strategijos potencialą.

Padėka

Šis darbas parengtas remiant Lietuvos valstybinio mokslo ir studijų fondo projektui B-03/2007 „Globalus sudėtingų sistemų optimizavimas naudojant didelio našumo skaičiavimus ir GRID technologijas“.

Literatūra

- Amirjanov, A. 2008. Investigation of a changing range genetic algorithm in noisy environments, *International Journal for Numerical Methods in Engineering* 73(1): 26–46. doi:10.1002/nme.2053
- Barauskas, R.; Belevičius, R.; Kačianauskas, R. 2004. *Baigtinių elementų metodo pagrindai*. Vilnius: Technika. 612 p.
- Belevičius, R.; Šešok, D. 2008. Global optimization of grillages using genetic algorithms, *Mechanika* 6(74): 38–44.
- Belevičius, R.; Valentinavičius, S. 2000. Optimization of grillage-type foundations, *Journal of Civil Engineering and Management* 6(4): 255–261.
- Belevičius, R.; Valentinavičius, S. 2001. Optimisation of grillage-type foundations, in *Proceeding of 2nd European ECCO-MAS and IACM Conference "Solids, Structures and Coupled Problems in Engineering"*. Cracow, Poland 26–29 June. CD-ROM 2001.
- Belevičius, R.; Valentinavičius, S.; Michnevič, E. 2002. Multilevel optimization of grillages, *Journal of Civil Engineering and Management* 8(2): 98–103.
- Čiegis, R.; Baravykaitė, M.; Belevičius, R. 2004. Parallel global optimization of foundation schemes in civil engineering, in *Para'04: workshop on state-of-the-art in scientific computing*, June 20–23, Lyngby, 157–162.
- Čiegis, R.; Baravykaitė, M.; Belevičius, R. 2006. Parallel global optimization of foundation schemes in civil engineering, in *Applied Parallel Computing: 7th International Conference, PARA 2004, Lyngby, Denmark 20–23 June, 2004: revised selected papers. Lecture Notes in Computer Science*, Vol. 3732: 305–313.
- Goldberg, D. E. 1989. *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley, New York. 412 p.
- Groenwold, A. A.; Hindley, M. P. 2002. Competing parallel algorithms in structural optimization, *Structural and Multidisciplinary Optimization* 24(5): 343–350. doi:10.1007/s00158-002-0246-7
- Holland, J. H. 1975. *Adaptation in natural and artificial systems*. Ann Arbor: The University of Michigan Press. 211 p.
- Kennedy, S. A. 1993. Five ways to a smarter genetic algorithm, *AI Expert* 8(12): 35–38.
- Puiša, R. 2005. *The adaptive stochastic algorithms for CAD-based structure optimization of mechanical parts*: doctoral dissertation. Vilnius. 195 p.
- Ragauskas, P.; Skukis, E. 2007. Material properties identification. Comparison of two techniques, *Mechanika* 6(68): 39–44.
- Spyrakos, C.; Raftoyiannis, J. 1997. *Linear and nonlinear finite element analysis in engineering practice*. Algor Publishing Division. 498 p.
- Srinivas, M.; Patnaik, L. M. 1994. Adaptive probabilities of crossover and mutation in genetic algorithms, *IEEE Transactions on Systems, Man and Cybernetics* 24(4): 656–667. doi:10.1109/21.286385
- Šešok, D.; Belevičius, R. 2007. Use of genetic algorithms in topology optimization of truss structures, *Mechanika* 2(64): 34–39.
- Tazawa, I.; Koakutsu, S.; Hirata, H. 1996. An immunity based genetic algorithm and its application to the VLSI floorplan design problem, in *Proceedings of IEEE International Conference on Evolutionary Computation*. Nagoya, Japan 20–22 May, 417–421.
- Žilinskas, J. 2001. Black Box global optimization inspired by interval methods, *Information Technology and Control* 21(4): 53–60.
- Žilinskas, J. 2002. *Black Box global optimization: covering methods and their parallelization*: PhD dissertation. Kaunas Technological University.

GLOBAL OPTIMIZATION OF GRILLAGES WITH A HYBRID GENETIC ALGORITHM

D. Šešok, R. Belevičius

Abstract. Aim of the article is to suggest technology for optimization of pile positions in a grillage-type foundations seeking for the minimum possible pile quantity. The objective function to be minimized is the largest reactive force that arises in any pile under the action of static loading. When piles of the grillage have different characteristics, the alternative form of objective function may be employed: the largest difference between vertical reaction and allowable reaction at any pile. Several different allowable reactions with a given number of such piles may be intended for a grillage. The design parameters for the problem are positions of the piles. The feasible space of design parameters is determined by two constraints. First, during the optimization process piles can move only along the connecting beams. Therefore, the two-dimensional grillage is “unfolded” to a one-dimensional construct, and the supports are allowed to range through this space freely. Second, the minimum allowable distance between two adjacent piles is introduced due to the specific capacities of pile driver. The initial data for the problem are the following: the geometrical scheme of the grillage, the cross-section and material data of connecting beams, minimum possible distance between adjacent supports, characteristics of piles, and the loading data given in the form of

concentrated loads or trapezoidal distributed loadings. The results of solution are the required number of piles and their positions. The entire optimization problem is solved in two steps. First, the grillage is transformed to a one-dimensional construct, and the optimizer decides about a routine solution (i.e. the positions of piles in this construct). Second, the backward transformation returns the pile positions into the two-dimensional grillage, and the “black-box” finite element program returns the corresponding objective function value. On the basis of this value the optimizer predicts the new positions of piles, etc. The finite element program idealizes the connecting beams as the beam elements and the piles – as the finite element mesh nodes with a given boundary conditions in form of vertical and rotational stiffnesses. The optimizing program is an elitist genetic algorithm or a random local search algorithm. At the beginning of problem solution the genetic algorithm is employed. In the optimization problems under consideration, the genetic algorithms usually demonstrate very fast convergence at the beginning of solution and slow non-monotonic convergence to a certain local solution point after some number of generations. When the further solution with a genetic algorithm refuses to improve the achieved answer, i.e. a certain local solution is obtained; the specific random search algorithm is used. The moment, at which the transition from genetic algorithm to the local search is optimal, is sought in the paper analyzing the experimental data. Thus, the hybrid genetic algorithm that combines the genetic algorithm itself and the local search is suggested for the optimization of grillages.

Keywords: global optimization, genetic algorithms, optimization of grillages, pile

Dmitrij ŠEŠOK. Postdoctoral Researcher at the Systems Analysis Department. Institute of Mathematics and Informatics. Associate Professor at the Dept of Engineering Mechanics, Vilnius Gediminas Technical University (VGTU), Lithuania. PhD at VGTU. Research interests: stochastic optimization methods in engineering.

Rimantas BELEVIČIUS. Doctor, Professor. Head of the Dept of Engineering Mechanics, Vilnius Gediminas Technical University (VGTU), Lithuania. Research interests: global optimization, finite elements method.