KOMPAKTNÍ DIFERENCIÁLNÍ EVOLUCE

Mgr. Michal Král

Katedra matematiky, Přírodovědecká fakulta, Ostravská univerzita v Ostravě, 30. dubna 22, 701 03 Ostrava, 607 248 130, p11145@student.osu.cz

Abstrakt

Práce pojednává o kompaktní diferenciální evoluci a její možné implementaci v prostředích Matlab a GNU Octave. Tato varianta diferenciální evoluce je testována na sadě testovacích funkcí.

Klíčová slova: *Diferenciální evoluce; mutace; křížení; selekce; ořezané normální rozdělení*

Úvod

Problém globální optimalizace se řeší v mnoha praktických úlohách. Definujme ho následovně. Pro danou účelovou funkci $f : D \to \mathbb{R}$, $D \subset \mathbb{R}^d$ hledáme bod globálního minima $\mathbf{x}^* = \arg\min_{\mathbf{x}\in D} f(\mathbf{x})$ v souvislé oblasti $D = \prod_{i=1}^{d} \langle a_i, b_i \rangle$, $a_i < b_i$, $i = 1, \ldots, d$. Samozřejmě za předpokladu, že pro $\forall \mathbf{x} \in D$ umíme vyhodnotit účelovou funkci $f(\mathbf{x})$ s požadovanou přesností.

Jelikož neexistuje deterministický algoritmus obecně řešící tento problém (jedná se o NP-obtížný problém), nastal rozvoj stochastických algoritmů. Ty jsou velmi často schopny nalézt prakticky použitelné řešení v rozumném čase. Jedním z těchto algoritmů je diferenciální evoluce (DE), která se stala velmi populární pro svou jednoduchost a dosažené výsledky při řešení mnoha různých problémů z praxe.

DE náleží do třídy stochastických algoritmů nazvané evoluční algoritmy. Ty pracují s populací jedinců (bodů) reprezentujících možná řešení problému v oblasti D a využívající evoluční operátory, kterými jsou mutace, křížení a selekce. Principem DE je pohybovat populací v oblasti směrem k nejlepšímu řešení za pomoci evolučních operátorů. Algoritmus DE vytváří nové generace jedinců tak, že každému jedinci ze staré generace vytvoří jeho konkurenta a do nové generace je začleněn lepší z této dvojice podle hodnoty účelové funkce. Více informací o DE viz [4].

DE uchovává neustále v paměti celou svou populaci jedinců. Doporučuje se volit populaci o velikosti $10d$, což s rostoucí dimenzí řešeného problému je pro slabší hardware limitujícím faktorem pro paměťové nároky či výpočetní výkon. Proto byla navržena kompaktní diferenciální evoluce (KDE) [1], která i přes tato omezení je schopna provést optimalizační proces.

KDE je v jistém smyslu odlehčenou verzí DE, jelikož nepracuje s celou populací jedinců. Když je nutné získat jedince, protože si ho žádají evoluční operátory, dojde k jeho postupnému vygenerování po jednotlivých souřadnicích x_i. Souřadnice jsou popsány pomocí hustot pravděpodobností s parametry středních hodnot μ_i a směrodatných odchylek σ_i. Principem KDE je adaptace parametrů μ_i a σ_i proměnných x_i. Algoritmus se snaží hodnotami σ_i přiblížit k nule a hodnotami μ_i k nejlepšímu řešení. Tedy v paměti stačí uchovávat pouze $2d$ hodnot.

Generování jedince

Autoři článku [1] bez újmy na obecnosti vychází ze vzorce pro ořezanou hustotu pravděpodobnosti (truncated Gaussian PDF) pro i-tou souřadnici $x_i \in \langle -1, 1 \rangle$ bodu $\mathbf{x}$. Výsledný vztah pro její vygenerování je ve tvaru

$$x_i = \mathrm{CDF}^{-1}\left(\mathrm{rand}\left(0, 1\right)\right),$$

kde CDF^{-1} je inverzní funkce (kvantilová funkce) k distribuční funkci CDF (cumulative distribution function) pro hustotu PDF a $\mathrm{rand}(0, 1)$ je náhodné číslo z rovnoměrného rozdělení na intervalu $\langle 0,1)$. Je nutné dodat, že pokud by se jednalo o obecný interval $\langle a_i, b_i\rangle$, musela by se provést transformace souřadnice x_i do tohoto intervalu.

Výhodnější je zvolit pro ořezanou hustotu pravděpodobnosti i-té souřadnice $x_i \in \langle a_i, b_i\rangle$ vztah (1), viz [3]. Souřadnice se netransformují a prostředí Matlab či GNU Octave obsahují již implementované potřebné funkce pro její výpočet.

$$\mathrm{PDF}\left(x_i \,|\, \mu_i, \sigma_i\right) = \frac{\mathrm{e}^{-\frac{(x_i-\mu_i)^2}{2\sigma_i^2}}}{\sqrt{2\pi}\sigma_i\left(\Phi\left(\frac{b_i-\mu_i}{\sigma_i}\right) - \Phi\left(\frac{a_i-\mu_i}{\sigma_i}\right)\right)} \tag{1}$$

Tedy výsledný vztah pro vygenerování i-té souřadnice x_i bodu $\mathbf{x}$ je ve tvaru

$$x_i = \sigma_i\, \Phi^{-1}\left[\left(\Phi\left(\frac{b_i-\mu_i}{\sigma_i}\right) - \Phi\left(\frac{a_i-\mu_i}{\sigma_i}\right)\right)\mathrm{rand}(0,1) + \Phi\left(\frac{a_i-\mu_i}{\sigma_i}\right)\right] + \mu_i\,, \tag{2}$$

kde funkce $\Phi(\alpha) := \mathrm{normcdf}(\alpha)$ a $\Phi^{-1}(\alpha) := \mathrm{norminv}(\alpha)$ jsou distribuční a kvantilovou funkcí pro hodnotu α z normovaného normálního rozdělení s parametry $\mu = 0$ a $\sigma^2 = 1$.

Popis algoritmu

Algoritmus KDE se skládá z pěti důležitých částí zvýrazněných podtržením v pseudokódu níže, které budou postupně vyloženy. Popisovaná verze se označuje pe-cDE/rand/1/bin, viz [1]. Jedná se o tzv. trvale elitářskou verzi. To znamená, že na začátku algoritmu se vygeneruje náhodně bod **elite** reprezentující nejlepší řešení, který je pak v každém cyklu porovnáván se svým konkurentem $\mathbf{x}_{\mathit{off}}$, který vzniká pomocí operátorů mutace a křížení. Vyrovná-li se mu hodnotou účelové funkce nebo jej předčí, stane se $\mathbf{x}_{\mathit{off}}$ novým **elite**.

t=0
Inicializace parametrů rozdělení
generuj **elite** (nejlepší řešení) pomocí parametrů rozdělení
while (nesplněná ukončovací podmínka) **do**
    Mutace
    Křížení
    Selekce
    Adaptace parametrů rozdělení
    t=t+1
end while

Existuje i verze KDE nazvaná ne-cDE/rand/1/bin. Tato verze je opakem elitářské verze, tedy **elite** je po pevně zvoleném nepřerušeném počtu vítězných iterací η nahrazen aktuálním $\mathbf{x}_{\mathit{off}}$.

Inicializace parametrů rozdělení

Inicializují se hodnoty středních hodnot μ_i a směrodatných odchylek σ_i jednotlivých souřadnic x_i popsaných svými hustotami. Střední hodnoty se nastavují na nulovou hodnotu a hodnoty směrodatných odchylek na nějaké velké kladné číslo λ. Volba tohoto parametru závisí na zadané optimalizační úloze.

Mutace

Užitá mutace se označuje jako rand/1. Vygenerují se tři různé body $\mathbf{x}_r, \mathbf{x}_s$ a $\mathbf{x}_t$, ze kterých podle vztahu (3) dostaneme bod $\mathbf{x}'_{\mathit{off}}$,

$$\mathbf{x}'_{\mathit{off}} = \mathbf{x}_t + F(\mathbf{x}_r - \mathbf{x}_s). \tag{3}$$

Parametr $F \in \langle 0, 2\rangle$ se nazývá mutační konstanta a doporučuje se volit její hodnotu v rozmezí 0,3 – 0,9.

Křížení

Použito je binomické křížení. Křížením bodů $\mathbf{x}'_{\mathit{off}}$ s **elite** vzniká nový bod (konkurent) $\mathbf{x}_{\mathit{off}}$ a to tak, že každá jeho souřadnice $elite_i$ je nahrazena souřadnicí x'_{off_i} s pravděpodobností CR. Pokud žádná souřadnice bodu **elite** nebyla nahrazena souřadnicí bodu $\mathbf{x}'_{\mathit{off}}$ nebo nastal-li případ volby CR $= 0$, pak se vybere náhodně jedna souřadnice x'_{off_j}, $j \in \{1, \ldots, d\}$, která nahradí $elite_j$. Parametr CR $\in \langle 0, 1\rangle$ se nazývá práh krížení a je doporučeno jej volit v rozmezí $0{,}8 - 0{,}9$. Binomické křížení je schématicky znázorněno níže.

$$x_{\mathit{off}_i} = \begin{cases} x'_{\mathit{off}_i}, & \text{if} \quad \mathrm{rand}(0,1) \leq \mathrm{CR} \quad \vee \quad i = j \\ \\ elite_i, & \text{if} \quad \mathrm{rand}(0,1) > \mathrm{CR} \quad \wedge \quad i \neq j \end{cases}$$

Selekce

Zde se porovnají body $\mathbf{x}_{\mathit{off}}$ a **elite**. Podle hodnoty jejich účelové funkce se určí vítěz a poražený. Pokud zvítězí $\mathbf{x}_{\mathit{off}}$, stane se novým **elite**. Selekce je schématicky znázorněna níže.

$$\mathbf{elite} = \begin{cases} \mathbf{x}_{\mathit{off}}, & \text{if} \quad f(\underbrace{\mathbf{x}_{\mathit{off}}}_{winner}) \leq f(\underbrace{\mathbf{elite}}_{loser}) \\ \\ \mathbf{elite}, & \text{if} \quad f(\underbrace{\mathbf{x}_{\mathit{off}}}_{loser}) > f(\underbrace{\mathbf{elite}}_{winner}) \end{cases}$$

Adaptace parametrů rozdělení

V této části algoritmu se adaptují parametry μ_i^t a σ_i^t. Vystupuje zde nový parametr N_p, který udává počet jedinců naší virtuální populace. Adaptivní vztahy jsou:

$$\mu_i^{t+1} = \mu_i^t + \frac{1}{N_p}\left(winner_i - loser_i\right), \tag{4}$$

$$\sigma_i^{t+1} = \sqrt{(\sigma_i^t)^2 + (\mu_i^t)^2 - (\mu_i^{t+1})^2 + \frac{1}{N_p}(winner_i^2 - loser_i^2)}. \tag{5}$$

Nastane-li případ, že parametr μ_i^{t+1} v (4) nabude hodnoty mimo interval $\langle a_i, b_i\rangle$, pak máme různé možnosti, jak nastalou událost řešit. Můžeme např. μ_i^{t+1} náhodně vygenerovat z intervalu $\langle a_i, b_i\rangle$, lze použít zrcadlení a nebo můžeme tento stav ignorovat. Zvolme možnost zrcadlení.

Rozveďme dále adaptivní vztah (5). Aby výraz pod odmocninou měl smysl, společně s podmínkou plynoucí pro σ_i^{t+1} z (2) musí platit, že $\left|(\mu_i^t)^2 - (\mu_i^{t+1})^2 + \frac{1}{N_p}(winner_i^2 - loser_i^2)\right| < (\sigma_i^t)^2$. Pak lze vždy provést adaptaci parametrů σ_i^t a vztah (5) lze přepsat do pseudokódu:

if $\left|\,(\mu_i^t)^2 - (\mu_i^{t+1})^2 + \frac{1}{N_p}(winner_i^2 - loser_i^2)\,\right| < (\sigma_i^t)^2$ **then**
  $(\sigma_i^{t+1})^2 = (\sigma_i^t)^2 + (\mu_i^t)^2 - (\mu_i^{t+1})^2 + \frac{1}{N_p}(winner_i^2 - loser_i^2)$
else
  neadaptuj σ_i^t
end if

Experimentální výsledky

Pro testovací účely byla vybrána šestice různě obtížných funkcí pro hledání globálního minima, včetně zvolených oblastí z [2]. Dimenze testovacích úloh byly zvoleny $d = 5$ a $d = 10$. Parametry byly zvoleny $F = 0{,}9$, $CR = 0{,}9$ a $N_p = 2d$. Parametr λ byl volen na základě

Tabulka 1. Experimentální výsledky

		d=5		d=10	
ε	funkce	pr. rel. ch. $\pm$ sm. odch.	f_{best}	pr. rel. ch. $\pm$ sm. odch.	f_{best}
ε_1	Ackley	2.00e-03 $\pm$ 3.50e-03	1.05e-04	3.52e+00 $\pm$ 6.74e+00	3.29e-05
	Dejong1	4.05e-06 $\pm$ 9.43e-06	1.49e-09	2.29e-07 $\pm$ 4.09e-07	1.15e-09
	Griewank	1.75e-01 $\pm$ 1.28e-01	1.23e-02	1.31e-01 $\pm$ 2.70e-02	9.16e-02
	Rastrigin	4.98e-01 $\pm$ 7.4e-01	4.10e-07	9.95e-01 $\pm$ 7.71e-01	7.21e-06
	Rosenbrock	1.42e+00 $\pm$ 1.10e+00	2.01e-02	8.76e+00 $\pm$ 14.22e+00	9.50e-03
	Schwefel	-8.92e-02 $\pm$ 7.71e-02	-2.10e+03	-1.40e-01 $\pm$ 8.29e-02	-4.19e+03
ε_2	Ackley	3.10e-04 $\pm$ 8.47e-04	7.19e-08	8.91e-01 $\pm$ 3.88e+00	1.06e-07
	Dejong1	2.21e-07 $\pm$ 7.16e-07	1.40e-14	1.98e-12 $\pm$ 4.54e-12	1.56e-16
	Griewank	1.19e-01 $\pm$ 7.90e-02	9.90e-03	1.24e-01 $\pm$ 8.05e-02	2.46e-02
	Rastrigin	5.97e-01 $\pm$ 9.12e-01	3.77e-13	4.98e-01 $\pm$ 5.89e-01	6.68e-13
	Rosenbrock	1.05e+00 $\pm$ 1.12e+00	6.01e-04	5.97e+00 $\pm$ 5.46e+00	4.70e-02
	Schwefel	-9.11e-02 $\pm$ 7.07e-02	-2.10e+03	-1.15e-01 $\pm$ 8.31e-02	-4.19e+03

konkrétní úlohy. Pro všechny zadané oblasti platí $b_i = -a_i$ a proto $\lambda_i = 2\lceil b_1 \rceil$. Ukončovací podmínka byla sestavena jako výraz $(\max(\sigma_1^{t+1}, \ldots, \sigma_d^{t+1}) < \varepsilon \wedge t < 5000d)$, kde $\varepsilon_1 = 1{,}0 \times 10^{-3}$ a $\varepsilon_2 = 1{,}0 \times 10^{-6}$. Pro každou vybranou funkci za daných parametrů, dimenze a ukončovací podmínky bylo provedeno 20 spuštění programu. Experimentální výsledky zobrazuje tabulka 1, kde jsou uvedeny průměry relativních chyb (pr. rel. ch.) s jejich směrodatnou odchylkou (sm. odch.) a hodnoty účelové funkce f_{best} nejlepšího nalezeného minima.

Závěr

KDE vždy našla použitelné řešení vyjma Rosenbrokovy a Griewankovy funkce, kde je nutné snížit ε a navýšit maximální počet cyklů k nalezení lepšího řešení. Experimentální výsledky na testovacích úlohách potvrdily, že volbou menšího ε se dosáhne lepších řešení.

Poděkování

Rád bych poděkoval doc. Ing. Josefu Tvrdíkovi, CSc. za cenné připomínky a neutuchající zásobu pozitivní energie ke studiu KDE.

Literatura

[1] Mininno, E., Neri, F., Cupertino, F., Naso, D. *Compact differential evolution*, IEEE Transactions on Evolutionary Computation, vol. 15, no. 1, February 2011, pp. 32-54

[2] Molga, M., Smutnicki, C. *Test functions for optimization needs* [online]. May 2005 [cit. 10. dubna 2012]. Dostupné na World Wide Web: http://www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf

[3] *Truncated normal distribution* [online]. 2012 [cit. 10. dubna 2012]. Dostupné na World Wide Web: http://www.ntrand.com/truncated-normal-distribution/

[4] Zelinka, I., Oplatková, Z., Šeda, M., Ošmera, P., Včelař, P. *Evoluční výpočetní techniky-principy a aplikace.* 1. vydání. Praha: BEN-Technická literatura, 2009, kap. 11, str. 235-245. ISBN 978-80-7300-218-3

Abstract

The work deals with the compact differential evolution and its possible implementation in environment Matlab and GNU Octave. This variant of differential evolution is tested on benchmark test functions.