

Aplikace matematiky

Jaroslav Hrouda

Roklový algoritmus pro určení minima funkce několika proměnných

Aplikace matematiky, Vol. 11 (1966), No. 4, 271–277

Persistent URL: <http://dml.cz/dmlcz/103029>

Terms of use:

© Institute of Mathematics AS CR, 1966

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* <http://dml.cz>

ROKLOVÝ ALGORITMUS PRO URČENÍ MINIMA FUNKCE
NĚKOLIKA PROMĚNNÝCH

JAROSLAV HROUDA

(Došlo dne 22. května 1965.)

1. ÚVOD

Mějme diferencovatelnou funkci n proměnných

$$Q(x_1, x_2, \dots, x_n),$$

definovanou v otevřené oblasti. Předpokládáme, že Q má ve své definiční oblasti lokální minima. Chceme určit polohu některého z nich.

Do dnešní doby bylo vyvinuto množství metod spádového typu, které se zpravidla používají pro řešení této úlohy při $n \geq 2$ a složitější funkci Q . Tyto metody jsou však buď příliš komplikované nebo pomalu konvergující, jsou-li aplikovány na funkci obsahující *rokli*. Gelfand a Cetlin navrhli k překonání obtíže jednorozměrných roklí tzv. *roklovou metodu* [1], [2]. K extrapolaci směru rokly používají roklový krok pevné délky považující jej za charakteristiku dané funkce Q . Prakticky je však obtížné tuto charakteristiku dobře stanovit. Není rovněž možné zakončit proces konvergence ke stacionárnímu bodu.

Baer v jedné poznámce [3] načrtl algoritmus, jehož idea je shodná s roklovou metodou. Tento článek je pokusem o rozvedení a úpravu Baerovy poznámky. Popisuje se zde jednoduchý algoritmus (roklový algoritmus) realizující roklový princip s automatickou regulací délky roklového kroku.

2. ROKLOVÝ ALGORITMUS

Sestrojíme dvě posloupnosti n -rozměrných bodů

$$x^{-1}, x^0, x^1, \dots, x^{k-1}, x^k, \dots,$$

$$y^0, y^1, \dots, y^{k-1}, \dots$$

a dvě posloupnosti kladných čísel

$$\lambda^0, \lambda^1, \dots, \lambda^k, \dots,$$

$$\mu^0, \mu^1, \dots, \mu^k, \dots$$

tak, že posloupnost $\{x^k\}$ bude konvergovat k bodu lokálního minima funkce Q .

Roklový postup. Body x^{k-1}, x^k proložíme přímkou (roklová přímka). Je-li $Q(x^k) > Q(x^{k-1})$, zaměníme označení bodů x^{k-1}, x^k . Označíme vektor

$$v^k = \frac{1}{\|x^k - x^{k-1}\|} (x^k - x^{k-1}).$$

Postupujeme po roklové přímce v bodech

$$x^k(m) = \begin{cases} x^k + m\mu^k v^k, & m = 1, 2, \dots, M_2, \\ x^k + (M_2 + \delta + \delta^2 + \dots + \delta^{m-M_2}) \mu^k v^k, & m = M_2 + 1, \dots, \end{cases}$$

dokud nenastane pro nějaké $m = m_0^k$ vzrůst Q ,¹⁾ tj.

$$Q(x^k(m-1)) \geq Q(x^k(m)), \quad m = 1, 2, \dots, m_0^k - 1,$$

$$Q(x^k(m_0^k - 1)) < Q(x^k(m_0^k)).$$

Zde je $1 < \delta \leq 2$, význam M_2 viz níže. Nyní položíme

$$y^k = \begin{cases} x^k + \beta \mu^k v^k, & \text{je-li } m_0^k = 1, \\ x^k(m_0^k - 1), & \text{je-li } m_0^k > 1, \end{cases}$$

$$\mu^{k+1} = \begin{cases} \frac{1}{2} \mu^k, & \text{je-li } m_0^k < M_1, \\ \mu^k, & \text{je-li } M_1 \leq m_0^k \leq M_2, \\ 2\mu^k, & \text{je-li } M_2 < m_0^k. \end{cases}$$

M_1, M_2 jsou přirozená čísla splňující $1 < M_1 < M_2$; pro koeficient β platí $0 \leq \beta \leq 1$.

Spádový postup. Je-li grad $Q(y^k) = 0$, dosáhli jsme stacionárního bodu. Jinak označíme vektor

$$u^k = - \frac{1}{\|\text{grad } Q(y^k)\|} \text{grad } Q(y^k).$$

Postupujeme ve směru antigradientu v bodech

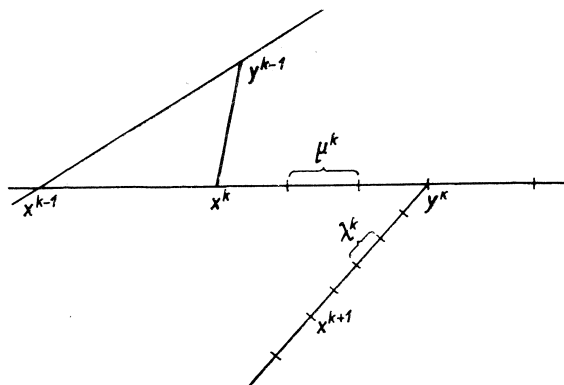
$$y^k(l) = \begin{cases} y^k + l\lambda^k u^k, & l = 1, 2, \dots, L_2, \\ y^k + (L_2 + \delta + \delta^2 + \dots + \delta^{l-L_2}) \lambda^k u^k, & l = L_2 + 1, \dots, \end{cases}$$

¹⁾ O možnosti, že nedojde ke vzrůstu, viz poznámku v odst. 3.

dokud nenastane pro nějaké $l = l_0^k$ vzrůst Q .²⁾ Pak položíme

$$x^{k+1} = \begin{cases} y^k + \alpha \lambda^k u^k, & \text{je-li } l_0^k = 1, \\ y^k(l_0^k - 1), & \text{je-li } l_0^k > 1, \end{cases}$$

$$\lambda^{k+1} = \begin{cases} \frac{1}{2}\lambda^k, & \text{je-li } l_0^k < L_1, \\ \lambda^k, & \text{je-li } L_1 \leq l_0^k \leq L_2, \\ 2\lambda^k, & \text{je-li } L_2 < l_0^k. \end{cases}$$



Obr. 1.

Zde obdobně jako dříve jsou L_1, L_2 přiřazená čísla splňující $1 < L_1 < L_2$; koeficient α je v mezích $0 < \alpha \leq 1$.

Celá tato procedura představuje jednu iteraci roklového algoritmu.

3. POZNÁMKY K ALGORITMU

Z popisu algoritmu je patrné, že číslo μ^k je analogie Gelfand-Cetlinova roklového kroku. Číslo λ^k můžeme nazvat spádovým krokem. Délky těchto kroků se v průběhu výpočtu samočinně upravují tak, aby počet potřebných funkčních hodnot Q byl zhruba v rozmezí M_1 až M_2 při roklovém postupu a L_1 až L_2 při spádovém postupu. V blízkosti ostrého lokálního minima konvergují k nule. Parametr δ způsobuje geometrické prodlužování délky kroku μ^k , resp. λ^k v případě, že není dosaženo vzrůstu Q dosti brzo (po M_2 , resp. L_2 krocích).

K otázce konvergence: připusťme na okamžik závislost koeficientů α, β na indexu k ; volbou dostatečně malých α, β lze dosáhnout zmenšení hodnoty Q v každé iteraci, takže roklový algoritmus bude pracovat jako spádová metoda. Prakticky však není nutné, aby Q stále klesalo — koeficienty α, β mohou být konstantní.

²⁾ O možnosti, že nedojde ke vzrůstu, viz opět odst. 3.

Fluktuace z monotonie posloupnosti $\{Q(x^k)\}$ se budou vyskytovat zejména při $l_0^k = 1$ nebo $m_0^k = 1$. Vždy se však záhy zlikvidují po zmenšení roklového nebo spádového kroku.

Menší efektivnost roklového postupu, dokud není nalezena rokle, se vyrovnává skutečností, že se při něm nepočítá gradient. Z tohoto důvodu je roklový algoritmus někdy výhodný i pro funkce neobsahující rokle. Dodejme, že programování algoritmu je usnadněno tím, že roklový a spádový postup se dá vyjádřit jednou procedurou.

Na začátku výpočtu zvolíme bod x^0 a čísla λ^0, μ^0 (podle variability funkce Q ; není-li o povaze Q nic známo, raději menší). Položíme

$$\begin{aligned} x_1^{-1} &= x_1^0 + \mu^0, \\ x_i^{-1} &= x_i^0, \quad i = 2, \dots, n. \end{aligned}$$

Lze doporučit tyto hodnoty parametrů algoritmu:

$$\alpha = \frac{1}{3}, \quad \beta = 1, \quad \delta = \frac{3}{2}, \quad L_1 = 3, \quad L_2 = 5, \quad M_1 = 2, \quad M_2 = 3.$$

V odst. 2 nebyly uvažovány následující eventuality:

- funkce Q je v definiční oblasti nerostoucí v bodech $x^k(m)$ roklové přímky,
- analogicky pro body $y^k(l)$ v antigradientním směru,
- bod x^k splývá s bodem x^{k-1} ,
- proces se zacykloval.

Možnost případů a,b) závisí na tvaru funkce Q , počátečním bodu a okamžité „přesnosti“ algoritmu, tj. na λ^k, μ^k, δ . Nastane-li a) nebo b), můžeme vzít poslední získaný bod x^k nebo y^{k-1} jako nový počátek a začít algoritmus znovu s menšími hodnotami λ^0, μ^0, δ . Neuspějeme-li, bude nejlépe zvolit zcela jiný počáteční bod. Příklad c) nastane nejspíše při numerické realizaci algoritmu, počítá-li se na hranici dosažitelné přesnosti. Pak samozřejmě výpočet ukončíme. Jinak je možno c) ošetřit jako předchozí případy. K zacyklování procesu (d) může dojít např., je-li $x^{k+1} \equiv x^{k-2}$, $\lambda^{k+1} = \lambda^{k-1}$, $\mu^{k+1} = \mu^{k-1}$ a měnilo se označení x^k, x^{k-1} . Tato a podobné situace jsou však krajně nepravděpodobné; není proto nutné na ně brát zřetel při programování algoritmu.

Je-li y^k stacionární bod, může být lokálním minimem, maximem nebo sedlovým bodem. První možnost je ovšem nejpravděpodobnější.

Prakticky se osvědčilo aproximovat $\partial Q / \partial x_i$ diferencemi

$$[Q(x_1, \dots, x_i + h_i, \dots, x_n) - Q(x_1, \dots, x_i, \dots, x_n)] / h_i.$$

Je však třeba dbát, aby přírůstky h_i nebyly řádově větší než λ^k (stačí učinit h_i závislými na k a položit $h_i^k = \min \{h_i^0, \lambda^k\}$, kde h_i^0 jsou dána).

Výpočet ukončíme podle jednoho z těchto kritérií:

- čísla λ^k, μ^k jsou v několika následných iteracích menší než dané ε ,
- ustane pokles Q .

První kritérium zajišťuje — do jisté míry — požadovanou přesnost výsledku. Druhé kritérium je obecnější vzhledem k tomu, že v případě neostrého minima nemusí čísla λ^k, μ^k konvergovat k nule.

4. NUMERICKÁ ILUSTRACE

Ukážeme chování roklového algoritmu na funkci, která se již tradičně objevuje v literatuře jako test nových metod [5], [3], [6], [8]:

$$Q(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2.$$

Tato funkce obsahuje roklí parabolického tvaru. V tabulce je uvedeno 27 iterací algoritmu z počátečního bodu $x^0 = (-1,2; 1)$, v němž je $Q = 24,2$. Byly použity

Tabulka

k	$Q(y^k)$	m_0^k	μ^k	$Q(x^{k+1})$	l_0^k	λ^k	$\ x^{k+1} - x^k\ $
0	4,000 000	4	0,05	3,999 599	2	0,01	0,160 000
1	7,892 609	1	0,1	3,929 892	10	0,005	0,050 954
2	2,561 141	8	0,05	2,428 628	4	0,01	0,760 234
3	2,247 338	2	0,1	2,203 781	3	0,01	0,105 478
4	1,819 524	3	0,1	1,750 670	3	0,01	0,204 349
5	1,580 815	2	0,1	1,512 440	4	0,01	0,109 495
6	1,141 638	3	0,1	1,057 782	4	0,01	0,203 529
7	0,950 760	2	0,1	0,858 197	4	0,01	0,101 406
8	0,739 718	2	0,1	0,684 416	3	0,01	0,102 136
9	0,603 409	2	0,1	0,540 796	3	0,01	0,100 768
10	0,463 916	2	0,1	0,422 524	3	0,01	0,101 855
11	0,313 052	3	0,1	0,248 229	3	0,01	0,198 863
12	0,207 516	2	0,1	0,184 171	2	0,01	0,100 108
13	0,149 427	2	0,1	0,134 601	3	0,005	0,100 481
14	0,077 072	3	0,1	0,062 775	2	0,005	0,199 911
15	0,068 671	1	0,1	0,038 045	5	0,002 5	0,099 838
16	0,011 728	5	0,05	0,007 027	2	0,002 5	0,224 941
17	0,014 870	1	0,1	0,001 283	5	0,001 25	0,099 847
18	0,000 489	2	0,05	0,000 234	2	0,001 25	0,049 975
19	0,000 380	1	0,05	0,000 110	2	0,000 625	0,049 971
20	0,000 470	1	0,025	0,000 453	2	0,000 312	0,024 939
21	0,000 001	3	0,012 5	0,000 002	1	0,000 156	0,025 044
22	0,000 044	1	0,012 5	0,000 044	1	0,000 078	0,012 506
23	0,000 005	1	0,006 25	0,000 003	3	0,000 039	0,006 231
24	0,000 008	1	0,003 125	0,000 006	3	0,000 039	0,003 136
25	0,000 002	1	0,001 562	0,000 000	3	0,000 039	0,001 549
26	0,000 001	1	0,000 781	0,000 000	2	0,000 039	0,000 773

hodnoty parametrů $\alpha, \beta, \delta, L_1, L_2, M_1, M_2$ doporučené v předchozím odstavci. Složky grad Q byly aproximovány prvními pravými diferencemi s $h_1^0 = h_2^0 = 0,0001$. Konečné hodnoty nezávisle proměnných jsou $x_1^{27} = 1,000\ 037$, $x_2^{27} = 1,000\ 078$ a příslušné $Q \doteq 0,296 \cdot 10^{-8}$. Celkem se při výpočtu potřebovaly 202 funkční hodnoty Q .³⁾

Aby bylo možno si učinit představu o efektivnosti roklového algoritmu, byl příklad spočítán též s vynecháním roklového postupu v každé iteraci, tedy srovnatelnou metodou největšího spádu. K dosažení bodu (1,000 178; 1,000 338), v němž $Q \doteq 0,642 \cdot 10^{-7}$, bylo zapotřebí 2786 hodnot Q ve 443 iteracích.

Literatura

- [1] Гельфанд, И. М., Цетлин, М. Л.: Принцип нелокального поиска в системах автоматической оптимизации; ДАН СССР 137 (1961), № 2, 295—298.
- [2] Гельфанд, И. М., Цетлин, М. Л.: О некоторых способах управления сложными системами; Успехи мат. наук 17 (1962), № 1, 3—25.
- [3] Baer, R. M.: Note on an extremum locating algorithm; Comput. J. 5 (1962), No 3, 193.
- [4] Spang, H. A.: A review of minimization techniques for nonlinear functions; SIAM review 4 (1962), No 4, 343—365.
- [5] Rosenbrock, H. H.: An automatic method for finding the greatest or least value of a function; Comput. J. 3 (1960), No 3, 175—184.
- [6] Fletcher, R., Powell, M. J. D.: A rapidly convergent descent method for minimization; Comput. J. 6 (1963), No 2, 163—168.
- [7] Marquardt, D.: An algorithm for least-squares estimation of nonlinear parameters; Journal SIAM 11 (1963), No 2, 431—441.
- [8] Powell, M. J. D.: An efficient method for finding the minimum of a function of several variables without calculating derivatives; Comput. J. 7 (1964), No 2, 155—162.

Резюме

ОВРАЖНЫЙ АЛГОРИФМ ДЛЯ ОПРЕДЕЛЕНИЯ МИНИМУМА ФУНКЦИИ НЕСКОЛЬКИХ ПЕРЕМЕННЫХ

Ярослав Гроуда (JAROSLAV HROUDA)

Основная идея овражного метода Гельфанда-Цетлина заключается в следующем: сойдем с помощью метода наискорейшего спуска в овраг, сделаем шаг постоянной длины на прямой, экстраполирующей направление оврага, поправим эту экстраполяцию методом наискорейшего спуска и т.д. Этот метод в настоящей статье видоизменен таким образом: в овраг сойдем посредством единственного антиградиентного направления. Минимумы на антиградиентной полупрямой и экстраполирующей „овражной“ прямой получим на осно-

³⁾ Výpočet byl proveden na počítači Elliott 803 ve VÚTEChP.

вании таблиц значений минимизированной функции Q . Длины шагов этих таблиц выбираются в ходе счета в зависимости от формы функции Q таким образом, чтобы требовалось как можно меньше вычислений функции Q для достижения желаемой точности. В начале процесса нет никакой отдельной подготовительной фазы. Две основные части алгоритма — овражный и спусковой ходы — формально совпадают, что облегчает программирование. Сходимость овражного алгоритма имеет тот же характер, как сходимость метода спуска (не зависит слишком от расстояния начальной точки от минимума). — Поведение алгоритма иллюстрируется на численном примере.

Summary

THE VALLEY ALGORITHM FOR MINIMIZING A FUNCTION OF SEVERAL VARIABLES

JAROSLAV HROUDA

The basic idea of Gelfand - Cetlin's valley method is to descend down a valley by the method of steepest descent, then to make step of fixed length on the line extrapolating the direction of the valley, to correct this extrapolation by the method of steepest descent, etc. In the present paper, this method is modified as follows: Descend down a valley along only one anti-gradient direction; obtain minima along the anti-gradient ray and the extrapolating „valley“ line on the basis of tables of values of the minimized function Q . The step lengths of these tables are automatically adapted according to the form of Q , so as to have as small number of evaluations as possible to achieve desirable accuracy. The process does not start with any separate preparatory phase. Two basic parts of the algorithm — the valley and descent procedures — are formally identical, which facilitates programming. Convergence properties of the valley algorithm are of the same type as of descent methods (it is not particularly sensitive to the distance from the initial point to a minimum). — A numerical example is given to illustrate the behaviour of the algorithm.

Adresa autora: Jaroslav Hrouda, prom. mat., Výzkumný ústav technicko-ekonomický chemického průmyslu, Štěpánská 15, Praha 2.