

# Optimizavimo algoritmų parametrų priderinimas chaotiniams atraktoriams rekonstruoti

Miglė Drūlytė, Kristina Lukoševičiūtė, Erika Mekšunaitė

*Kauno technologijos universitetas, Matematikos ir gamtos mokslų fakultetas*

Studentų 50, LT-51368 Kaunas

E. paštas: migle.drulyte@ktu.edu, kristina.lukoseviciute@ktu.lt, erika.meksunaite@ktu.edu

**Santrauka.** Optimalių laiko vėlinimų parinkimas rekonstruojamai laiko eilutei yra svarbus uždavinys laiko eilučių analizėje bei prognozavime. Rekonstruojant laiko eilutę į nereguliarių laiko vėlinimų matavimo erdvę, laiko vėlinimų parinkimas tampa sudėtingu optimizavimo uždaviniu. Norint išspręsti šią problemą, šiame darbe pristatomi du optimizavimo algoritmai: gegučių paieškos algoritmas ir bičių spiečiaus elgsenos imitavimo algoritmas.

**Raktiniai žodžiai:** gegučių paieška, bičių spiečiaus imitavimo algoritmas, atraktoriaus rekonstravimas, laiko eilutės.

## Įvadas

Vienas svarbiausių uždavinių tiriant chaotines laiko eilutes yra jų rekonstravimas į laiko vėlinimų erdvę [2, 4]. Norint tinkamai rekonstruoti atraktorių, svarbu parinkti tinkamus rekonstravimo parametrus, t. y. rekonstravimo dimensiją  $m$  ir laiko vėlinimus  $\tau$ . Dažniausiai literatūroje sutinkamas rekonstravimas reguliariais laiko vėlinimais. Tačiau rekonstravimas naudojant nereguliarus laiko vėlinimus labiau atspindi rekonstruojamo atraktoriaus topologines savybes [2]. Nors šis rekonstravimas yra efektyvus, tačiau susiduriama su problema. Šiuo atveju turi būti optimaliai parinktas visas laiko vėlinimų rinkinys  $\tau_i$ , ( $i = 1, 2, \dots, m$ ). Literatūroje galima surasti metodus kaip genetinius (*GA*) ar skruzdžių kolonijos optimizavimo (*ACO*) algoritmus, naudojamus optimalios dimensijos ir optimalaus laiko vėlinimų rinkinio radimui [2, 4].

Norint pasiekti geresnius rekonstravimo rezultatus su mažesniais kompiuterio išnaudojimo resursais, šiame straipsnyje siūlomi du optimizavimo metodai: gegučių paieškos algoritmas (*CSA*) [5, 3] bei dirbtinio bičių spiečiaus algoritmas (*ABC*) [1]. Algoritmų tikslo funkcijomis buvo pasirinkta rekonstravimo tikslo funkcija. Remiantis straipsniu [2], optimalus sprendinys randamas su didžiausia tikslo funkcijos reikšme. Taip pat sprendinys laikomas optimaliu, jei jo tikslo funkcijos reikšmė buvo didesnė už tikslo funkcijos reikšmę esant reguliariam rekonstravimui.

## 1 Kriterijus, nusakantis nereguliaraus vėlinimo optimalumą

Šiame darbe taikomas nereguliaraus rekonstravimo į laiko vėlinimo erdvę metodas, t. y. rekonstruojant laiko eilutę tarp gretimų koordinacių bus parinkti nevienodi laiko vėlinimai. Kriterijus pagrįstas plotų skaičiavimu kiekvienoje laiko vėlinimų erdvės

projekcijoje, jis sudarytas naudojantis harmoninio signalo savybėmis, kai atraktoriaus projekcijos visose projektavimo plokštumose yra elipsės [2]. Skirtingų projekcijų skaičius  $m$ -matėje vėlinimo koordinatinių erdvėje yra  $\frac{m(m-1)}{2}$ . Rekonstravimo į vėlinimų erdvę kokybės funkcija, charakterizuojanti atraktoriaus dinamiką rekonstruotoje erdvėje aprašoma (1) formule [2]:

$$Q(\tau_1, \dots, \tau_{m-1}, \omega) = \frac{2}{m(m-1)} \left( \sum_{i=1}^{m-1} |\sin(\omega\delta\tau_i)| \right) + \frac{2}{m(m-1)} \left( \sum_{i=1}^{m-2} |\sin(\omega\delta(\tau_i + \tau_{i+1}))| + \left| \sin \left( \omega\delta \sum_{j=1}^{m-1} \tau_j \right) \right| \right), \quad (1)$$

čia  $\tau_i$  – nereguliaraus rekonstravimo laiko vėlinimas,  $\tau_i \in N$ ,  $i = 1, 2, \dots, (m-1)$ ,  $\delta$  – intervalas tarp gretimų skaliarinės laiko eilutės taškų,  $\omega$  – harmoninio signalo ciklinis dažnis. Sukonstruota tikslo funkcija, kuri įvertina, kaip atskiros diskrečiosios amplitudinio spektro komponentės yra paveikiamos rekonstrukcijos, kokybės funkcijos prasme. Tai reprezentuoja tikslo funkcijos  $F$  netiesioginis integralas įverčio skaitiklyje, kai pereinami visi dažniai ir surandama kokias Furjė amplitudinio spektro  $A(\omega)$  komponentes labiausiai paveikė kokybės funkcija  $Q$ .

$$F(\tau_1, \dots, \tau_{m-1}) = \frac{\pi}{2} \cdot \frac{\int_0^\infty A(\omega) \cdot Q(\tau_1, \dots, \tau_{m-1}, \omega) d\omega}{\int_0^\infty A(\omega) \cdot d\omega}. \quad (2)$$

Skaitinė tikslo funkcijos  $F$  reikšmė, gauta pagal (2) formulę, naudojama kaip rodiklis, rodantis prie kokio nereguliaraus laiko vėlinimų rinkinio  $\{\tau_1, \dots, \tau_{m-1}\}$  laiko eilutės atraktorius užima didžiausią turį vėlinimų erdvėje.

## 2 Gegučių paieškos algoritmas

Gegučių paieškos algoritmas prasideda su pradine gegučių populiacija [3], kuri turi padėti tam tikrą skaičių kiaušinių svetimuose lizduose. Kiaušiniai, kurie yra panašūs į lizdų šeimininkų kiaušinius, turi galimybę užaugti, kiti yra pašalinami. Išgyvenę kiaušiniai parodo lizdų tinkamumą toje srityje. Kuo daugiau išgyvena kiaušinių tame regione tuo didesnė to regiono vertė.

Gegutės ieško pačio tinkamiausio regiono padėti kiaušinius, kad išgyvenimo lygis būtų maksimalus. Užaugusios gegutės sudaro tam tikrą grupę, kuri turi savo regioną. Gegučių grupės migruoja link regiono su didžiausiu išgyvenimo lygiu ir apsigyvena jo apylinkėse. Atsižvelgiant į vienos gegutės padedamų kiaušinių skaičių ir į atstumą iki geriausio regiono yra nustatomas kiaušinių dėjimo spindulys. Gegutės deda kiaušinius į atsitiktinai pasirinktus lizdus tuo spinduliu. Šis procesas tęsiamas tol kol surandamos maksimalios vertės pozicijos.  $N_{var}$  – dimensijos optimizavimo problemoje *aplinka* reprezentuoja dabartinę gegutės gyvenimo vietą ir užrašomas [3]:

$$aplinka = [x_1, x_2, \dots, x_{N_{var}}]. \quad (3)$$

Aplinkos vertė randama skaičiuojant tikslo funkcijos  $f_p$  vertes:

$$verte = f_p(aplinka) = f_p(x_1, x_2, \dots, x_{N_{var}}). \quad (4)$$

Šis optimizavimo uždavinys yra tikslo funkcijos maksimizavimo uždavinys. Gamtoje, kiekviena gegutė padeda nuo 5 iki 20 kiaušinių. Šios vertės yra naudojamos kaip apatinė  $var_{low}$  ir viršutinė  $var_{hi}$  riba kiaušinių, kuriuos gali padėti kiekviena gegutė. Gegutės deda kiaušinius maksimaliu atstumu nuo jų gyvenamos aplinkos, kuris randamas pagal (5) formulę [3]:

$$ELR = \alpha \times \frac{DKS}{VKS} \times (var_{hi} - var_{low}), \quad (5)$$

kur  $DKS$  yra dabartinis kiaušinių skaičius,  $VKS$  yra visas kiaušinių skaičius,  $\alpha$  yra skaičius, kuris turėtų kontroliuoti maksimalią  $ELR$  vertę.

### 3 Bičių spiečiaus imitavimo algoritmas

$ABC$  algoritmas susideda iš trijų fazių, vadinamų: bičių darbininkių, bičių stebėtojų ir bičių žvalgių [1]. Kiekvienam maisto šaltiniui yra tik viena įdarbinta bitė. Bitė darbininkė apleidusi maisto šaltinį tampa bite žvalge. Dirbtinų bičių atliekama paieška gali būti užrašoma [1]:

- Bitės ieško kuo vertingesnių maisto šaltinių ir juos įsimena.
- Bitės darbininkės eikvoja konkretų joms priskirtą maisto šaltinį ir saugo informaciją apie jį. Jos gali pasidalinti šia informacija su kitomis bitėmis.
- Neįdarbintos bitės – ieško maisto šaltinių. Stebėtojos – laukia avilyje ir pasirenka geriausią maisto šaltinį stebėdamos įdarbintų bičių „šokį“, kurio metu bitės darbininkės pasidalija informacija apie jų rastus maisto šaltinius, žvalgės – atsitiktai ieško naujų maisto šaltinių avilio aplinkoje.

$ABC$  algoritmu bitės atsitiktinai randa pradinių sprendinių populiaciją, kuri toliau iteraciniu būdu gerinama, judant link geresnio sprendinio. Tam naudojamas kaimyninės paieškos metodas. Maisto šaltinio pozicija reprezentuoja galimą sprendinį, o maisto šaltinio nektaro kiekis atitinka sprendinio kokybę.

### 4 Eksperimentiniai rezultatai

Eksperimentai buvo atliekami su chaotine Mackey–Glass laiko eilute, kuri yra išreikšiama diferencialine lygtimi [2, 4]:

$$\frac{dx}{dt} = \frac{0.2x(t-\lambda)}{1+x^{10}(t-\lambda)} - 0.1x(t). \quad (6)$$

Šioje lygtyje laiko vėlinimo konstanta ir modelio parametrai parinkti tokie ( $\lambda = 17$ ), kad Mackey–Glass lygtis generuotų chaotinę laiko eilutę. Pagal straipsnį [2] šią eilutę geriausia rekonstruoti į šešiamatę laiko vėlinimų fazinę erdvę,  $m = 6$ . Laiko eilutės ilgis yra  $N = 1000$ . Ieškosime tokio laiko vėlinimų rinkinio  $\{\tau_1, \tau_2, \tau_3, \tau_4, \tau_5\}$ , su kuriuo tikslo funkcijos reikšmė būtų didesnė už tikslo funkcijos reikšmę, kai chaotinė laiko eilutė rekonstruojama į fazinę erdvę su reguliariais laiko vėlinimais, t. y.  $F(9, 9, 9, 9, 9) = 1.0990$  gaunama su laiko vėlinimų rinkiniu  $\{9, 9, 9, 9, 9\}$ . Jei ši sąlyga tenkinama, tai toks sprendinys laikomas sėkmingu. Lukoševičiūtės ir Ragulskio [2] straipsnyje siūloma laiko vėlinimų  $\tau_i \in N$ ,  $i = 1, 2, 3, 4, 5$  reikšmės apriboti intervalu  $1 \leq \tau \leq T$ , kur  $T = 30$ .

1 lentelė. Parametrų palyginimas.

	$\tau_i$	$F_{\max}$
Populiacija	(7, 5, 5, 5, 5)	1.1012
$var_{low}$	(11, 5, 5, 5, 5)	1.1003
$var_{hi}$	(7, 4, 5, 5, 6)	1.1049

#### 4.1 Gegučių paieškos algoritmas

Visų pirma yra parenkami gegučių paieškos algoritmo parametrai. Randama pradinė gegučių populiacija, t. y. tikrinama tikslo funkcijos reikšmės, keičiant pradinę gegučių populiaciją intervale [1, 35]. Imant didesnes reikšmes nei 35, funkcijos reikšmė  $F\{\tau_1, \tau_2, \tau_3, \tau_4, \tau_5\} < 1.0796$ . Didžiausia tikslo funkcijos reikšmė gaunama, kai pradinis gegučių populiacijos skaičius yra lygus 8. Tokiu atveju tikslo funkcijos reikšmė  $F_{\max} = 1.1012$  su laiko vėlinimų rinkiniu {7, 5, 5, 5, 5}.

Toliau ieškoma, koks turi būti apatinis rėžis, t. y. minimalus padėtų kiaušinių skaičius. Šis skaičius ieškomas iš intervalo [2, 20], imant didesnę kiaušinių skaičių nei 20, tikslo funkcijos reikšmės tik mažėja, be to remiantis gamta, gegutės gali padėti iki 20 kiaušinių. Didžiausia tikslo funkcijos reikšmė gaunama, kai apatinis rėžis, t. y. minimalus skaičius padėtų kiaušinių yra lygus 6. Tokiu atveju tikslo funkcijos reikšmė  $F_{\max} = 1.1003$ , su laiko vėlinimų rinkiniu {11, 5, 5, 5, 5}.

Galiausiai ieškomas maksimalus kiekis kiaušinių, kuriuos gali padėti viena gegutė. Ši reikšmė gaunama iš intervalo [7, 20]. Didžiausia tikslo funkcijos reikšmė gaunama, kai viršutinis rėžis yra lygus 8. Tokiu atveju tikslo funkcijos reikšmė  $F_{\max} = 1.1049$ , su laiko vėlinimų rinkiniu {7, 4, 5, 5, 6}. Apibendrinti rezultatai pateikti 1 lentelėje.

Didžiausia tikslo funkcijos reikšmė gaunama, kaip pradinis gegučių populiacijos skaičius yra lygus 8 ir kiekviena gegutė gali padėti nuo 6 iki 8 kiaušinių.

#### 4.2 Dirbtinio bičių spiečiaus algoritmas

Pirmiausia tiriamame populiacijos dydį. Imame populiacijos reikšmes lygias: 10, 20, 30, 40, 50, 60, 70, 80, 90, 100. Ištyrus populiacijos dydžio kitimą, didžiausia tikslo funkcijos reikšmė  $F_{\max} = 1.1056$  pasiekama, kai populiacijos dydis  $m = 90$  (bičių kolonijos dydis lygus  $2m = 180$ ).

Tiriamame iteracijų skaičių kas 10 vienetų, t. y. 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110. Ištyrus maksimalaus iteracijų skaičiaus kitimą, didžiausia  $F_{\max} = 1.1056$  pasiekama naudojant 80 iteracijų.

Su populiacijos dydžiu  $m = 90$  ir maksimaliu iteracijų skaičiumi  $g = 80$  toliau tiriamame *Limit* dydžio kitimą. *Limit* – kontroliuoja bičių žvalgių fazės vykdymą. Likantis standartinio ABC algoritmo prielaidos, kad kiekvienos iteracijos metu galima ne daugiau kaip viena žvalgė, pasirenkame straipsnyje [1] siūlomas *Limit* dydžio tyrimo rekomendacijas:  $Limit = (m \cdot n)$ ,  $0.1 \cdot (m \cdot n)$ ,  $0.5 \cdot (m \cdot n)$ , kur  $m$  yra populiacijos dydis t. y. maisto šaltinių skaičius populiacijoje, o  $n$  yra maisto šaltinio parametrų skaičius. Taigi tirsime *Limit*: 450, 45, 225. Taip pat tiriamas algoritmas, kai visai nedominuoja žvalgių fazė, ir kai kiekvieno ciklo metu dalyvauja po vieną bitę žvalgę (t. y.  $Limit = 0$ ).

Iš 2 lentelės matyti, kad didžiausia tikslo funkcijos reikšmė  $F(7, 4, 5, 4, 7) = 1.1056$  gaunama, kai  $Limit = 45$ . Atsižvelgiant į gautus rezultatus, didžiausia tikslo funkcijos

**2 lentelė.** *Limit* rezultatų palyginimas.

	0	45	225	450
$F_{\max}$	1.1009	1.1056	1.0955	1.1007
$\tau_i$	(9, 4, 7, 9, 11)	(7, 4, 5, 4, 7)	(5, 7, 9, 10, 10)	(8, 8, 4, 9, 7)

**3 lentelė.** Algoritmų palyginimas.

	$\tau_i$	$F_{\max}$	Laikas, s
<i>GA</i>	(6, 5, 5, 5, 5)	1.1026	171.65
<i>CSA</i>	(7, 4, 5, 5, 6)	1.1049	70.63
<i>ABC</i>	(7, 4, 5, 4, 7)	1.1056	132.66

reikšmė gaunama, su populiacijos dydžiu  $m = 90$ , maksimaliu iteracijų skaičiumi  $g = 80$  ir  $Limit = 45$ .

Algoritmų palyginimas pateiktas 3 lentelėje, palyginimui taip pat naudojamas genetinis algoritmas (*GA*) [2].

## 5 Išvados

Gegučių paieškos algoritmas, bei dirbtinio bičių spiečiaus algoritmas buvo pasiūlyti optimaliam, rekonstruojamos laiko eilutės į nereguliarių laiko vėlinimų erdvę, laiko vėlinimų rinkinio parinkimui. Eksperimentiniai rezultatai parodė, kad pasiūlyti algoritmai gali ganėtinai lengvai ir greitai surasti rekonstravimo parametrus. Kaip eksperimentai parodė, gegučių paieškos algoritmas laiko atžvilgiu yra tinkamiausias naudojimui, tačiau bičių spiečiaus imitavimo algoritmas tikslumo prasme buvo geresnis, t.y. surado didesnę tikslo funkcijos reikšmę ir ją surado daugiau kartų nei kiti optimizavimo algoritmai.

## Literatūra

- [1] D. Karaboga and B. Basturk. On the performance of artificial bee colony (*ABC*) algorithm. *Appl. Soft Comput.*, **8**:687–697, 2008.
- [2] K. Lukoševičiūtė and M.K. Ragulskis. Evolutionary algorithms for the selection of time lags for time series forecasting by fuzzy inference systems. *Neurocomputing*, **73**:2077–2088, 2010.
- [3] R. Rajabioun. Cuckoo optimization algorithm. *Appl. Soft Comput.*, **11**:5508–5518, 2011.
- [4] M. Shen, W.-N. Chen, J. Zhang, H.S.-H. Chung and O. Kaynak. Optimal selection of parameters for nonuniform embedding of chaotic time series using ant colony optimization. *IEEE Trans. Cybern.*, **43**(2):790–802, 2013.
- [5] X.S. Yang. *Nature-Inspired Metaheuristic Algorithms. Second Edition*. Luniver Press, Frome, BA11 6TT, United Kingdom, 2010.

## SUMMARY

### Adaptation of optimization algorithm parameters to reconstruction of chaotic attractor

*M. Drulytė, K. Lukoševičiūtė, E. Mekšunaitė*

Optimal selection of time delay for time series reconstruction is an important problem in time series analysis and forecasting. When reconstructing the time series into phase space with non-uniform

time delay, a time delay selection becomes a difficult optimization problem. To solve this problem, this paper presents two optimization algorithms: cuckoo search algorithm and artificial bee colony optimization algorithm.

*Keywords:* cuckoo search (*CS*), artificial bee colony (*ABC*), attractor embedding, time series.