

# Pareto tipo modeliai modeliuojamojo atkaitinimo algoritmuose

Gražvydas FELINSKAS, Leonidas SAKALAUSKAS (ŠU)  
*el. paštas:* grazvis@fm.su.lt

## Įvadas

Praktiniai optimizavimo uždaviniai dažnai yra tokie sudėtingi, kad uždavinį laikyti iškilu nerealu. Globalaus optimumo neiškiluose uždaviniuose radimui yra taikomi keletas principinių algoritmų. Daugeliu atvejų šių algoritmų elgesys gali būti nagrinėjamas atsitiktinių klaidžiojimų teorijos požiūriu, minimizuojant arba maksimizuojant tikslo funkciją lokalinės (pvz. stochastinių gradientų metodais) arba globalinės (modeliuojamojo atkaitinimo, genetiniai algoritmai, Monte Karlo [7] metodais) paieškos būdais. Sprendžiant praktinius globalinės optimizacijos uždavinius susiduriama su bendrų kriterijų, kuriais remiantis būtų galima tvirtinti, jog pasiektas globalus optimumas, stoka, bei rekomendacijų algoritmų parametrų efektyviam parinkimui nebuvimu [2,5].

Modeliuojamojo atkaitinimo (simulated annealing – SA) metodo konvergavimo teoriniai tyrimai rodo, kad šiuose algoritmuose paieškos sekos formavimui efektyviai gali būti pritaikomi Pareto tipo modeliai [2], t.y. tokie modeliai, kurių skirstiniai turi „sunkias uodegas“ [1]. Taikydami modeliuojamojo atkaitinimo algoritmą, mes galime įsitikinti, jog tokie modeliai leidžia efektyviau rasti globalų optimumą.

Darbo tikslas – ištirti konvergavimo į globalų ekstremumą greitį kompiuterinio modeliavimo būdu. Algoritmo efektyvumo charakteristikomis yra laikomos minimizuojamos funkcijos reikšmė ir globalaus optimumo radimo tikimybė po tam tikro iteracijų skaičiaus.

## Modeliuojamojo atkaitinimo (Simulated Annealing) algoritmas

Nagrinėkime optimizavimo uždavinį (minimizavimo)

$$\min_{x \in \Xi} f(x), \quad (1)$$

čia  $\Xi$  yra  $R^n$  poaibis ir  $f$  yra reali funkcija, apibrėžta srityje  $\Xi$  ir turinti globalų minimumą  $f^*$  srityje  $\Xi$ .

Aprašysime SA algoritmą sprendžiant (1) uždavinį [2,3,4].

**1 žingsnis.** Pasirenkame pradinį tašką  $x^0 \in \Xi$ , pradinę temperatūrą  $T_0 > 0$ , tam tikrą nuo temperatūros priklausančią generavimo tikimybinę tankio funkciją, atitinkamą temperatūros atnaujinimo funkciją, ir seką  $\{\rho_j; j \geq 0\}$  monotoniškai mažėjančių teigiamų skaičių. Suskaičiuojame  $f(x^0)$ . Nustatome  $X^0 = x^0$  ir  $k = 0$ .

**2 žingsnis.** Generuojame atsitiktinį vektorių  $Z^k$  naudodami generavimo tikimybinę tankio funkciją. Jei egzistuoja  $1 \leq i \leq n$  toks, kad  $|Z_i^k| < \rho_k$ , kur  $Z_i^k$  yra  $i$ -asis vektoriaus  $Z^k$  komponentas, kartoti 2 žingsnį. Kitu atveju, gaunamas naujas taškas  $Y^k$  pridant atsitiktinį vektorių  $Z^k$  prie dabartinės iteracijos taško  $X^k$ ,

$$Y^k = X^k + Z^k. \quad (2)$$

Jei  $Y^k \notin \Xi$ , kartojame 2 žingsnį; kitu atveju, skaičiuojame  $f(Y^k)$ .

**3 žingsnis.** Naudojame Metropolio kriterijų naujam iteracijos taškui  $X^{k+1}$  parinkti, t.y. generuojame atsitiktinį skaičių  $\eta$  tolygiai pasiskirsčiusi intervalė (0, 1), ir tikimybę  $P(Y^k, X^k, T_k)$ . Jei  $\eta \leq P(Y^k, X^k, T_k)$ , tada  $X^{k+1} = Y^k$  ir  $f(X^{k+1}) = f(Y^k)$ , kitu atveju paliekame  $X^{k+1} = X^k$  ir  $f(X^{k+1}) = f(X^k)$ ; čia

$$P(Y^k, X^k, T_k) = \min \{1, \exp\{[f(X^k) - f(Y^k)]/T_k\}\}. \quad (3)$$

**4 žingsnis.** Jei iš anksto numatyta sustojimo sąlyga (pvz. nurodytas iteracijų skaičius) yra patenkinta, tada sustoti. Kitu atveju atnaujiname temperatūrą naudodami temperatūros atnaujinimo funkciją ir grįžtame prie 2 žingsnio.

Aprašytas metodas yra nagrinėtas teoriškai [2,3,4,6], tačiau išsamių skaičiuojamųjų tyrimų dar nėra padaryta. Šiame darbe daugiausia remiamasi R.L. Yang straipsniu [2], kuriame nustatytos SA algoritmo sąlygos. Šios sąlygos nustato temperatūros atnaujinimo funkcijos pavidalą, parametrus bei teigiamų skaičių sekos  $\{\rho_j; j \geq 0\}$  parinkimą, garantuojanti modeliujamojo atkaitinimo algoritmo konvergavimą į tikslo funkcijos globalų minimumą tiriamoje srityje. Teigiamų skaičių seka  $\{\rho_j; j \geq 0\}$  kiekvienoje iteracijoje nustato mažiausią ribą, kurią turi viršyti generuojamo vektoriaus  $Z^k$  koordinatės.

Teoremos išvadose (pasekmėse) tvirtinama, kad temperatūros atnaujinimo funkcija  $T_k$  bei teigiamų skaičių seka  $\{\rho_j; j \geq 0\}$  turi būti suderintos su generavimo tikimybine tankio funkcija [2]. Straipsnyje išnagrinėti keli generavimo skirstiniai, pasižymintys Pareto savybe (Pareto tipo modeliai pasižymi „sunkiomis uodegomis“). Pagrindinis šių modelių parametras –  $\alpha$ , kuris reiškia „uodegos sunkumo“ laipsnį. Tiriamame modelyje (b)  $\alpha = 1/m$  (1 lentelė).

Darbe naudojami šie žymėjimai:  $0 < \rho_0 < \min_{1 \leq i \leq n} r_i$ , čia  $r \in R^n$ ,  $r_i = \max_{x,y \in \Xi} |x_i - y_i|$ ,  $1 \leq i \leq n$ ,  $0 \leq \gamma \leq 1/n$ ,  $\{T_k; k \geq 0\}$  – temperatūrų seka,  $m \geq 1$  – sveikas skaičius,  $\lambda > 1$ ,  $0 < \gamma < \min\{\lambda, m/n\}$ . Remiantis 1 lentele, buvo gautos taisyklės dydžiams  $Z_k$  antrajame algoritmo žingsnyje modeliuoti.

## Algoritmų testavimo metodika

Globalinės optimizacijos uždaviniuose taikant įvairius optimizavimo algoritmus reikia patikrinti algoritmų ir metodų patikimumą, efektyvumą. Tam naudojamos įvairios testinės funkcijos, kurių optimumo taškai iš anksto žinomi. Yra specialiai sukonstruojamos įvairios funkcijos, kurios turi vieną ar kelis globalius minimumus, taip pat gali turėti ir lokalių minimumų. Šių funkcijų pagalba mes galime patikrinti, ar mūsų naudojamas algoritmas „neužstringa“ lokaliame minimume, galime stebėti konvergavimo greitį, tikslumą

1 lentelė. Generavimo skirstiniai ir atitinkamos SA algortimo charakteristikos

Tankis	Tikimybė $F(z) = P(Z < z)$	$\rho_k, T_k$
(a) $p(z, T_k) = \frac{T_k}{\pi(z^2 + T_k^2)}$	$P(Z < z, T_k) = \frac{1}{\pi}(\arctg(\frac{z}{T_k}) + \frac{\pi}{2})$	$\rho_k = \rho_0/k^{\gamma/4n}$ $T_k = T_0/k^{1/n}$
(b) $p(z, T_k) = \frac{T_k^{1/m}}{2m[ z  + T_k]^{(m+1)/m}}$	$P(Z < z, T_k) = \frac{1}{2(1 - \frac{z}{T_k})^{1/m}}, z < 0$ $P(Z < z, T_k) = 1 - \frac{1}{2(\frac{z}{T_k} + 1)^{1/m}}, z > 0$	$\rho_k = \rho_0/k^{\gamma m/\lambda(m+1)n}$ $T_k = T_0/k^{m/n}$

ir kitus parametrus. Padarę tinkamas išvadas, parinkę tinkamus parametrus, algoritmus ir metodus galime taikyti ir kitoms funkcijoms, kitiems optimizavimo uždaviniams spręsti. Buvo naudojamos šios testinės funkcijos: *Branino* (turi 3 globalius ekstremumus), *Rastrigino* (maždaug 50 lokalinių ekstremumų ir vienas globalus), *Goldšteino–Praisio* (4 lokalūs ekstremumai ir vienas globalus) [5].

Buvo tiriamos šios algoritmo efektyvumo charakteristikos: minimizuojamos funkcijos reikšmė bei globalaus optimumo radimo tikimybė po tam tikro iteracijų skaičiaus. Šios charakteristikos buvo vertinamos Monte–Karlo metodu, atliekant  $N$  nusileidimų po  $K$  iteracijų ( $N = 1000$ , kai  $K \leq 3000$ ,  $N = 500$ , kai  $3000 < K \leq 10000$  ir  $N = 100$ , kai  $K > 10000$ ). Efektyvumo charakteristikų pasikliautiniai intervalai buvo nustatomi remiantis normaline aproksimacija. Globalaus optimumo radimo tikimybė buvo vertinama apskaičiuojant optimizavimo taško patekimo dažnumą į globalaus optimumo aplinką:  $|f_k - f^*| < \varepsilon$ . Pradinis taškas buvo parenkamas atsitiktinai.

Modeliuojant SA algoritmą testinėms funkcijoms su dviem skirtingais skirstiniais bei manipuliuojant įvairiais leidžiamais keisti parametrais, buvo siekiama išsiaiškinti:

- kuris iš duotų skirstinių užtikrina spartesnę konvergavimą į globalų minimumą pagal funkcijos reikšmę;
- kokios yra „nusileidimo“ į globalų minimumą tikimybės ir kaip jas gali paveikti tam tikrų parametru keitimas;
- koks reikalingas iteracijų skaičius, kad surasti globalų minimumą su reikalinga tikimybe.

Tyrimų metu naudotos konkrečios parametru reikšmės:  $\lambda = 2$ ,  $\gamma = 0,3$  (a),  $\gamma = 0,9$  (b),  $\rho_0 = 0,01$  (Goldšteino–Praisio ir Rastrigino funkcijoms),  $\rho_0 = 0,1$  (Branino funkcijai),  $T_0 = 10$ . (Jie buvo keičiami, bet su šiomis reikšmėmis gauta pastebimai gerų rezultatų.)

## Modeliavimo rezultatai

Modeliavimo rezultatai pateikti lentelėse. Matyti, jog naudojant (b) skirstinį konvergavimas į globalų minimumą pagal funkcijos reikšmę yra spartesnis (2 lentelė). Buvo

2 lentelė. Konvergavimas į globalų minimumą pagal funkcijos reikšmę  $F_k$ 

Iteracijų skaičius	$F_k$ reikšmės					
	Branino f-ja		Goldšteino–Praisio f-ja		Rastringo f-ja	
	(a) skirst.	(b) skirst.	(a) skirst.	(b) skirst.	(a) skirst.	(b) skirst.
100	1,41265	0,052039	13,45722	9,87812	-0,40869	-1,67522
500	0,83006	0,417289	3,75233	5,53644	-1,20333	-1,94236
1000	0,68788	0,406829	3,36933	4,73926	-1,32400	-1,98219
3000	0,54042	0,401299	3,16755	3,78732	-1,64010	-1,99647
10000	0,48771	0,39878	3,10922	3,00125	-1,79196	-1,99814
30000	0,44670	0,39828	3,05393	3,00038	-1,90625	-1,99957
Glob. min.	0,397887		3,0		-2,0	

3 lentelė. Patekimo į globalų minimumą tikimybės Goldšteino–Praisio funkcijai ( $\epsilon = 0.01$ )

Iteracijų skaičius	Patekimo į globalų minimumą tikimybės		
	$m = 2$	$m = 3$	$m = 4$
100	0,025 ± 0,008	0,160 ± 0,019	0,248 ± 0,023
200	0,145 ± 0,018	0,482 ± 0,026	0,427 ± 0,026
300	0,261 ± 0,023	0,598 ± 0,026	0,495 ± 0,026
500	0,485 ± 0,026	0,627 ± 0,025	0,527 ± 0,026
1000	0,764 ± 0,022	0,658 ± 0,025	0,530 ± 0,026

nagrinėjama parametro  $m$  įtaka konvergavimo greičiui (3 lentelė).

Patekimo į globalų minimumą tikimybės *Goldšteino–Praisio* funkcijai:

1) Kai  $m = 2$ . Garantuoja didesnes patekimo į globalų minimumą tikimybes nei su kitomis parametro  $m$  reikšmėmis, ypač, jei norime didesnio tikslumo. Ši parametro reikšmė rekomenduojama, kai iteracijų skaičius didelis. Tikimybės, esant mažam iteracijų skaičiui, auga lėčiau nei prie didesnių  $m$  reikšmių.

2) Kai  $m = 3$ . Garantuoja didesnes patekimo į globalų minimumą tikimybes, kai iteracijų skaičius nedidelis. Didinant iteracijų skaičių, tikimybės auga lėtai. Kai iteracijų skaičius yra 1000 ir daugiau, rekomenduojama parametro reikšmė  $m = 2$ .

3) Kai  $m = 4$ . Garantuoja dar didesnę globalaus minimumo suradimo tikimybę nei atvejais  $m = 2$  ir  $m = 3$ , kai iteracijų skaičius mažas (100).

Iš šių trijų atvejų akivaizdu, kad  $m = 3$  ir  $m = 4$  atvejais konvergavimas yra greitesnis, bet su „blogo“ tikimybe, tuo tarpu  $m = 2$  atveju konvergavimas lėtesnis, bet tikimybė gana didelė.

*Goldšteino–Praisio* funkcijai modeliavimo būdu buvo nustatytos patekimo į lokaliuosius minimumus ir į globalų minimumą tikimybės. Buvo naudojamas (b) skirstinys, kuris užtikrina spartesnę konvergavimą (4 lentelė).

Kaip matyti iš šio tyrimo, algoritmas praktiškai niekada „neužstringa“ 1-ame ir 4-ame lokaliame minimume. „Užstrigimo“ kituose dviejuose lokaliuose minimumuose ti-

4 lentelė. Patekimo į lokaliuosius minimumus ir globalųjį minimumą tikimybės Goldšteino–Praisio funkcijai ( $m = 2$ ,  $\varepsilon = 0, 1$ , iki 1000 realizacijų)

Iteracijų skaičius	Patekimo į globalųjį minimumą tikimybės				
	Į lok. min. $F(1, 2; 0, 8)$ = 840	Į lok. min. $F(1, 8; 0, 2)$ = 84	Į lok. min. $F(-0, 6; -0, 4)$ = 30	Į lok. min. $F(-0, 4; -0, 6)$ = 35	Į glob. min. $F(0; -1) = 3$
<b>100</b>	0	0,003	0,005	0,001	<b>0, 111 ± 0, 016</b>
<b>300</b>	0	0,011	0,012	0	<b>0, 799 ± 0, 021</b>
<b>500</b>	0	0,016	0,016	0	<b>0, 948 ± 0, 012</b>
<b>1000</b>	0	0,017	0,009	0	<b>0, 973 ± 0, 008</b>
<b>3000</b>	0	0,007	0,003	0	<b>0, 984 ± 0, 007</b>
<b>5000</b>	0	0,002	0	0	<b>0, 993 ± 0, 004</b>
<b>10000</b>	0	0	0	0	<b>1</b>

5 lentelė. Globalaus minimumo, kai  $\varepsilon = 0, 001$  suradimo tikimybės Goldšteino–Praisio funkcijai

Iteracijų skaičius	Tikimybė
<b>1000</b>	<b>0, 09 ± 0, 047</b>
<b>5000</b>	<b>0, 32 ± 0, 077</b>
<b>10000</b>	<b>0, 55 ± 0, 082</b>
<b>20000</b>	<b>0, 79 ± 0, 067</b>
<b>30000</b>	<b>0, 93 ± 0, 042</b>
<b>50000</b>	<b>0, 97 ± 0, 028</b>

kimybės taip pat mažos, bet sąlyginai yra didžiausios, kai iteracijų skaičius nedidelis (300–1000). Esant dideliame iteracijų skaičiui, algoritmas su tikimybe artima vienetui nurodytu tikslumu patenka į globalų minimumą.

Ištirta ir globalaus minimumo suradimo su dideliu tikslumu tikimybė Goldšteino–Praisio funkcijai. Parinktos parametrų reikšmės  $\varepsilon = 0.001$ ,  $m = 2$ . Atlikta 100 realizacijų kiekvienam atvejui. Kaip matyti iš tyrimo rezultatų, iteracijų skaičių parinkus 30000 ir didesnę, tikslaus minimumo suradimo tikimybės gana didelės (artimos vienetui) (5 lentelė).

**Pastaba.** 3, 4 ir 5 lentelėse pasikliautiniai intervalai buvo nustatyti remiantis normaline aproksimacija.

## Išvados

Galima įsitikinti, kad Pareto tipo modeliai su „sunkių uodegų“ savybe garantuoja didesnę globalaus optimimo suradimo tikimybę. Todėl šių modelių taikymas suteikia galimybę pagerinti euristinių algoritmų efektyvumą.

Reikia remtis teoriniais rezultatais ([2]), kad įvairiems skirtingiems modeliams turime parinkti tam tikras atitinkamas  $\rho_k$  ir  $T_k$  sekų atnaujinimo funkcijas.

Algoritmo efektyvumui padidinti, pirmiausia (kai iteracijų skaičius yra mažas) reikėtų parinkti mažą parametro  $\alpha$  reikšmę, vėliau (didėjant iteracijų skaičiui) parametro  $\alpha$  reikšmę rekomenduojame didinti.

## Literatūra

- [1] A. Janicki, A. Weron, *Simulation and Chaotic Behavior of Stable Processes*, Marcell Dekker, N.Y. (1994).
- [2] R.L. Yang, Convergence of the simulated annealing algorithm for continuous global optimization, *Journal of Optimization Theory and Applications*, **104**(3), 691–716 (2000).
- [3] M. Locatelli, Simulated annealing algorithms for continuous global optimization: convergence conditions, *Journal of Optimization Theory and Applications*, **104**(1), 121–133 (2000).
- [4] M. Lundy, A. Mees, Convergence of an annealing algorithm, *Mathematical Programming*, North–Holland, **34**, 111–124 (1986).
- [5] A. Čilinskas, *Global optimization*, Vilnius, Mokslas (1986) (in Russian).
- [6] G. Dzemyda, E. Senkienė, Convergence of the parameter clustering based on the simulated annealing, *Informatica*, **8**(4), 465–476 (1997).
- [7] L. Sakalauskas, Non-linear stochastic programming by Monte–Carlo estimators, *European Journal on Operational Research*, **137**, 558–573 (2002).

## Parrett's type models in simulated annealing algorithms

G. Felinskas, L. Sakalauskas

In this paper a simulated annealing algorithm for continuous global optimization is considered. There are a lot of theoretical research and very few computer modeling of efficiency of this algorithm. Modeling results, recommendations for efficiency and accuracy of the SA algorithm are given in the paper.